

JEDE AUSGABE MIT FIRMIENREGISTER

04/2015 € 6,90

OFFIZIELLER PARTNER VON
game
Bundesverband e.V.

making games

DESIGN | BUSINESS | ART | TECHNOLOGY



THE AGE OF STEAM



ENGINE-VERGLEICH
DIE PROS UND CONS VON UNITY UND
UNREAL ENGINE FÜR MOBILE GAMES.

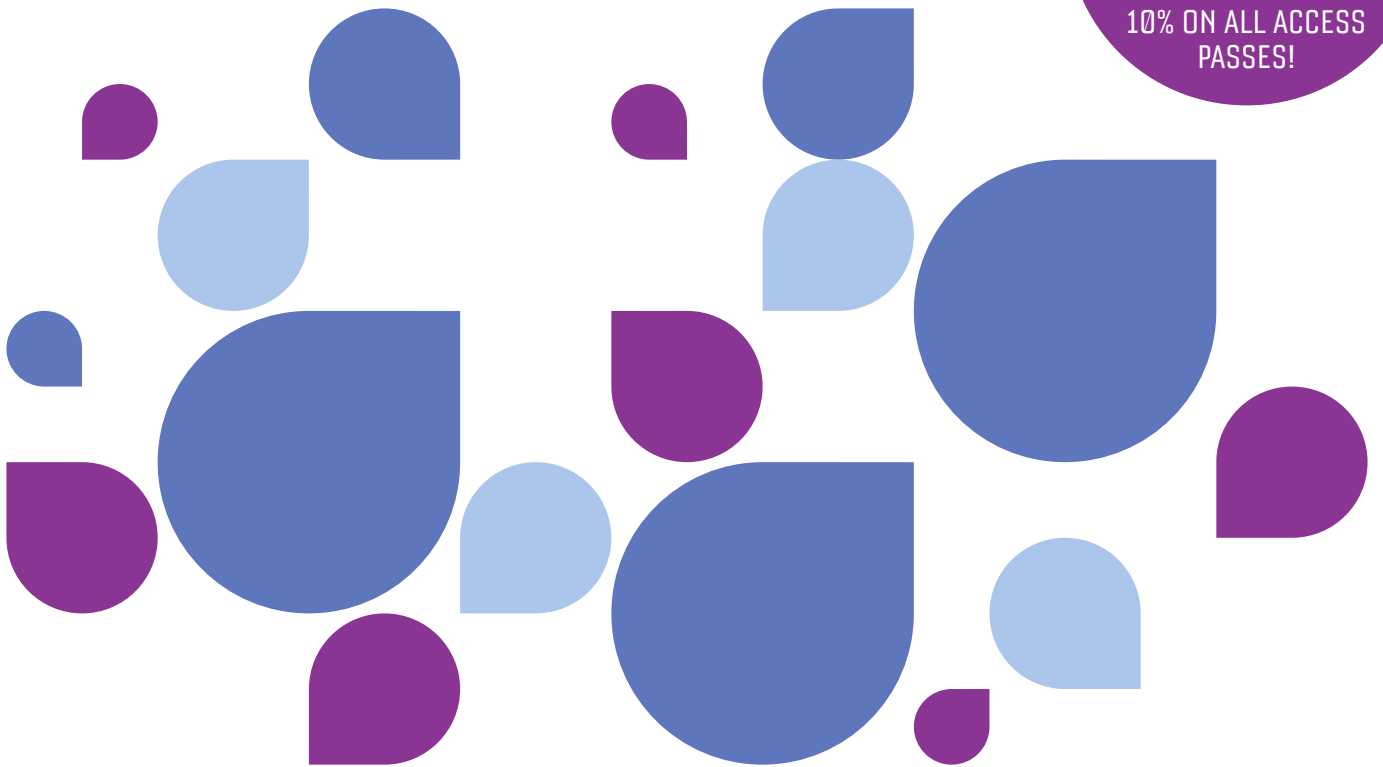
KICKSTARTER-INTERVIEW
YANCEY STRICKLER ÜBER ERFOLGS-
REZEPTE UND DEN DEUTSCHLANDSTART.

DIE ZUKUNFT VON UI-DESIGN
WARUM HTML5 DAS PERFEKTE
WERKZEUG FÜR INTERFACES IST.

GDC EUROPE RETURNS AUGUST 3-4, 2015

**making
games**

READERS, USE CODE
GDCE15MG
TO SAVE AN ADDITIONAL
10% ON ALL ACCESS
PASSES!



SAVE 200€ WITH EARLY REGISTRATION BEFORE JULY 8!

GDC **Eu**

GAME DEVELOPERS CONFERENCE™ EUROPE

CONGRESS-CENTRUM OST KOELNMESSE · COLOGNE, GERMANY

AUGUST 3-4, 2015

GDCEUROPE.COM



GRENZENLOSE FREIHEIT FÜR ENTWICKLER DANK STEAM

Seitdem Mitte Mai »The Witcher 3« erschienen ist, scheint die gesamte Spielerschaft in Geralts Haut um die monsterbedrohten Häuser zu ziehen – so wirkt es zumindest, wenn man Facebook im Auge behält und auch das Team von Making Games nimmt sich hier nicht aus. Der Wirbel um CD Projekts Rollenspielgiganten war gar so groß, dass andere prominente Themen mehr oder weniger untergingen – etwa die Enthüllung von »Assassin's Creed: Syndicate«. Doch dann schaffte es ein anderes polnisches Team namens Destructive Creations, die Fantasy-Idylle zu erschüttern: Als die Entwickler Anfang Juni ein Spiel auf Steam veröffentlichen, das nichts anderes ist als ein »Amoklauf-Simulator«, verdrängt der maue und nur auf Provokation ausgelegte Schund in einigen Ländern »The Witcher 3« sogar von seinem wohlverdienten Platz 1 der Steam-Verkaufscharts – in anderen Ländern, wie Deutschland und Australien, ist er gar nicht erst erschienen. Während Spielmechanik und Inhalt also mehr als fragwürdig sind, hat die Marketingmaschine und Aufregung im Internet hier ihre volle Wirkung gezeigt. Gleichzeitig verdeutlicht Destructive Creations damit aber auch den großen Vorteil von Valves Plattform, auch wenn man es sich in diesem Fall anders gewünscht hätte.

Wie funktioniert Steam eigentlich?

Denn während ein solches Projekt vor Jahren vermutlich gar nicht erst das Licht der digitalen Welt erblickt hätte, da kein Publisher sich die Finger und gleichzeitig eine Menge Kohle verbrennen wollte, haben kleine Teams heute die Möglichkeit, ihre Spiele einfach selbst an den Mann zu bringen. Dass es dabei nicht nur provokativ und amoralisch zugehen muss, zeigen unsere Experten in unserem Titelthema »The Age of Steam«. King Art zum Beispiel erklärt, warum Steam generell großartig für Entwickler ist (Seite 12) und wir werfen einen Blick darauf, welche Faktoren die Verkaufszahlen dort beeinflussen (Seite 16) und wie viel Macht die Spieler mit ihren Reviews ausüben können (Seite 20). Vostok Games, die ehemaligen »S.T.A.L.K.E.R.«-Entwickler, zeigen, wie sie von AAA zu Free2Play gewechselt sind (Seite 36) und die stillalive studios verdeutlichen wie Kickstarter, Steam Greenlight und Early Access einander beeinflussen (Seite 30). So decken wir alle wichtigen Aspekte der Online-Plattform ab und hoffen, dass mit all den Tipps und Tricks jeder Entwickler erfolgreich sein kann, ohne in die unterste Schublade der Gewaltverherrlichung greifen zu müssen.

Viel Spaß mit der aktuellen Ausgabe!

das Team von Making Games

Sebastian Weber
ist Managing Editor vom
Making Games Magazin.



»Dank Steam
haben kleine Teams
heute die Möglichkeit,
ihre Spiele selbst
an den Mann
zu bringen.«

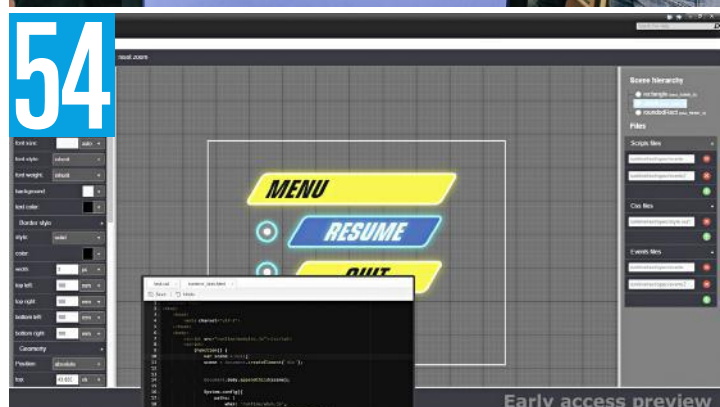
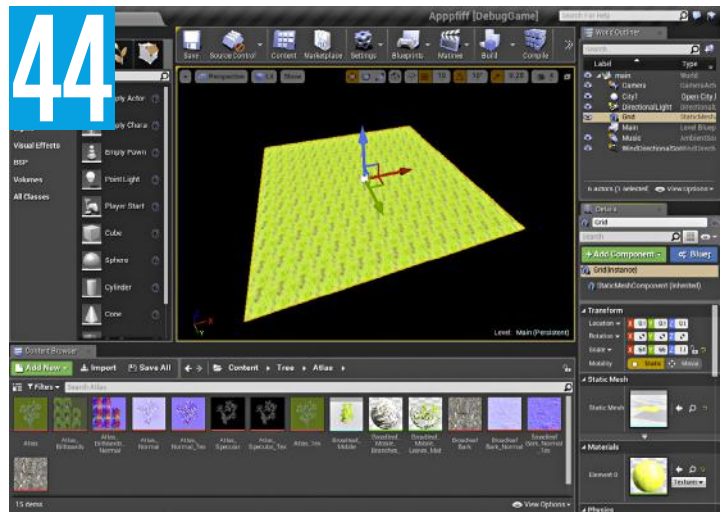


INHALT

04/2015

- 03 Editorial
- 80 Ein Tag bei ... OFM
- 82 Vorschau
- 82 Impressum

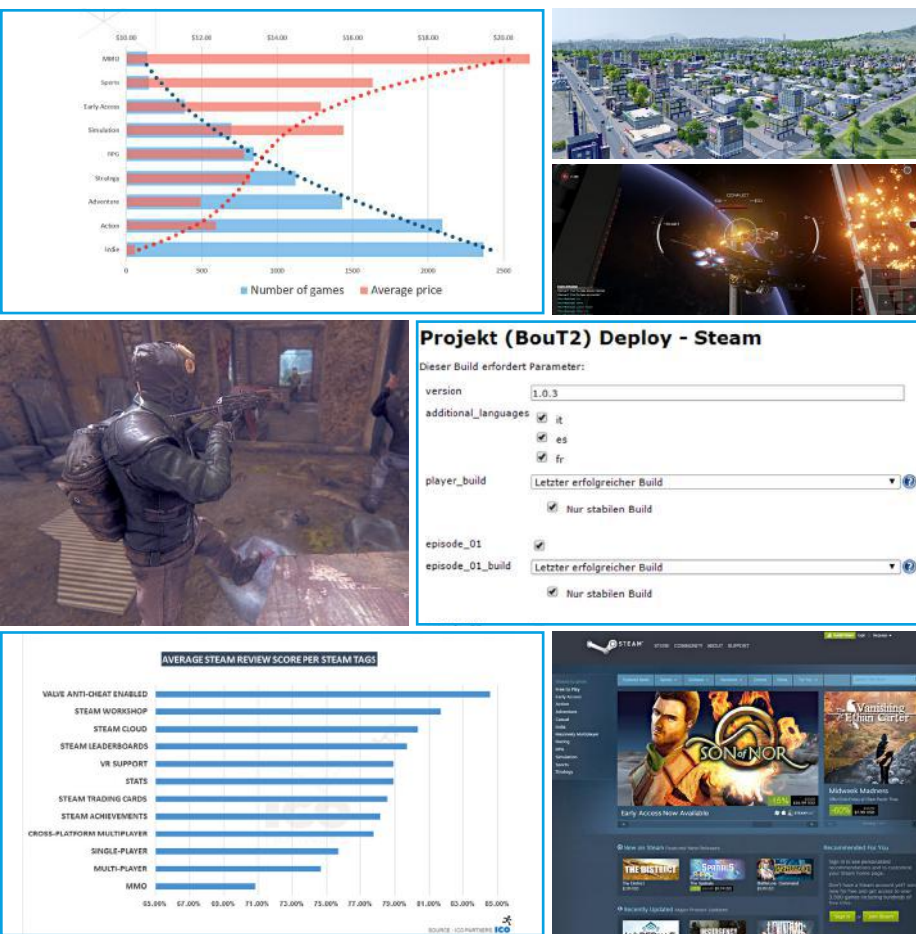
- 06 **Branche / Köpfe**
Bemerkenswertes aus der Branche
- 07 **Eventkalender**
Wichtige Branchenveranstaltungen
- 08 **Passiert auf makinggames.biz**
Interessantes aus dem World Wide Web
- 10 **Making Games Tools**
Die Meinung der Profis zu aktueller Soft- und Hardware
- 11 **Fokus G.A.M.E.**
von André Bernhardt



TITELTHEMA

The Age of Steam

- 12 **Case Study**
Warum Steam für Entwickler großartig ist
von Arne von Ohlsen
- 16 **Analysis**
How to be Successful on Steam
von Sergey Galyonkin
- 20 **Analysis**
A Look into Steam Reviews Data
von Thomas Bideaux
- 23 **Best Practice**
Modding in »Cities: Skylines«
with Steam Workshop
von Jakob Munthe
- 26 **Case Study**
Free Weekend Promotion or the Power of Freebies
von Sam Faulkner
- 30 **Post Mortem**
Son of Nors steiniger Weg auf Steam
von Chris Polus
- 36 **Case Study**
How to Change from AAA to Free2Play on Steam
von Oleg Yavorsky



KICKSTARTER Discover Start Search Projects Sign up Log in

Bloodstained: Ritual of the Night
by Koji Igarashi

31,897 backers
\$2,512,004 pledged of \$600,000 goal
22 days to go

[Back This Project](#) [Raising now](#)

This project will be funded on Sat, Jan 15/2016 5:00 AM CEST

ICAVANIA (n): A gothic, exploration-focused action platformer, designed by one of the godfathers of the genre!

Tokyo, Japan Video Games [Share this project](#)

Koji Igarashi
First created 10/1/2015
@igavania.com
[See full bio](#) [Contact me](#)

- 40 **Post Mortem**
Super Carl – Von Zäunen und Möchtegernhelden
von Marco Apel
- 44 **Case Study**
Unreal vs. Unity – which Engine is Better for Mobile Games
von Guido Schmidt, Daniel Wilkins und Salomon Zwecker
- 50 **Interview mit Dominik Abé, Angelika Achter und Benjamin Feld**
Vom Kinderbuch zur Spiele-App
von Sebastian Weber
- 54 **Case Study**
The Future of Fast and Easy UI-Design
von Stoyan Nikolov
- 58 **Programming Case Study**
Model View Controller mit Unity
von Nurdogan Erdem
- 62 **Case Study**
Rapid Game Development with V-Play
von Christian Feldbacher
- 68 **Interview mit Yancey Strickler**
Ein Tool zur Umsetzung von Ideen
von Patricia Geiger
- 73 **Marktforschung DataFlow**
GameStar-/GamePro-Leserdaten
von Patricia Geiger
- 74 **Firmenregister**
Die Branche im Überblick



MESSE LEIPZIG

HOLT SICH GAMING ZURÜCK

DREAMHACK
LEIPZIG 2016

Seit dem Jahr 2002 ist die Messe Leipzig stark mit der Spieleindustrie verbandelt, immerhin fand seinerzeit die erste Games Convention in ihren Hallen statt. Bis 2008 strömten Spielebegeisterte und Industrievertreter in die sächsische Großstadt. Doch dann entschied der BIU, der als Träger der Games Convention fungierte, die Messe nach Köln umzuziehen und als gamescom neu aufzulegen. Die Messe Leipzig wollte in den Folgejahren mit der Games Convention Online – einer Messe rund um Online- und Mobile-Gaming – weiter mitmischen, musste die Versuche aber 2011 endgültig zu Grabe tragen. Ende Mai 2015 folgte der Paukenschlag: Die Leipziger Messe stellte eine Teaser-Website online, die den Titel »Gaming is Coming Home« trug. Das sorgte für allerlei Spekulationen. Anfang Juni letztlich gab eine Pressekonferenz Aufschluss über die Pläne der Leipziger Messe: Vom 22. bis 24. Januar 2016 wird die Dreamhack in Leipzig stattfinden. Das eSport-Festival, das seine Ursprünge bereits 1994 in Schweden hat und mehrmals im Jahr in unterschiedlichen

europäischen Städten gastiert, gilt als größte LAN-Party der Welt (2014: 104.000 Besucher). Nach den Plänen der Veranstalter sollen bis zu 1.000 LAN-Fans nach Leipzig kommen, um dort drei Tage lang rund um die Uhr in einem speziellen Bereich mit- und gegeneinander zu spielen. Dazu soll es eSport-Turniere und einen Ausstellerbereich für Hardware, Software, Spiele und Zubehör geben. Als Unterstützung hat sich die Messe Leipzig Hardware-Hersteller Schenker an Bord geholt.

Interessant dürfte dabei jedoch werden, welche Firmen die Dreamhack in ihre Jahresplanung neben Messen wie der gamescom oder der bereits etablierten Role Play Convention aufnehmen werden. Außerdem muss die Messe Leipzig dabei genug Besucher anlocken, um sich als der deutsche Standort der Dreamhack langfristig zu etablieren – was zumindest laut Pressemitteilung der Plan zu sein scheint. Doch in Zeiten, in denen »League of Legends«-Matches tausende Zuschauer vor Twitch-Streams locken, dürfte zumindest das mit den eSports-Fans aus dem deutschsprachigen Raum wohl gelingen.

Sebastian Weber

WWW.makinggames.biz
■ aktuelle News und Gerüchte

CLIFFHANGER PRODUCTIONS MELDET INSOLVENZ AN



**CLIFFHANGER
PRODUCTIONS**



Ende Mai musste der österreichische Entwickler Cliffhanger Productions rund um Michael Paeck und Jan Wagner Insolvenz beim Handelsgericht in Wien anmelden. Grund für das Aus soll vor allem ein schwerwiegender Programmierfehler im Ende April veröffentlichten RPG-Strategie-Mix »Shadowrun Chronicles« gewesen sein. Dieser sei vor Veröffentlichung nicht bemerkt worden und hatte dann dazu geführt, dass die Käufer den Titel nicht starten konnten. Es folgte eine wochenlang andauernde Phase, in der das Team den Bug gesucht und letztlich behoben hat. Jedoch blieben in dieser Zeit Einnah-

men aus anderen Projekten aus, was letztlich dazu führte, dass Cliffhanger Productions die laufenden Betriebskosten wie Miete und Gehälter nicht mehr zahlen konnte. Jan Wagner erklärte gegenüber dem österreichischen Nachrichtenportal derstandard.at aber, dass alle Spiele weiterlaufen und soweit möglich weiterhin ausgebaut würden, da der Betrieb unter dem Dach der Schwesterfirma Cliffhanger Productions Games fortgesetzt wird. Inwieweit unter den bisher 29 Mitarbeitern Stellen gestrichen werden müssten, müsste zum Zeitpunkt des Redaktionsschlusses Anfang Juni noch geklärt werden.



GDC EUROPE STELLT ERSTE SPRECHER VOR

Am 03. und 04. August 2015 findet traditionell die GDC Europe im Vorfeld der gamescom statt und lädt Entwickler – ob angehende oder erfahrene – mit Vorträgen, Workshops und interessanten Ausstellern dazu ein, sich auszutauschen und fortzubilden. Inzwischen haben die Veranstalter die ersten Sprecher bekannt gegeben: So wird etwa Adalsteinn Alli Ottarsson, Development Director von Riot Games, erzählen, wie der Erfolg von »League of Legends« die Entwicklung beeinflusst hat und Marinus Rorbech, Tools Programmierer bei IO Interactive, gibt Einblicke, welche Änderungen der Workflows nötig waren, um »Hitman: Absolution« möglich zu machen. Weitere bekannte Unternehmen, die ihre Erkenntnisse teilen, sind unter anderem Warhorse Studios, Ubisoft, Bohemia Interactive, Epic Games und Wargaming. Bis zum 08. Juli gibt es die Tickets für die GDC Europe noch zum vergünstigten Early-Bird-Tarif und Leser der Making Games können zusätzlich zehn Prozent sparen (Promotioncode: GDCE15MG).

JULI 2015

Develop:Brighton

Ort Brighton, Hilton Brighton Metropole
Termin 14. bis 16. Juli 2015
Thema Konferenz
Kosten 103 bis 747 Euro
Homepage www.developconference.com

AUGUST 2015

GDC Europe

Ort Köln, Congress-Centrum Ost Koelnmesse
Termin 03. bis 04. August 2015
Thema Konferenz
Kosten 149 bis 895 Euro
Homepage www.gdceurope.com

gamescom

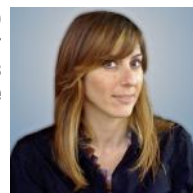
Ort Köln, Messeplatz 1
Termin 05. bis 09. August 2015
Thema Messe
Kosten 13 bis 39 Euro
Homepage www.gamescom.de

gamescom jobs & karriere

Ort Köln, Messeplatz 1
Termin 06. bis 09. August 2015
Thema Messe
Kosten 13 bis 39 Euro
Homepage www.makinggames.biz/jobsundkarriere

Vier Fragen an ...

Meggan Scavio
 ist General Manager
 der Game Developers
 Conference
 bei UBM TechWeb.



Making Games Habt ihr einen Themenschwerpunkt oder einen Trend für die GDC Europe?

Meggan Scavio Diese Frage ist immer schwer zu beantworten, denn die Vorträge drehen sich um so viele verschiedene Themen, dass sich kaum ein einzelner Trend ableiten lässt. Oftmals kristallisieren sich diese eher aus den Gesprächen heraus, die vor Ort geführt werden. Ein sehr weit verbreitetes Thema ist auf jeden Fall das Problem der Sichtbarkeit. Sprich, wie können Entwickler in einem übersättigten Markt aus der Masse herausstechen? Aber auch Fragen wie man ein Free2Play-Spiel am besten startet oder ob man lieber eine vorhandene Engine lizenziert oder eine eigene entwickelt, werden auf der GDC gerne diskutiert, da sie bei jedem Entwickler irgendwann aufkommen.

Making Games Gibt es in Bezug auf die Konferenz und die Preisgestaltung größere Veränderungen?

Meggan Scavio Das Preismodell werden wir nicht ändern, aber wir möchten für die GDC Europe 2015 ein neues Programm starten, das wir »GDC Scholarships« nennen. Dafür stehe ich gerade in Kontakt mit Universitäten und ähnlichen Organisationen, die Verbindung zur Spiele- und IT-Branche haben, um ihnen kostenfreie Tickets zur

Verfügung zu stellen. Als Veranstalter der GDC Europe wollen wir das Wachstum der europäischen Spieleindustrie unterstützen und daher jungen Entwicklern und Studenten unter die Arme greifen, die sich eine Teilnahme sonst nicht leisten könnten.

Making Games Habt ihr für die Ausstellung konkrete Pläne?

Meggan Scavio Die Ausstellungsfläche ist recht begrenzt, da die gamescom direkt nebenan stattfindet. Aber wir arbeiten daran, eine intime Atmosphäre zu schaffen, damit sich die Teilnehmer entspannt unterhalten und mit den Ausstellern sprechen können. Zwar sind wir noch mitten in der Planung, aber ich kann schon jetzt versprechen, dass die Ausstellerbereiche voll sein werden.

Making Games Was ist dein persönliches Ziel für die diesjährige GDC Europe?

Meggan Scavio Im Grunde habe ich das gleiche Ziel wie für jede GDC: Ich möchte, dass die Teilnehmer, Aussteller und Sponsoren am Ende nach Hause gehen und das Gefühl haben, dass sie die Zeit (und das Geld) gut investiert haben. Ich möchte, dass jeder etwas gelernt hat, das er vorher nicht wusste, dass jeder vielversprechende Kontakte geknüpft hat und wenn möglich, dass keiner das Catering gehasst hat!

Eventkalender

www.makinggames.biz

Online-Eventkalender mit Suchfunktion und Veranstaltungsinfos

SIGGRAPH conference

Ort Los Angeles, Los Angeles Convention Center
Termin 09. bis 13. August 2015
Thema Konferenz
Kosten 46 bis 1.454 Euro
Homepage s2015.siggraph.org

Casual Connect USA

Ort San Francisco, Hilton Union Square
Termin 11. bis 13. August 2015
Thema Messe
Kosten 456 bis 4101 Euro
Homepage usa.casualconnect.org

SEPTEMBER 2015

IFA

Ort Berlin, Berlin ExpoCenter City
Termin 04. bis 09. September 2015
Thema Messe
Kosten 37 bis 69 Euro
Homepage www.ifa-berlin.de

Tokyo Game Show

Ort Tokyo, Makuhari Messe
Termin 17. bis 20. September 2015
Thema Messe
Kosten Noch nicht bekannt
Homepage expo.nikkeibp.co.jp/tgs/2015/exhibition/english

OKTOBER 2015

Brasil Game Show

Ort São Paulo, Expo Center Norte
Termin 08. bis 12. Oktober 2015
Thema Messe
Kosten 11 bis 24 Euro
Homepage www.brasilgameshow.com.br

animago Award & Conference

Ort Potsdam-Babelsberg, Filmpark Babelsberg
Termin 15. bis 16. Oktober 2015
Thema Konferenz
Kosten 48 bis 78 Euro
Homepage www.animago.com

Passiert auf makinggames.biz



Telltale Games geht Partnerschaft mit Marvel ein

telltalegames



Nach bekannten Marken wie »The Walking Dead«, »Game of Thrones« und »Minecraft« hat Adventure-Spezialist Telltale Games Ende April den nächsten großen Fisch an Land gezogen: Auf einem Event in San Francisco erklärte Comicbuch-Verlag Marvel (unter anderem bekannt für Figuren wie Hulk, Iron Man, Thor, The Fantastic Four und Spider-Man) seine Pläne für die Zukunft, die sich neben Comics und deren Verfilmungen vor allem auch auf Videospiele beziehen. Dabei gaben die Verantwortlichen bekannt, eine Vereinbarung mit Telltale Games geschlossen zu haben. Zwei Spiele sollen demnach im Jahr 2017 erscheinen. Informationen darüber, für welche Plattformen diese in Entwicklung sind und um welchen der vielen Marvel-Figuren sie sich drehen, blieben sie jedoch schuldig. Ziel der Zusammenarbeit seien Spiele, die sich authentisch und ausgezeichnet anfühlen und gleichzeitig überwältigend aussehen.

Zynga streicht mehr als 300 Stellen



Der amerikanische Browserspiel-Hersteller Zynga hat Mitte Mai verkündet, 18 Prozent seiner Belegschaft streichen zu wollen, was sich auf 364 Arbeitsplätze summiert. CEO Mark Pincus, der erst Mitte April diesen Posten wieder von Ex-Microsoft-Manager Don Matrick hatte, nannte es eine schwierige, aber notwendige Entscheidung. Die Entlassungen sollen auch dazu führen, dass Zyngas Studio in Orlando, Florida geschlossen wird. Grund für den harten Sparkurs dürfte der Verlust von rund 6,7 Millionen Dollar sein, den die Firma im vergangenen Quartal verzeichnen musste (im Vorjahr waren es im gleichen Zeitraum 6,3 Millionen Dollar).

Oculus VR gibt Termin für den Verkaufsstart des Rift und die Hardware-Anforderungen bekannt



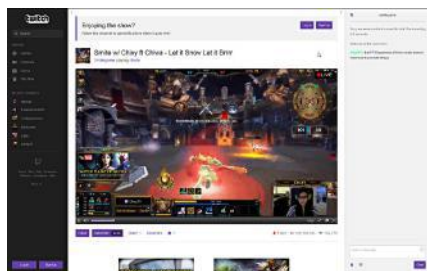
Oculus VR hat Anfang Mai einige Details zu seinem Headmounted-Display Rift bekannt gegeben. So soll das Gerät im ersten Quartal 2016 offiziell in den Handel kommen, während Vorbestellungen bereits Ende 2015 entgegen genommen werden. Wer die voraussichtlich 300 Dollar teure VR-Brille nutzen möchte, der sollte nach Angaben der Entwickler mindestens acht Gigabyte Arbeitsspeicher, einen Prozessor mit der Leistung eines Intel i5-4590 und eine Grafikkarte wie die Nvidia GeForce GTX 970 oder die AMD

Radeon R9 290 in seinem PC verbaut haben. Spiele-Enthusiasten, die in den Genuss des Virtual-Reality-Erlebnisses kommen möchten, müssen also für Brille und benötigte Hardware mit Ausgaben bis zu 1.500 Dollar rechnen. Atman Binstock von Oculus VR gab aber zumindest dahingehend Entwarnung, dass diese minimalen Hardware-Anforderungen über den Lebenszyklus des Rift konstant bleiben sollen und somit im Laufe der Zeit für mehr und mehr Nutzer erschwinglich werden dürften.

Twitch verbietet Streaming von Spielen für Erwachsene



Ende Mai hat Streaming-Plattform Twitch in einem offiziellen Blog-Eintrag verkündet, künftig keine Video-Streams von Spielen für Erwachsene mehr zu dulden. Titel, die vom amerikanischen Entertainment Software Rating Board (kurz: ESRB) die Einstufung »Adults Only« erhalten haben, dürfen demnach nicht mehr ausgestrahlt werden. In Ordnung seien dagegen Versionen solcher Spiele, die das Rating »Mature« erhalten haben – als Beispiel nennt Twitch hier »Grand Theft Auto: San Andreas«, das in einer so gekürzten Fassung vorliegt. Spiele, die von ausländischen Behörden als für Erwachsene gekennzeichnet wurden – also zum Beispiel mit einer USK-Einstufung »ab 18 Jahren« – seien dagegen für Streams erlaubt, solange die ESRB sie niedriger bewertet hat.



E3 öffnet seine Tore teilweise für die Öffentlichkeit



Seit 20 Jahren pilgert die Spieleindustrie zur E3 in den USA und schon immer war die Messe nur für Fachbesucher zugänglich. Anfang Juni hat der Veranstalter The Electronic Software Association (kurz: ESA) bekannt gegeben, dass er das in diesem Jahr ändern wird. Geplant sei, dass alle Firmen, die ESA-Mitglied sind, Tickets bekommen sollen, die sie an ihre Kunden verteilen dürfen. Zwischen 4.000 und 5.000 solcher Eintrittskarten sollen unter die Leute gebracht werden, da laut ESA heute vor allem durch soziale Netzwerke und Videostreaming-Portale die Spieler selbst ein wichtiger Werbepartner für die Industrie seien. Künftig könnte die E3, je nach Erfolg des Testballons, noch weiter geöffnet werden.



Frauennationalmannschaften in FIFA 16 sorgen für Proteste



Mitte Mai hat Electronic Arts angekündigt, dass in »FIFA 16«, das im September erscheinen soll, erstmals auch 12 Frauennationalmannschaften enthalten sein werden, aus denen die Spieler wählen können. Zu diesen zählen unter anderem Deutschland, England, Brasilien, Spanien, USA, Frankreich, Italien und Mexiko. Die virtuellen Mannschaften der Männer und der Frauen ließen sich dabei jedoch nicht durchmischen. Neben Lobeshymnen einiger weiblicher Videospieler, die nun erstmals zu FIFA greifen wollen, hagelte es in den sozialen Netzwerken wie Twitter und Facebook aber auch Spott und Häme, teilweise gar Beschimpfungen für diese Entscheidung – und entfachte so die zuvor geführte Gleichberechtigungsfrage in Videospielen. Peter Moore, Präsident von EA Sports hatte mit solchen Reaktionen offensichtlich nicht gerechnet und kommentierte die Aufregung lediglich damit, dass er traurig sei, dass eine solche Ankündigung so viel Aufmerksamkeit erzeuge.

Kickstarter startet in Deutschland

Mitte Mai ist die Crowdfunding-Plattform Kickstarter offiziell auch in Deutschland gestartet, sodass es für Kreative, Entwickler und Erfinder nun einfacher ist, über die Website Geld für ihre Projekte einzusammeln. Bisher war dafür zum Beispiel ein Konto in den USA nötig – eine durchaus ernst zu nehmende Hürde. Kickstarter-Kampagnen auf der deutschen Seite unterscheiden sich nicht von den Kampagnen wie sie bisher zum Beispiel in den USA, in England, den Niederlanden, Schweden, Dänemark oder Norwegen gestartet werden konnten. Alle gespendeten Beträge von Unterstützern werden in Euro abgerechnet und können innerhalb von Deutschland bis zu einem Betrag von 250 Euro per Lastschrift verbucht werden. Kickstarter selbst erhebt bei erfolgreichen Projekten eine Gebühr von fünf Prozent, außerdem

kommt eine Gebühr von drei Prozent plus 20 Cent pro Beitrag vom Zahlungsdienstleister oben drauf – Beiträge unter 10 Euro werden dagegen mit fünf Prozent plus 5 Cent veranschlagt. Zum Start tummelten sich bereits allerlei spannende bis hin zu verrückten Projekten, von einem Induktionswasserkocher, über eine Zeitmaschine und einem Videospiel zur Nibelungensaga bis hin zu Filmprojekten, etwa von Uwe Boll.

KICKSTARTER



Links

Wie sich CD Projekts »The Witcher« von 2002 vom heutigen Rollenspielhit unterscheidet.
www.makinggames.biz/witcher-2002

Wieso »große« Indie-Studios Kickstarter kaputt machen.
www.makinggames.biz/kickstarter

Warum »Tetris« süchtig macht und doch grausam ist.
www.makinggames.biz/joy-of-tetris

Was mit DirectX 12 alles möglich sein wird.
www.makinggames.biz/dx12-demo



BRANCHEN-GEZWITSCHER @MAKINGGAMES_DE/INTERESTING-PEOPLE



@NOTCH

I'm impressed with how inefficient twitter is at blocking sunglasses spam. It's always sunglasses. Never, like, computer games or chips.
Markus Persson, Minecraft-Erfinder



@GRUMPYGAMER

Sometimes the best tool in your toolbox is a hammer, especially if you just hit things hard enough.
Ron Gilbert, Erfinder von Monkey Island



@DANIELVAVRA

Honestly, Witcher 3 has more powerful female characters, more humanity and emotions and complicated ethical choices than anything in years.
Daniel Vávra, Mitgründer der Warhorse Studios



@THEREALCLIFFYB

Journos, do an anonymous bathroom expose from all the horrible things people have seen in game development restrooms. Ask around. Seriously.
Cliff Bleszinski, Lead Designer von Unreal Tournament



@LOCUST9

Guys, I keep saying it but if you want to make games there is nothing stopping you but you. Really. Go forth and crush it.
David Goldfarb, Game Director von Payday 2



@GEORGE3DR

Should be a horror movie where search engines go away and people have to use libraries and the yellow pages again.
George Broussard, Erfinder von Duke Nukem



@ARMINPOSTS

Saw that @HIDEO_KOJIMA_EN was removed from the credits for Metal Gear Solid 5 so we added him to the credits for GoatZ
Armin Ibrisagic, Game Designer von Goat Simulator



@BENJAMINCOUSINS

Fun game is to imagine the amazingly chaotic games company you could create from all the people who wished me happy birthday on Facebook
Ben Cousins, ehemaliger Executive Producer bei Dice



@LARALYN

I want to find a reason my nerdy text adventure RPG should be in VR so I have an excuse to get a Rift or Vive dev kit.
Laralyn McWilliams, Creative Director bei The Workshop

JOIN THE CONVERSATION

TOOLS

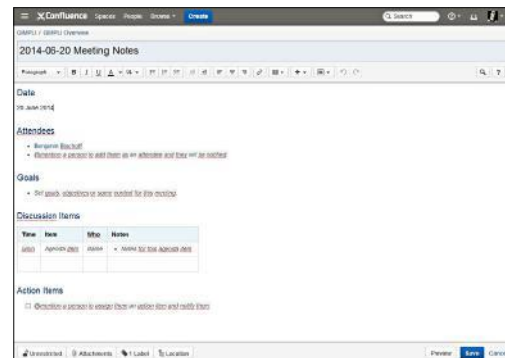
Keine guten Werkzeuge – keine guten Spiele! Experten aus der Branche sagen ihre Meinung zu aktuellen Programm-Updates sowie neuen Tools und verraten ihre persönlichen Software-Geheimtipps.

CONFLUENCE 5.1 WWW.ATLASSIAN.COM/SOFTWARE/CONFLUENCE



Benjamin Bischoff
ist Senior Programmer
Tools & Services
bei Blue Byte.

Die Enterprise Wiki-Lösung Confluence 5.1 ermöglicht uns effizientes teaminternes Knowledge-Sharing und fungiert gleichzeitig als zentrale Stelle für Software-Dokumentationen, Evaluationen und Meeting Notes. Templates und die neuen Blueprints sorgen dabei für inhaltliche und optische Konsistenz und durch die Einbindung von Custom CSS konnten wir unseren Confluence-Space sehr gut personalisieren. Ein weiteres wichtiges Feature neben den zahlreichen neuen Makros ist auch die direkte JIRA-Anbindung, wodurch unter anderem die automatisierte Generierung und Einbindung von Release-Notes möglich ist.



Ein wichtiges Feature von Confluence ist die JIRA-Anbindung, welche die Einbindung von Release-Notes ermöglicht.



PickFu hilft dabei, Meinungsstreitigkeiten in Firmen zu beseitigen.

PICKFU WWW.PICKFU.COM

PickFu ist ein Art automatisiertes, qualitatives A/B-Testing. Man wählt zwei Komponenten aus (Icons, Logos, Schriftzüge, Produkt-namen, Überschriften et cetera) und pflegt diese auf der Webseite ein. PickFu schickt das dann an seinen Pool von Heimarbeitern (meist in den USA), nach einer Stunde hat man 50 Bewertungen mit je einem Satz Begründung. PickFu kostet um die 20 Dollar pro Durchlauf, löst dafür aber allerlei Meinungsstreitigkeiten in Firmen im Nichts auf. Gerade bei schwer zu bewertenden Dingen wie Produktnamen erdet es ungemein, wenn der Lieblingsname durchfällt und sich bei den Begründungen Anmerkungen wie »Das erinnert an eine Seifenmarke« häufen.



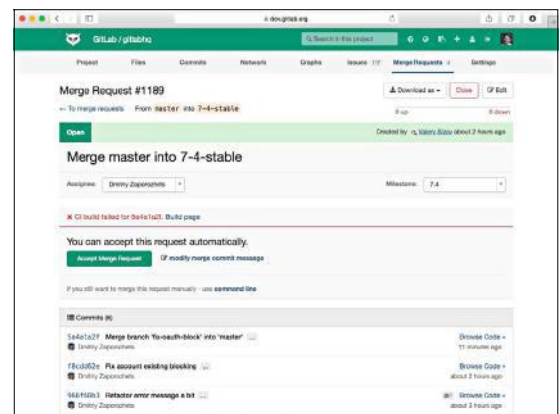
Gunnar Lott
ist Geschäftsführer
von Visibility Communications.

GITLAB WWW.GITLAB.COM



Hans Hamm
ist Head of R&D
bei Deep Silver FISHLABS.

Wie GitHub ist GitLab ein umfangreiches System zur kollaborativen Entwicklung von Software-Projekten, deren Versionsverwaltung mit git erfolgt. Während standardmäßig alle Daten auf einem lokalen Server gespeichert werden, besteht auch die Option, günstige Online-Server auf GitLab.com zu mieten. Die kostenlose, mit MIT lizenzierte Community Edition ist für kleine bis mittelständische Unternehmen vollkommen ausreichend und bietet neben einer kompletten Projektverwaltung mit Rechtevergabe auch die Möglichkeit von Code-Reviews und Annotationen an Commits. Mit Hilfe von GitLab CI kann GitLab außerdem direkt als Continuous-Integration-Lösung eingesetzt werden. Das stellt sicher, dass der Source Code stets sauber und lauffähig bleibt. Die Enterprise Edition ist derzeit ab \$19,90 pro Benutzer pro Jahr erhältlich und verspricht einen besseren Umgang mit großen Daten, Git Hooks zur Kontrolle von Commit-Nachrichten und eine bessere LDAP-Anbindung.



Mit Hilfe von Merge Requests werden Commits erst nach einem Code Review in den Master-Branch zugelassen.

FOKUS GAME

TASK FORCE INDIE – WER WIR SIND!

Vor gut einem Jahr hat der GAME die »Task Force Indie« ins Leben gerufen, um den Wissensaustausch etablierter und neuer Studios zu fördern. André Bernhardt über die Beweggründe und wo der Arbeitskreis inzwischen steht.

Indies sind heutzutage in aller Munde: Waren sie vor sieben Jahren noch ein Phänomen wie die Jungs von 2D Boy, die im Coffeeshop ihres Vertrauens zu zweit das äußerst erfolgreiche »World of Goo« entwickelt und dann auch vermarktet haben, sind sie spätestens seit »Indie Game: The Movie« und dem 2,5 Milliarden-Dollar-»Minecraft«-Exit von Mojang beziehungsweise Notch auf dem Radar der deutschen Games-Industrie aufgetaucht. Vor längerer Zeit machten sich so auch hierzulande die ersten Entwickler abseits der ausgetretenen Pfade auf den Weg, um ihre eigenen Projekte unabhängig zu entwickeln. Seien es Mimimi Productions, Studio Fizbin, Happy Tuesday oder viele, viele andere, die man in einer Kolumne wohl kaum alle erwähnen könnte: Es herrschte Aufbruchsstimmung.

Lebhaftes Indie-Szene

Und die hält an, denn bis zum heutigen Tag im Jahr 2015 hat sich einiges getan, sodass wir mittlerweile über eine lebhaftes deutschsprachige Indie-Szene verfügen, wenn man mal einen kurzen Blick auf die Mitgliederzahl des größten deutschsprachigen Forums wirft: der Indie Arena. Die ist nicht erfolglos, wie man regelmäßig bei den Preisverleihungen der Republik – dem Deutschen Computerspielpreis oder dem Entwicklerpreis – sieht.

Wissensaustausch fördern

Seit 2012 arbeite ich mit unabhängigen Entwicklern zusammen und lerne dabei regelmäßig neue Studios kennen. Diese zumeist kleineren Teams sind mutig, kreativ und leidenschaftlich und es macht großen Spaß, mit ihnen zusammenzuarbeiten. Je nach Hintergrund mangelt es manchmal an Erfahrung, aber war das bei Sproing und Deck13 vor langer Zeit nicht genauso? Man wächst an seinen Herausforderungen.

Im GAME haben wir nun festgestellt, dass kleinere Studios in der Verbandslandschaft keine richtige Plattform haben, um ihre Interessen auch gegenüber der Gesellschaft

und der Politik zu artikulieren. Zudem wollen wir den Wissensaustausch der »Etablierten« und der Newcomer Made in Germany nach vorn bringen. Nicht jeder Fehler muss zweimal gemacht, nicht jede Erfahrung persönlich durchlebt werden. Warum sind also nicht mehr Indies im GAME?

Task Force Indie

Als im Verband die Idee geboren wurde, einen Arbeitskreis zu gründen, der dieses Missverhältnis beleuchtet, war ich sofort Feuer und Flamme. Gemeinsam mit Dieter Schoeller, der als Geschäftsführer des etablierten Indie-Publishers Headup Games auch Sprecher des Arbeitskreises ist, und weiteren Freiwilligen starteten wir im April 2014 den Arbeitskreis Indie, der den GAME attraktiver für kleine Entwickler machen sollte. Von vornherein waren dabei zwei Dinge klar: Der Verband muss erstens hinterfragen, warum es bisher nicht besser gelang, Indies mit ins Boot zu holen, und zweitens sollten kleine Entwickler als wichtiger Teil der deutschen Games-Industrie für die Solidargemeinschaft des GAME gewonnen werden. Das Ziel: den Austausch zwischen etablierten und aufstrebenden Mitspielern zu fördern sowie den Bedürfnissen kleiner Studios mehr Gehör zu verschaffen.

Viele Ideen sind inzwischen generiert, viele Gespräche in- und extern geführt worden. Mit dem Ergebnis, dass sich viele kluge Köpfe der »Task Force Indie«, wie der Arbeitskreis unterdessen heißt, angeschlossen haben. Der Verband hat sich ein neues Gesicht gegeben, und Indies gehören nun fest zu unserer Mitgliedschaft. Wir konnten eine neue, auf die Bedürfnisse und Möglichkeiten der Indies ausgerichtete Mitgliedskategorie im GAME einführen. Für Indies steht die größte deutsche Interessenvertretung mit mittlerweile knapp 100 Unternehmen offen. Ich freue mich, hier aktiv mitgestalten zu können. André Bernhardt



André Bernhardt
ist Mitbegründer
von Target Games.

André Bernhardt ist seit 18 Jahren innerhalb der deutschen Spieleindustrie tätig. Seine Karriere begann bei Sony Computer Entertainment mit dem Release der ersten PlayStation. Danach arbeitete er bei verschiedenen Publishern wie JoWood und Sunflowers, bevor er bei RTL Games anheuerte. Der Schwerpunkt lag dabei auf Konsolenspielen für die seinerzeit relevanten Plattformen wie PS3, Xbox 360, Wii, DS, etc. 2009 wurde er Teil der Free2Play-Revolution, indem er zu Travian Games wechselte. Aufgrund seiner Liebe zu Indie-Produktionen verließ er Travian 2012 und machte sich als »Indie Advisor« selbstständig. In dieser Rolle unterstützt er Entwickler auf der Suche nach einem Publisher und umgekehrt. Darüber hinaus unterrichtet er als Gastdozent an Hochschulen wie der MDH Berlin und der FH Salzburg und hält außerdem auf Fachkonferenzen Vorträge. 2015 gründete er gemeinsam mit Thorsten Unger Target Games.



Spätestens seit »Indie Game: The Movie« und dem Verkauf von »Minecraft« sind Indies auf dem Radar der deutschen Spieleindustrie aufgetaucht.



WARUM STEAM FÜR ENTWICKLER GROSSARTIG IST

Mit Steam hat Valve nicht nur eine digitale Vertriebsplattform geschaffen, um welche die Entwickler heute beinahe nicht mehr herumkommen. Arne von Öhsen von KING Art Games erklärt, wie Steam einem das Leben sonst noch erleichtert.



Arne von Öhsen ist Programmierer bei KING Art Games.

Arne von Öhsen, Cuthulu-Anhänger und Motorrad-Enthusiast, ist seit 2010 beim Bremer Spieleentwickler KING Art Games als Programmierer angestellt. Er ist dort zuständig für die Entwicklung von Kerntechnologien, für die Optimierung der firmeninternen Workflows sowie das Release-Management.



Wenn man als Entwickler digitale Vertriebsplattformen wie zum Beispiel den App Store von Apple, den Google Play Store oder Amazon nutzt, ist man einiges an Kummer gewohnt. Die Backends sind häufig nicht sehr anwenderfreundlich, scheinen »hysterisch gewachsen« zu sein oder sind schlicht veraltet. Ganz anders Steam. Als Entwickler kann man sich kaum ein System vorstellen, das einem nervige Arbeiten eleganter und zuverlässiger abnimmt. Einige von Steams Stärken wollen wir euch hier vorstellen.

Versionen, Versionen, Versionen

Der Aufwand, ein Computerspiel auszuliefern und zu aktualisieren, wird gern unterschätzt. Je nach Sprachversionen, Betriebssystem oder Zusatzfeatures (wie zum Beispiel für eine Collector's Edition) können in Kombination schnell mehrere Dutzend Versionen pro Veröffentlichung entstehen. In der heiklen Schluss- und Release-Phase kann dies dazu führen, dass man tagelang damit beschäftigt ist, Versionen herzustellen und Unmengen an Daten hochzuladen, anstatt dem Spiel den finalen Feinschliff zu verpassen.

Bei der Entwicklung von »The Book of Unwritten Tales 2« kam erschwerend hinzu, dass wir es unseren Unterstützern von Kickstarter ermöglichen wollten, das Spiel so früh wie geht zu spielen, da wir das so gesammelte Feedback in die Entwicklung einfließen lassen wollten. Wir haben uns deshalb entschieden, jedes der fünf Kapitel nacheinander zu veröffentlichen und die gefundenen Probleme zwischen den Veröffentlichungen mit Patches auszubessern. Der Release-Aufwand hat sich dadurch vereinfacht. Wir brauchten also ein robustes System, um unser Spiel auszuliefern und zu patchen.

Ein Spiel auf Steam besteht aus einer Vielzahl von Depots. Depots sind Datenpakete, die bestimmten Bedingungen unterliegen. So ist es etwa möglich, ein Depot pro Sprache, Betriebssystem oder gekauftem DLC einzurichten.

»The Book of Unwritten Tales 2« wurde in fünf Depots unterteilt: in das Hauptdepot, das alle plattformübergreifenden Spieldaten enthält; je ein Depot für die plattformspezifischen Spieldaten von Windows, Mac und Linux; sowie ein Depot mit Bonusinhalten für Kickstarter-Backer und Kunden der Collector's Edition.

Wir bauen das Spiel mithilfe eines Jenkins Build Servers. Diesen konnten wir so einrichten, dass das Hochladen aller Depots mit der aktuellsten Spielversion in einem einzigen, voll automatisierten Arbeitsschritt erledigt wurde. Installiert ein Spieler das Spiel, werden nur jene Depots runtergeladen, die zum System und Account des Benutzers passen. Bei einem Windowsbenutzer, der die Standardversion gekauft hat, werden also nur das Basisdepot und das Windowsdepot heruntergeladen.

Standard Edition

Deutsch			Englisch		
Win	Mac	Linux	Win	Mac	Linux

Collectors Edition

Deutsch			Englisch		
Win	Mac	Linux	Win	Mac	Linux

2

4

12

Weil ein Spiel auf Steam aus verschiedenen Depots besteht, lassen sich leicht viele unterschiedliche Fassungen erstellen, da nur einzelne dieser Depots dann die Unterschiede ausmachen.



KING Art Games sammelte für »The Book of Unwritten Tales 2« gut 171.000 Dollar ein und erreichte seinen Finanzierungsbetrag auf der Plattform von 65.000 Dollar somit spielend. Später half Steam dabei, die verschiedenen Versionen für Backer, QA und so weiter zu verwalten.

Je nach Plattform kann Steam beim Spielstart sicherstellen, dass alle benötigten Abhängigkeiten, wie unter anderem DirectX oder Visual Studio Redistributable, installiert werden. Außerdem kann konfiguriert werden, welche Executable auf welchen Systemen gestartet werden soll, ohne dass der Benutzer darauf Einfluss nehmen muss. Es ist also nicht möglich, dass ein Kunde versehentlich eine 64-Bit-Version des Spiels auf einem 32-Bit-Betriebssystem installiert oder startet.

Updates

Ebenso wichtig wie die fehlerfreie und unkomplizierte (sowie kostenlose) Verbreitung des Spiels ist die Möglichkeit, problemlos Updates auf das Spiel anwenden zu können. Der direkteste Weg, der in der Vergangenheit oft angewendet wurde, ist, alle geänderten Spieldateien zu sammeln und zu einem Patch zusammenzufügen. Der Kunde muss nun über das Update informiert werden und lädt sich dann den Patch manuell herunter. Während der Installation des Patches werden die betroffenen Spieldateien des Kunden mit den neuen Versionen überschrieben.

Diese Methode hat zwei Schwachpunkte: Zum einen muss der Kunde darüber informiert werden, dass es Aktualisierungen gibt und in welcher exakten Reihenfolge diese angewendet werden müssen. Außerdem sind Patches, die nach diesem Prinzip erstellt wurden, sehr groß. Oftmals verwenden moderne Spiele nicht viele kleine Dateien, sondern wenige große Paketdateien. Soll also nur eine einzige, wenige Megabyte umfassende Textur ausgetauscht werden, kann es sein, dass dies

»The Book of Unwritten Tales 2« erschien auch als DRM-freie Version, was für KING Art Games sehr viel Arbeit bedeutete. Einen Launcher und andere Dienste im Hintergrund auf die Beine zu stellen, kostete mehrere Wochen, während Steam die gleichen Arbeitsschritte in nur wenigen Stunden ermöglicht.



Projekt (BouT2) Deploy - Steam

Dieser Build erfordert Parameter:

version	1.0.3
additional_languages	☒ it ☒ es ☒ fr
player_build	Letzter erfolgreicher Build ☒ Nur stabilen Build
episode_01	☒
episode_01_build	Letzter erfolgreicher Build ☒ Nur stabilen Build
episode_02	☒
episode_02_build	Letzter erfolgreicher Build ☒ Nur stabilen Build
episode_03	☒
episode_03_build	Letzter erfolgreicher Build ☒ Nur stabilen Build
episode_04	☒
episode_04_build	Letzter erfolgreicher Build ☒ Nur stabilen Build
episode_05	☒
episode_05_build	Letzter erfolgreicher Build ☐ Nur stabilen Build
password	*****
Build	

KING Art Games nutzt einen Jenkins Build Server, um die Spiele für Steam zusammenzustellen. Damit können alle Depots in einem einzigen, voll automatisierten Arbeitsschritt verarbeitet werden.

nur möglich ist, indem ein Hunderte Texturen umfassendes Texturpaket ausgetauscht wird. Ein Patch, der eigentlich nur wenige Megabyte groß sein sollte, wächst so schnell auf mehrere Gigabyte an. Hätten wir die Updates von »The Book of Unwritten Tales 2« auf diese Art verteilt, wäre ein Patch etwa 80 Prozent der Spielgröße schwer gewesen.

Um den Umfang eines Patches gering zu halten, ist es praktikabler, nur die tatsächlichen Änderungen auszuliefern. Mit Programmen wie xdelta können Dateien binär verglichen werden. Die Differenz der zwei Dateien wird

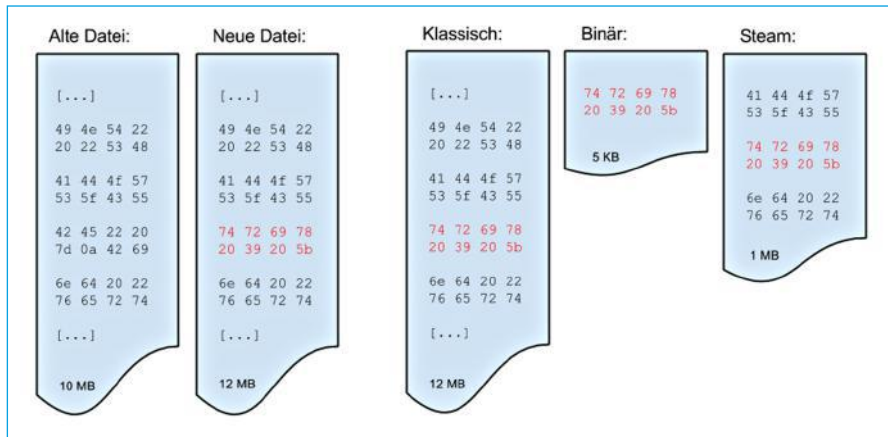
danach in eine Patch-Datei geschrieben. Ein so erstelltes Update ist signifikant kleiner. Damit es angewendet werden kann, ist aber eine spezielle Software erforderlich. Diese Software ändert dann die installierten Spieldateien des Kunden anhand der Patch-Datei. Dies macht es zwingend erforderlich, dass die Spieldateien des Kunden exakt den Dateien entsprechen, die dazu verwendet wurden, den Patch zu erstellen. Wurden die Dateien beschädigt oder geändert, so kann xdelta nicht in die Datei einsteigen und das Update schlägt fehl. Im schlimmsten Fall muss das Spiel neu installiert werden.

Für das Verteilen von Updates über Steam kann man auf ein etabliertes und robustes System zurückgreifen. Große Dateien in den Depots werden von Steam in etwa ein Megabyte große Chunks unterteilt. Wird eine neue Version zu Steam übertragen, vergleicht Steam die Chunks der neuen Version mit denen der alten. Es werden im Anschluss nur jene Chunks zu den Steam-Servern übertragen, die sich tatsächlich geändert haben. Alle übertragenen Versionen werden dann im Steamworks-Backend aufgelistet. Es kann nun entschieden werden, welche Version wann freigeschaltet wird. Auch beim Kunden werden nur die veränderten Chunks der zutreffenden Depots heruntergeladen und ausgetauscht.

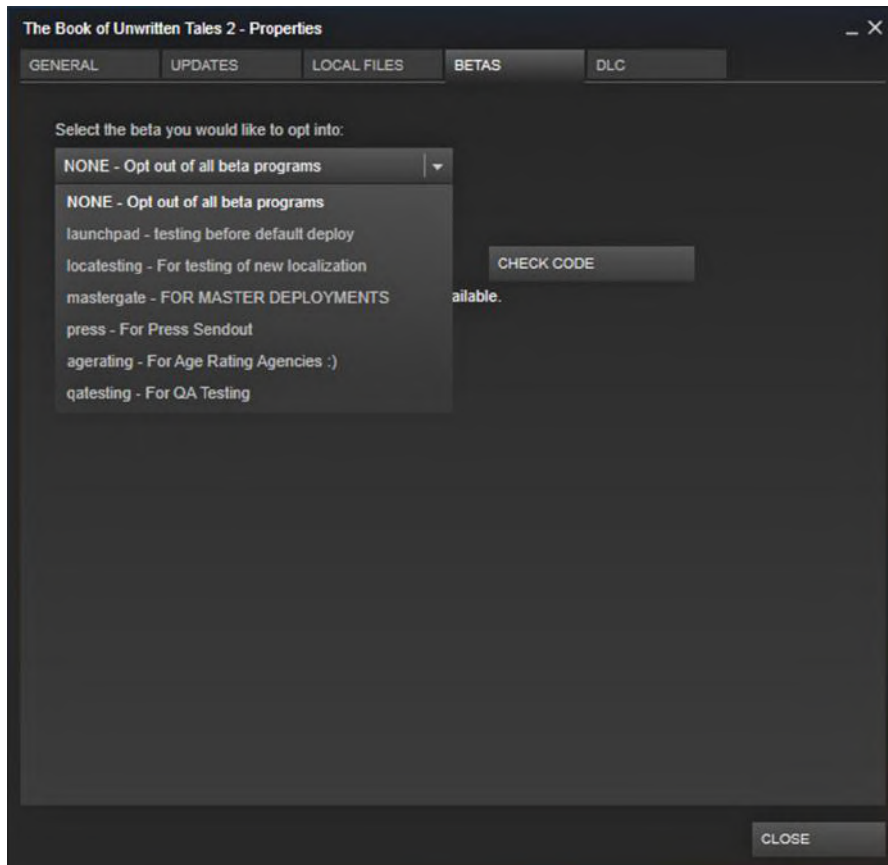
Ein Update bei Steam ist etwas größer als ein Update, das mit xdelta erstellt wurde, kann dafür aber schneller und sicherer angewendet werden. Wurden die Spieldateien verändert oder beschädigt, schlägt zwar auch ein Steam-Update fehl, doch Steam erkennt diese Änderungen und ist in der Lage, beschädigte Spieldateien zu reparieren oder neu herunterzuladen. Bei der Entwicklung von »The Book of Unwritten Tales 2« haben wir etwa 120 Versionen bei Steam hochgeladen. Der Upload einer neuen Version dauerte in der Regel nicht länger als eine Stunde und erforderte dank unseres Build Servers nicht mehr als einen Mausklick.

Im Vergleich: Der andere (aufwändigere) Weg, Patches auszuliefern

Weil wir »The Book of Unwritten Tales 2« auch ohne Steam in einer DRM-freien Version vertreiben wollten und Gigabyte-große Patches keine Option waren, mussten wir die »Nicht-Steam-Version« des Spiels mit einem Launcher ausstatten. Der Launcher, den wir in QT geschrieben haben, wird vor dem Spielstart gestartet, überprüft die Spielversion, lädt gegebenenfalls ein Update herunter und wendet es über xdelta an. Weil unser Update auf drei verschiedenen Betriebssystemen laufen muss, war das Herstellen und Testen des Launchers sehr aufwändig. Es mussten dabei Herausforderungen überwunden werden, wie zum Beispiel das Erhalten von Administratorrechten, um Patches anwenden zu können, oder das Erstellen einer Serverinfrastruktur zum Verteilen der Updates – wochenlange Arbeit, die mit der Steam-Lösung in wenigen Stunden erledigt werden kann.



Steam leistet auch bei Updates eines Spiels gute Dienste, weil nicht wie früher ein komplettes Datenpaket ersetzt werden muss. Stattdessen erkennt Steam, welche Teile des Spiels geändert wurden, und tauscht nur diese aus, was sehr viele Megabytes spart.



Für jedes Spiel lassen sich sogenannte Branches anlegen, die sogar per Passwort geschützt werden können – etwa eine Version für die Presse, eine für die externe QA oder für interne Tests. Diese lassen sich dann gezielt verwalten und mit verschiedenen Versionen versorgen.

Beta-Branches

Steam bietet die Möglichkeit, Beta-Branches anzulegen. Diese Branches erlauben es unter anderem, neue Versionen zuerst von einer kleinen Gruppe testen zu lassen, bevor man sie dann für alle Kunden freischaltet. Da Branches nicht für jedermann sichtbar sind und zusätzlich auch passwortgeschützt sein können, sind sie außerdem der ideale Weg, um Presse- oder QA-Versionen vor der eigentlichen Veröffentlichung zur Verfügung zu stellen.

Das ist für Entwickler wie Presse gleichermaßen bequem. Natürlich können auch die Branches leicht upgedatet werden, die Empfänger haben also immer aktuelle Versionen des Spiels. Wir hatten zeitweise eine Early-Access-Version, eine Preview-Version für die Presse, eine QA-Version für die externe Qualitätssicherung und eine interne Testversion gleichzeitig auf Steam. Die Rechteverwaltung funktioniert problemlos, und selbst mit so vielen verschiedenen Versionen behält man den Überblick.

Achievements und Bestenlisten

Es gibt kaum noch Spiele ohne Achievements oder Bestenlisten. Steam bietet eine API an, über die es sehr einfach ist, diese Features in ein Spiel einzubinden. Achievements und Statistiken werden so im Steamworks Backend definiert und können dann über einfache Calls im Spielcode freigeschaltet oder aktualisiert werden. Über Achievements, die im normalen Spielverlauf freigeschaltet werden, kann außerdem auf der globalen Achievement-Seite von Steam ein guter Einblick darüber gewonnen werden, wie viele Spieler welchen Anteil des Contents vom Spiel erspielt haben.

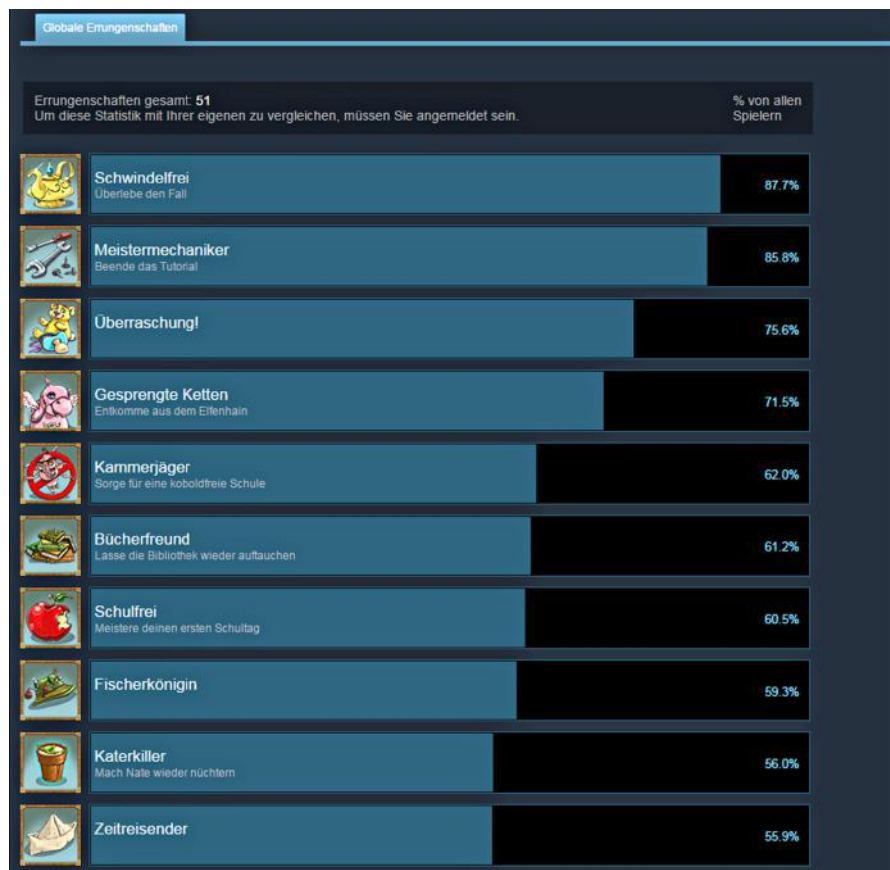
Cloud-Savegames

Das Synchronisieren von Savegames auf verschiedenen Systemen des Benutzers ist eine weitere, nützliche Funktion der Steam-API. Nachdem im Steamworks-Backend eine maximale Anzahl an Dateien und Speicherplatz definiert wurden, werden Savegames, die durch die API gelesen und geschrieben werden, nach dem Beenden des Spiels automatisch mit der Steam-Cloud synchronisiert.

Statistiken und User-Verhalten

Ein spannender Bereich, mit dem wir uns zukünftig sicherlich noch auseinandersetzen werden, ist die Möglichkeit, per Steam Daten über das Spielerverhalten zu gewinnen. Wir sehen den Anwendungsfall nicht im Ausspionieren unschuldiger Spieler, sondern im Tracking von Testerverhalten während der Entwicklung: Wie viele Tester haben wo im Level Schwierigkeiten? Welches Item wird sofort gefunden, welches wird ständig übersehen?

Steam bietet einige spannende Ansätze, relativ leicht an solche Daten heranzukommen und diese uns Entwicklern zur Verfügung zu stellen. Im Steamworks-Backend können dafür



Über die Achievements in Steam können Entwickler leicht den Fortschritt beobachten, den die Spieler (zum Beispiel während Beta- und QA-Phasen) machen, was schnell zu Erkenntnissen für das Balancing führt.

die Statistiken angelegt werden, die getrackt werden sollen, wie etwa getötete Gegner oder erhaltene Items. Diese Werte müssen dann, sobald das entsprechende Ereignis eingetreten ist, über die API im Spielcode aktualisiert oder inkrementiert werden.

Und, und, und ...

Außerdem bietet die Steam-API Lösungen für Matchmaking, Voice-over-IP, Anti-Cheat oder DRM an. Ist man bereit, sein Spiel exklusiv über Steam zu verbreiten, kann die Steam-API Entwicklungszeit und Kosten für Middleware sparen. Die API wird von Valve in C++ ausgeliefert, der Open-Source-Wrapper Steamworks.NET macht sie aber auch problemlos in .NET-Spielen (etwa Unity) nutzbar.

Fazit

Während der Entwicklung von »The Book of Unwritten Tales 2« hat uns Steam unsere Arbeit erheblich erleichtert. Steam ist den anderen Lösungen in puncto Technik, Anwenderfreundlichkeit und Komfort meilenweit voraus. Als Multiplattformentwickler und Antimonopolist können und wollen wir nicht exklusiv auf Steam setzen. Aber als nützliche Unterstützung während der Entwicklung und als die komfortabelste Vertriebsmöglichkeit haben wir Steam während unserer letzten Projekte schätzen gelernt.

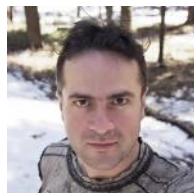
Arne von Ohsen





HOW TO BE SUCCESSFUL ON STEAM

Sergey Galyonkin from Steam Spy explains what influences sale numbers on Steam and what kind of mistakes and biases are common to hinder a game's success on Valve's platform.



Sergey Galyonkin is a games addict and operator of steamspy.com.

Sergey runs a blog about the gaming industry (in Russian) and co-hosts a podcast called »How games are made« with his friend Michael Kuzmin. Sergey also wrote a book called »Marketing videogames« (in Russian) and operates the website steamspy.com, a little side project of him. Steam Spy automatically gathers data from Steam user profiles, analyzes it and presents the results in simple, yet beautiful, manner. Steam Spy is designed to be helpful for indie developers, journalists, students and all parties interested in PC gaming and its current state of affairs.

The information in this article were gathered with the help of the tool Steam Spy that I've developed. Steam Spy acts a bit like an automated exit poll – it queries random user profiles to then extrapolate this data and create a whole picture. This method has its limitations, but it is pretty accurate for owners' data and usage patterns.

Geography

Let me start by saying that there is a lot of stereotypes about what people from different countries are playing and what genres they do prefer. Most of those stereotypes, as it turns out, are true, at least when it comes to purchasing and usage patterns.

USA

For example, US is the biggest country on Steam with roughly 22.5 million users (18 percent of the Steam audience) and as you would expect, Americans love their open-world games and shooters. Or open-world shooters (like »Borderlands 2« or »Grand Theft Auto V«) – they're the best.

That is not surprising and you wouldn't discount USA as a market anyway – it is too big, too rich and too loud.

But the US isn't the only market and, sometimes, the US should not be your first choice of the markets at all.

Europe

Let's talk about Russia. For a long time Russia was most popular for piracy and Russians still pirate a lot of games. Russia has 13.6 million users on Steam (11 percent of the Steam audience), but you would not know it if your games are not localized. For a typical game that does not have Russian localization or at least Russian pricing, you would see only 1.7 percent of your sales coming from this country – negligible, right?

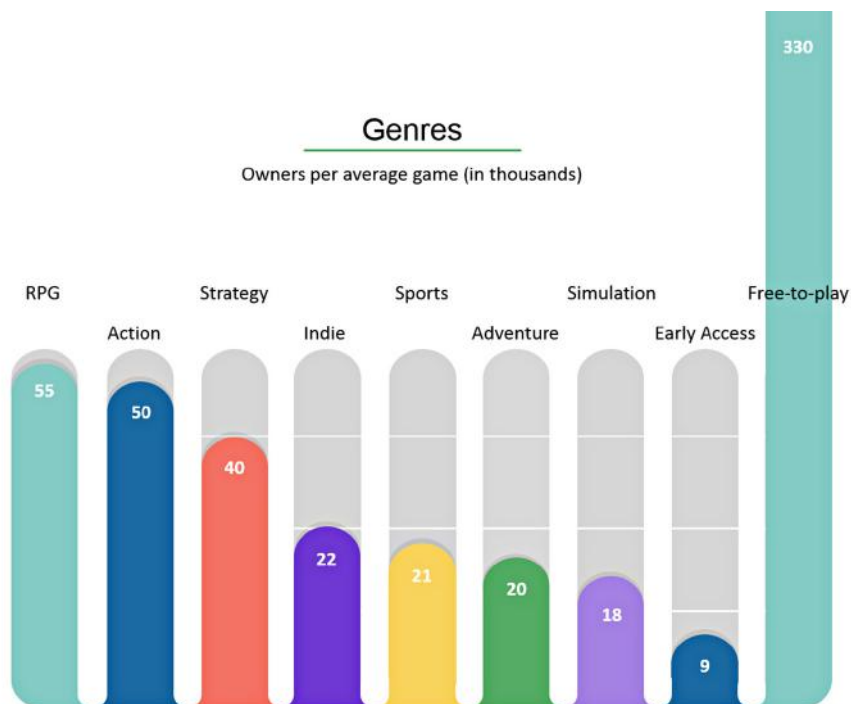
But if you do localize your game and offer a cheaper price (just as Valve suggests), you will see that the number of Russian players rises significantly, easily up to 6 percent. Add a viable multiplayer mode and you are looking at 15 percent of Russian users – they do love multiplayer games. Especially if the game is Free2Play – then Russians will play it like no other with up to 22 percent of players coming from Russia. »Dota 2« players know that, obviously.

When it comes to game themes, Russians love World War 2 and post-nuclear apocalypse, but other than that they're not really different from your typical Western gamer.

I would state that Germans are more different when it comes to game themes. If you have a game with »Simulator« in its title, then expect a lot of people from Germany (up to 15 percent) to buy your game – and prepare by localizing your game and your website. It is not limited to weird games like »Farming

Genres

Owners per average game (in thousands)



Choosing the right genre can be vital but not only the key to success. Sometimes a long ignored niche genre can be the best choice.

Simulator« or »EuroTruck Simulator«, they also love simulations in other genres, like first-person shooters (e.g. »ArMA 3«) or city-building games (e.g. »Cities: Skylines«). Germans also are the most active users on Steam with 92 percent of the existing accounts from Germany being active since 2009.

Never discount the United Kingdom: Despite the UK being a small country, British users are playing and paying the most, with an average Steam user spending 734 hours and having 41 game in his library (twice as much as the average Steam user in general). So as a market for Steam games the UK is bigger than Russia despite having significantly less users. They also seem to love football, which is kind of obvious, right?

Asia

China, on the other hand, is our current number one when it comes to piracy, with an average gamer only having six games in his library and most of these games are free to play. But, like Russia about ten years ago, China is slowly turning into a viable market for paid games. »Grand Theft Auto V« was recently released on Steam with Chinese localization and special pricing (29.99 dollars versus the usual pricing of 59.99 dollars) and managed to gather 11 percent of its users from China – that means 225,000 copies of the game were sold through Steam in China alone making about 6.75 million dollars in gross sales.

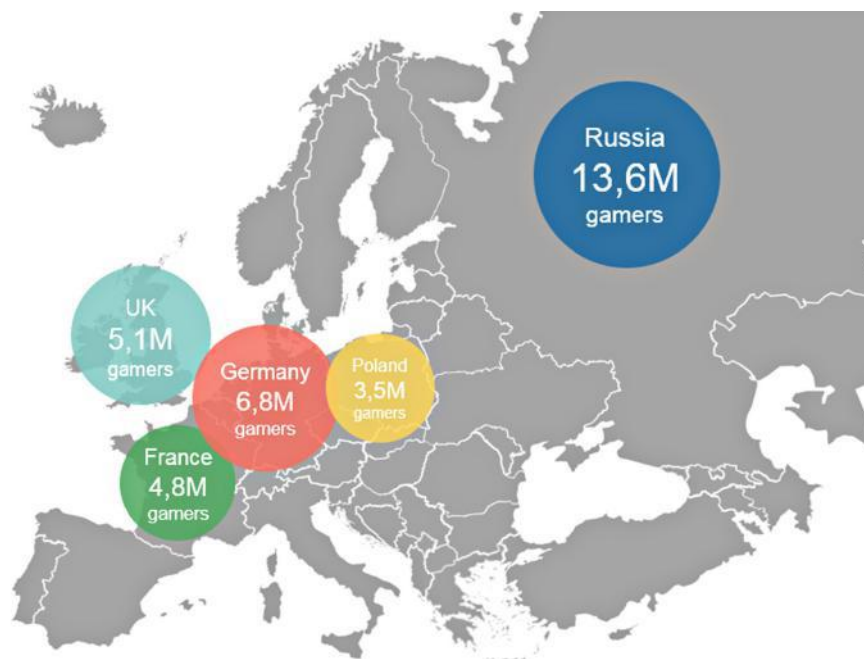
Of course, as Russians before, Chinese gamers are unlikely to buy just any game – the market for paid games is still pretty much hit-driven and games should have a strong multiplayer component so users would have an incentive to buy it instead of pirating it. But it is possible to make money from the Chinese market without going Free2Play.

Japan is often regarded as a console market with PC games being just a niche. It's true to some extent – only 1 percent of Steam gamers are coming from Japan. But add zombies into your game and suddenly you're looking at 4.5 percent of Japanese players, even before proper localization! So if your game has zombies, don't forget about a Japanese localization and some marketing there.

Anyway, what I'm trying to say is: do your research before announcing your game. You might find unexpected audiences for your game in some smaller countries. Never limit yourself to an English-speaking audience – while it is big without question, it is still less than a third of all Steam users.

Sales and genres

I will start by saying that Steam's own genre categories aren't really useful, because a lot of games have to be assigned to several genres and the list is limited. User tags on the other hand are a bit too vague to be used for analysis, but much more useful when you're trying to find a game to play.



This chart shows how big European markets are on Steam compared to each other – Russia by far has the highest amount of players.

The average game on Steam does not sell really well, at least not since 2012. While we all heard about breakdown successes of some big titles, it doesn't apply to your average title. »The Elder Scrolls: Skyrim« has about 8.6 million users on Steam, »Grand Theft Auto V« sold about 2.2 million copies in less than a months, but how does the average game on Steam perform? The answer is 32,000 copies.

It is a bit better if you check the genre distribution with RPGs selling 55,000 copies on an average while Early Access games (that is a genre by Steam's definition) only sell around 9,000 copies. And while free to play games look lucrative with 330,000 owners on an average, when you consider a rather optimistic conversion rate of 10 percent, you will end up with about 33,000 paying users – not really different from the average paid game on Steam.

This saturation also leads to a decrease for the average price for more popular genres. Indie games as the cheapest ones with an average price of 10 dollars, while paid MMOs manage to stay in the area of 20 dollars.

Of course it was not always like this. Until 2012 Steam's growth was matched with a slowly growing catalog of games and that led to gradually increasing average sales. But that stopped when Valve introduced an easier way for Indies – the snowballing quantity of games (some of them of dubious quality) led to a dramatic decrease in average sales. It is no longer enough to just launch your game on Steam to sell something. Now you have to do PR, marketing, support and all the other stuff that only big companies were paying attention to before.

Of course marketing and PR is not a guarantee for a success. While it will obviously amplify your game sales tenfold, it won't mean

Try Steam Spy yourself

You can try out Steam Spy and its insights at its official website:

<http://steamspy.com>

Beside of the website Steam Spy even offers an API with which you can request specific data easily and develop your own applications:

<http://steamspy.com/api.php>

much if the game is not performing really well on the word-of-mouth alone.

We're also seeing that Kickstarter while being a really good way to fund original games, is not very useful for establishing a series. Most sequels of games from Kickstarter fail to achieve comparable sales despite receiving better scores from both gamers and press alike.

Common misconceptions

I would like to address some of the common mistakes about launching games on Steam that I have encountered in the last few years.

Mobile gaming

Recent studies into mobile show that there are around 4.88 billion mobile users with access to game-enabled devices – that is an incredible market compared to the rather tiny number of 125 million Steam users.

But: There are 1.6 million applications on iOS alone, competing for the same audience and attention. And while games on an average make 2,800 dollars per year, this amount of money is not evenly distributed – the aver-

age game on iOS makes in face no money at all. Seriously.

The 32,000 copies that a game does sell on an average on Steam do not sound so bad now, right?

Games for women

While female gamers constitute a large part of the PC gaming audience (49 to 51 percent depending on the research), they are less likely to be on Steam. According to studies by Alexa Internet around 18 percent of the users who are visiting Steam's homepage are female and the actual number of female gamers on Steam might be even smaller.

Discussing reasons for this is outside of this article's scope, but games aimed at a female audience are less likely to succeed on Steam than on other more inclusive platforms like for example Facebook or iOS.

Early Access

For months Early Access was lauded by developers as having two launch events on Steam – one for an incomplete version of the game and the second one for the final version.

Well, I've got bad news for you: Every game still has only one launch event and if you're going to release it in Early Access this date it will be – there is no second chance.

Of course some games might be able to get the attention from the media and gamers alike the second time, but even heavy-hitters like Double Fine's »Broken Age« often fail here.

Clone a good game

While cloning might be a viable strategy on mobile (to be honest: it is not), it is rarely successful on Steam, where your audience is generally better educated about games from their favorite genres.

You could and should, however, improve on other game mechanics and visuals, without just doing a clone.

Ignoring the reviews

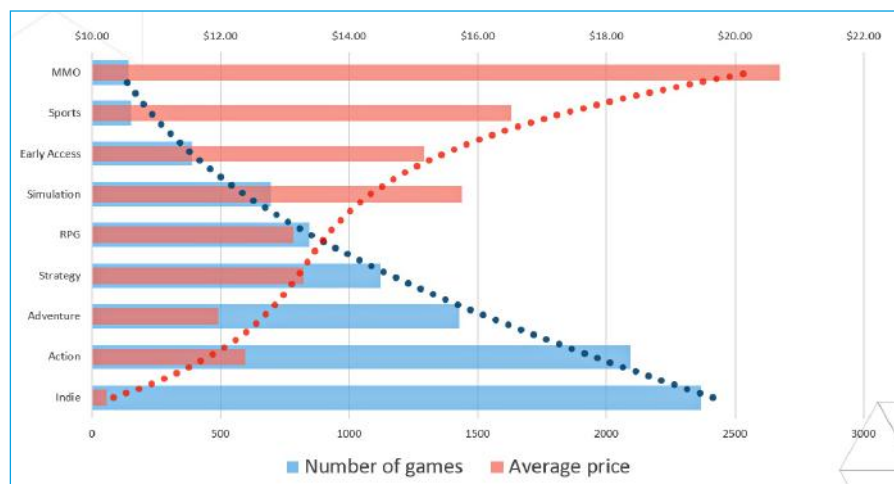
It is a common misconception amongst game developers to ignore gamers or press when they don't agree with you or with your vision. A good game is good regardless of what everyone says, right?

Well, yes, it is true, but good reviews are usually correlate with good sales, meaning that if the game is good, people will recommend it and when the game is bad, they will advise against it.

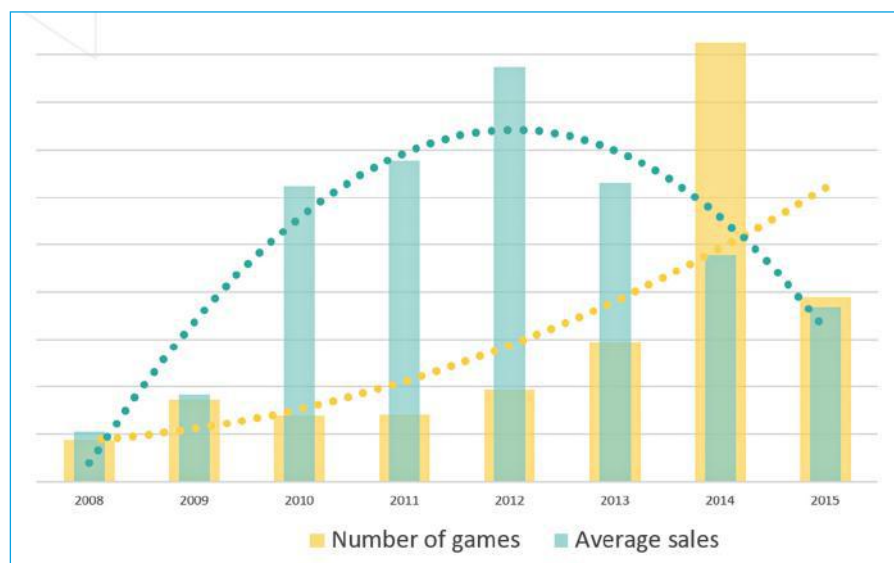
But on the other hand: Even bad reviews are better than complete silence surrounding your game.

Doom and gloom and a way to succeed

While it all might sound pretty grim – low average sales, no market for female-friendly games, just one chance to do a launch correctly – well, it's not as bad as it might sound. Dozens of independent games succeed on Steam each year and your game might be one of them.



The saturation of the market leads to a decrease for the average price for more popular genres.



Until 2012 the slowly growing catalog of games on Steam led to evenly increasing sales. When Valve open Steam for Indies this changed.

Let's talk about several successful games of 2015 for example. While they're all different, they all share some common characteristics worth noting.

Find the right niche

This one is easier said than done, but it is the only way to stand out in a crowded market like Steam, where almost 2,000 games get released every year.

Your expertise is probably limited to handful of genres anyway, so look into what hasn't been done lately. A couple of good examples: »Darkest Dungeons«, which is not a retro rogue-like (that's something these days) and »Cities: Skylines«, which is basically »Sim City« and we haven't seen a good »Sim City« game in decades.

You could check user tags statistics for this on Steam Spy to research which niches were potentially not cared of for a long time. For example the classic base-building RTS genre is relatively orphaned right now.

Talk about your game before it gets released

Almost every successful Steam game had some press coverage long before it was released – Early Access or not. Being successfully backed on Kickstarter obviously helps, but even if your game was funded without external help, you should not hide it from press and gamers.

There is a popular notion in marketing that a person needs to hear the name of your product at least five times before he or she can even remember it – and you probably won't be able to reach that for every potential consumer in a matter of hours after launching your game.

So, talk early, talk a lot, do marketing and PR long before you plan to release your game.

Localize

Not everyone knows English well enough to enjoy a game. In certain countries people will not even play a game if it is not fully dubbed in their language – blame the education or traditions, but that's the way it is.

Always do research for regional markets for your game, so you won't waste money on localization efforts on people that won't like your product anyway. Instead focus on countries where gamers are more likely to play your title. Switch the language in your zombie game from Italian to Japanese and you'll see your audience respond positively.

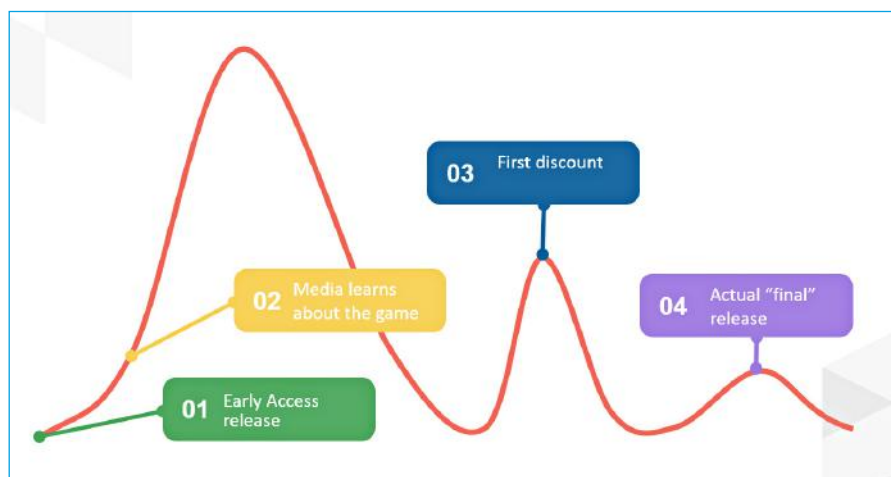
Don't discount too much

You can't raise the price of your game on Steam. Well, technically you can, but it won't make consumers happy, so the maximum you can do is cancelling a temporary launch discount.

So, don't set your launch price too low. You can always discount it later, but if you're starting with a low price, you're basically leaving money on a table from the beginning.

Early Access is your launch, treat it this way

You only launch your game once, so even if you are releasing an Early Access game, it should be more or less feature-complete and playable from the beginning till the end without any game breaking bugs.



Attention for Early Access games is depending on media coverage and discounts - the final launch of the game marks the end of the cycle.



The higher a game is rated in reviews (be it by gamers or press) the higher sale numbers will usually be.

Check out »Darkest Dungeons« – despite missing some content on launch, the game itself is ready (and awesome) even with the Early Access label on it.

I know it sounds unfair, but if you're selling your Early Access game next to already released games, you have to deal with it.

Support your game

While games tend to generate most of their profits in the first months, several games stand out here because developers cared enough to support their games with new content (free or paid) and gamers responded.

Additional support doesn't just generate sales because of sheer goodwill, it is also a great way to increase visibility both on Steam and in gaming media.

Paradox Interactive (»Cities: Skylines«) is especially good at this, learn from them.

Summary

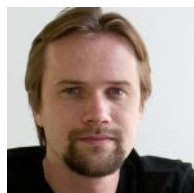
Steam is an awesome platform to sell your games and despite many shortcomings, it is still the most game developer friendly environment to work with. Just remember some basic rules: research your competition and the market that you are aiming for, promote your game before the release, localize, don't discount and support your users.

With a good game, hard work and a bit of luck you'll do fine on Steam. **Sergey Galyonkin**



A LOOK INTO STEAM REVIEWS DATA

When Valve added the ability for users to review games on the Steam platform, it opened an all new aspect to how the games are perceived there and a new parameter for developers to understand to help their game perform as well as possible.



Thomas Bideaux
is CEO and Co-Founder
of ICO Partners.

Thomas is a consultant for the online games industry and has been providing business intelligence to its clients for many years. In parallel, he has been developing a fascination for the crowd funding phenomenon and is regular commentator on its impact for the game industry.

Steam's review system is fairly simple: users who «own» the game can leave a text-review of the game, associated by either a thumbs up or a thumbs down opinion on the game. The review is then listed on the store page of the game and other users can thumb it down or up, depending on how useful they find it (they can also tag it as funny but I won't cover this).

Reviews that have been considered the most useful are listed higher on the page, and they also specify the user's time spent on the game, as an additional contextual information to consider when reading it.

More prominent on the page though is the game »Steam Review Score« which is based on

the percentage of positive reviews the game has received overall.

Using the data gathered through the Steam API, I went and explored games and their Steam Review Scores, and I tried to look at specific take aways when considering your own game.

Important Data Points

One important thing to understand first is the qualifier used on your Steam store is based on your percentage of positive reviews as follow and qualified from Negative to Mixed to Positive. The number of reviews will then also factor in qualifying the positive or negative trend of the reviews.

To be positive, you need your Steam Review Score to be about 70 per cent. The higher the better.

The review score is based on the percentage of positive reviews a game has received overall.

The Steam Rating is given a lot of presence not just on Steam, but also on search engine results pages.

If you fall below 40 per cent, your game reviews will be deemed as negative.

That's Steam own qualifier, but you might want to know how well you are doing compared to the other games scores.

Looking at the median Steam Review Score for launched games is very interesting. Half of the games on Steam are reviewed positively on Steam 81 per cent of the time. If your game is below that Steam Review Score of 81 per cent, it is in the bottom half of Steam. While the Steam community is often portrayed as ruthless and very unforgiving, this relatively high percentage is actually surprising. I suspect it mostly comes from the fact that a lot of players will more likely review a game they love than a game they hate (or so I would hope).

Interestingly as well, despite sometimes a bad press, Early Access games are on average getting more positive reviews. Lots of reasons for this, but mostly, I expect players who are joining an Early Access to be more tolerant of a game flaws and rate more generously their experience.

The distribution is done across thousands of games, and there are some other considerations that can be taken into account when considering the distribution of positive reviews.

Steam Review Score per Genre

Table 1 shows the top ten most represented genres on Steam (with the most games) as defined by the user tags.

Platformers and Puzzle games are both rated higher than the median, the Simulation genre, while being the fifth most common genre on the platform, has an average Steam Review Score of 70.9 per cent, more than 4 points below the average.

Anecdotally, the only game I found with the Overwhelmingly Negative qualifier was Airport Simulator 2014.

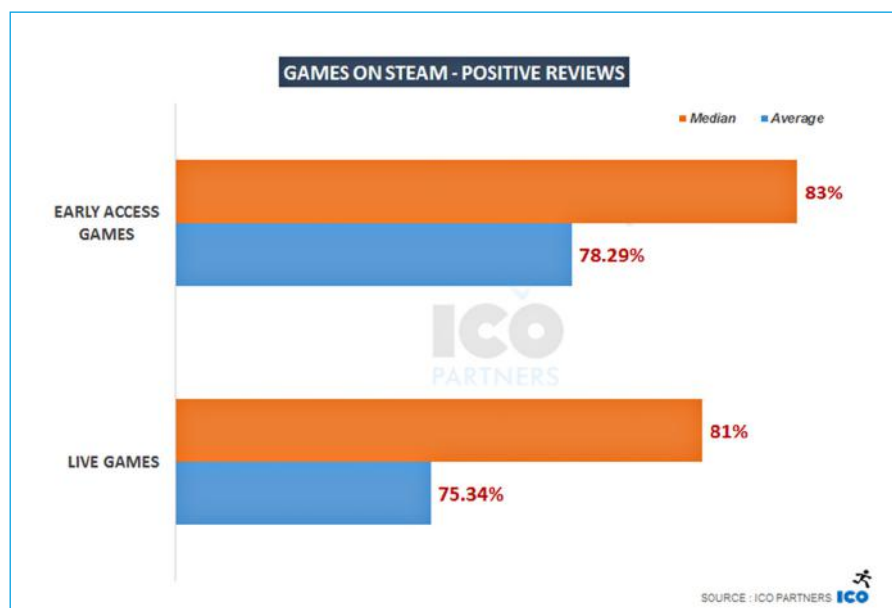
Historically, Valve is an FPS developer. A lot of the initial success of the platform was built on the popularity of their own games in that genre. Both FPS and Shooters categories (often overlapping tags), have their average Steam Review Score above the overall average. There is probably an interesting ecosystem component at play on the platform. With a strong FPS fanbase, they might be more enthusiastic about the core genre they like.

Steam Review Score per User Tags

Looking into another segmentation for the games, I picked some of the user tags that had the best average Score Review Score associated with their games for **Table 2**.

These tags can be seen in two ways: features added to games that are very well appreciated or features that are added to high quality games.

For instance, a great soundtrack is probably more common in games with larger budget and more capacity to invest in the talent to make it happen.



Looking at the median review score, half of the games on Steam are reviewed positively 81 per cent of the time. Early Access games are getting more positive views on average.

Still, there are some interesting insights. One could have expected that a game deemed difficult would be more harshly rated. But on Steam, games with the »Difficult« tag are very positively rated (there are 283 games rated as such if you are wondering).

In the same vein, Story Rich and Episodic games (two complementary tags) good average Steam Review Score are an excellent sign for narrative games on the platform.

Lastly, it seems that games with pixel arts are not suffering from the art direction choice.

Table 3 presents another selection of average scores based on tags, this time looking at the lower end of the spectrum.

I mentioned earlier the good performance of Shooters and FPS games, but it seems that this ecosystem of FPS fans don't like On-Rails Shooter (or they are just all horrible, that's not impossible).

GAME GENRE	AVERAGE STEAM REVIEW SCORE
ACTION	76.6%
ADVENTURE	78.6%
STRATEGY	74.6%
RPG	76.8%
SIMULATION	70.9%
SHOOTER	79.4%
PUZZLE	82.8%
FPS	78.0%
PLATFORMER	81.3%
POINT & CLICK	79.3%
SANDBOX	79.1%
HORROR	76.5%

SOURCE : ICO PARTNERS

Table 1 A lot of Steam's initial success was built on the popularity of their own games in the FPS genre. Both FPS and Shooters categories (often overlapping tags), have their average Steam Review Score above the overall average.

Table 2 Contrary to expectations, games with the »Difficult« tag are very positively rated on Steam.

GAME TOP TAGS	AVERAGE STEAM REVIEW SCORE
4 PLAYER LOCAL	89.8%
MODDABLE	88.6%
GREAT SOUNDTRACK	88.1%
REPLAY VALUE	87.7%
EPISODIC	87.4%
STORY RICH	87.1%
DIFFICULT	85.6%
PIXEL GRAPHICS	84.8%
GORE	84.1%
LOCAL CO-OP	83.1%

SOURCE: ICO PARTNERS ICO

Table 3 While Free2Play games score below the overall average Steam Review Score, it is not by a too significant margin.

GAME WORST TAGS	AVERAGE STEAM REVIEW SCORE
ON-RAILS SHOOTER	54.9%
FOOTBALL	59.4%
AGRICULTURE	66.2%
SNIPER	66.7%
TRAINS	69.0%
FLIGHT	69.3%
MANAGEMENT	71.3%
HIDDEN OBJECT	71.5%
SPORTS	71.5%
FREE TO PLAY	72.9%

SOURCE: ICO PARTNERS ICO

The poor average score of Simulators is also apparent here, with Agriculture, Trains and Flight tags whose games are on average positively rated below 70 per cent of the time.

Finally, I specifically looked for Free2Play games, as they are often decried publically. While they score below the overall average Steam Review Score, it is not by a too significant margin.

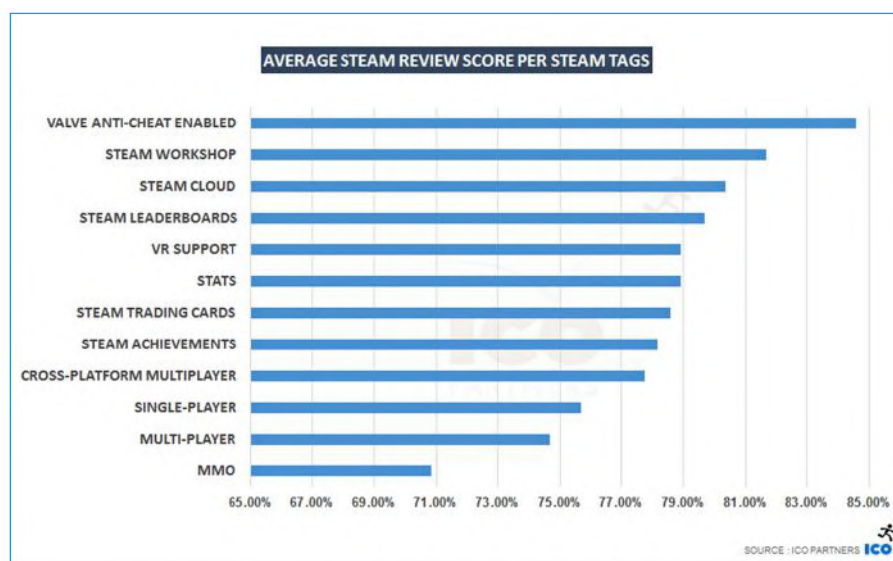


Table 4 The high Steam Workshop value is echoing the high score for game tagged as moddable earlier.

Steam Review Score per Steam Tags

Steam Tags are functional tags in the Steam backend. They often describe the game in a more functional way than user tags, and those are set by the developers.

Table 4 is a relevant selection of those tags and the average score their games have.

The high Steam Workshop value is echoing the high score for game tagged as moddable earlier. There is a significant amount of extra work required to make a game properly supported in the Steam Workshop. Like for the Great Soundtrack, I suspect that games that integrate Steam Workshop just have a generally better budget, to allow for these things, and representative of higher productions in general. This said, there is an undeniable correlation between the better review score and the moddability.

At the other end of the spectrum, MMO are really poorly rated in comparison. As MMOs need to be designed to entertain over very long period of times, the quality bar is set significantly higher than for other games where a few hours of fun are deemed a success.

How can you influence your Steam Review Score

There are number of things a developer can do to optimize its review score.

- 1. Get the message right:** Many negative reviews come from disappointed users. Make sure you message the right game to your audience. It is often tempting to go over the top on what you promise in your game, but those promises will become the expectations of the users. If you don't meet them, they will review your game negatively.
- 2. Get positive reviews early on:** A game flagged as »Negative« will influence incoming users. They will be more susceptible to see the game for worse than it is. The flipside is that if you can get early positive reviews, you support a positive trend on your reviews. This is the clear benefit of Early Access. By going this way, with a game that's already enjoyable and somewhat polished, you will get positive reviews that will help when you transition to Live. It is also a good thing to encourage players to review your game (without being specific on how they should review it). The players that you will likely reach are the ones engaged with the game. And if they are engaged, it is likely they appreciate it.

3. Discuss with authors of negative reviews:

You have the option to reply to a review and provide your own context about it. Many negative reviews are very short »Doesn't run on my machine«. If you can find what the issue was for the user, you get a shot of turning a negative review into a positive one, for a double whammy effect. Even getting a negative review to be removed is helping tremendously.

Thomas Bideaux

MODDING IN CITIES: SKYLINES WITH STEAM WORKSHOP

When Paradox Interactive started to work on their city builder the team decided to integrate support for Steam Workshop from the beginning on: the best choice they could take as their experience shows.

When we started working on »Cities: Skylines«, we knew from the start that modding would have to be an integral part of the product. »Cities: Skylines« was the first game from Paradox Interactive where we went up against an established AAA-title in a non-niche genre, and we simply lacked the resources to compete in terms of assets and production value.

This made modding a target of one of the game's most obvious flaws, and we implemented it early as one of the demands of the product. We also decided that in order to keep costs low and to let Colossal Order, the studio that developed »Cities: Skylines«, focus on doing what they do best so that we would build the whole game from start around Steam Workshop.

Integrating modding into the game would always be a bit of a challenge, Unity as an engine has never been the most mod-friendly, but Colossal proposed great solutions on how to make it more accessible. We also decided to design native tools for importing and exporting 3D objects into the game to be used in the asset editor for easy creating and sharing of assets.

While we had hoped that the modding tools of »Cities: Skylines« would be popular, nothing prepared us for the reception they got. Just a few days after release, we had thousands of pieces of content on Steam Workshop, and it soon became clear us that the majority of players used this option to enhance the gameplay.

The rest of this article will ask why it was so successful and what lessons can be learned from our experiences.

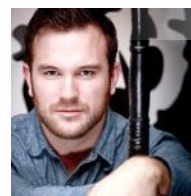
Modding as philosophy

In many ways, modding in game design boils down to a philosophy; do you sell a finished set experience to the player or do you create the

so often spoken about sandbox where players shape and create their own experience?

The AAA industry has always been dominated by the former thinking, which is natural. They have enormous budgets but also need to sell enormous amounts of copies which always might seem to be easier the less you demand of the player. Paradox Interactive on the other hand grew up way out on the other side of the scale, creating vast and rich sandboxes on tight budgets. Our experiments with post-release patching and DLC often rose out of necessity rather than choice but it proved the strength of digitally distributed content.

Already with the original »Europa Universalis« the first game-changing mods started to arrive, and I remember playing them myself as a 16 year-old. It was the start of Paradox's relationship with modding, and over the years several games have been released since based on mods made by the community.

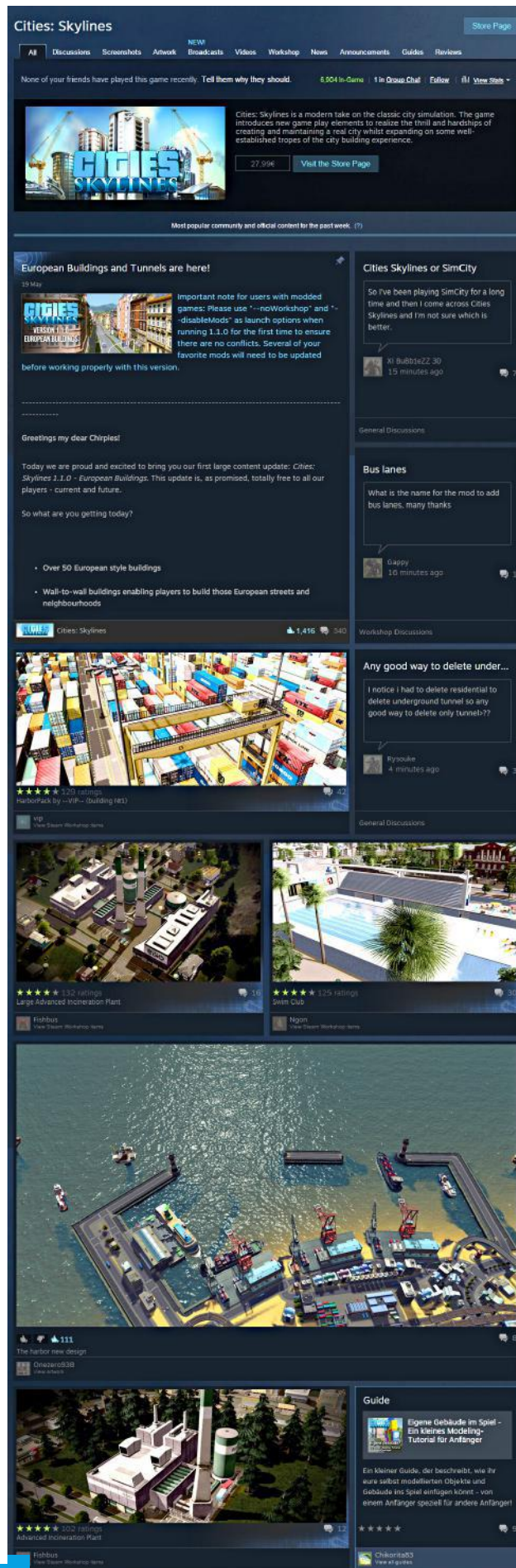


Jakob Munthe
is Brand Manager
at Paradox Interactive.

Jakob joined Paradox Interactive in 2012 where he first worked as DLC Producer. Shortly after that he became Brand Manager. Today he is responsible for »Cities: Skylines« and for Steam Workshop in particular, but also worked on games like »Hearts of Iron 4« before.



»Cities: Skylines« is a city builder that is very much comparable to »SimCity« - the only difference is that the team that worked on the game was way smaller and therefore the game can't compete content-wise. Steam Workshop was a big help there.



»Cities: Skylines« was released in March 2015 and about two months later the community on Steam has created more than 40,000 mods of all kind - an impressive number and a lot of stuff to scroll through.

When we started working on »Cities: Skylines« it was obvious that not only did we want to continue this tradition but we also wanted to push it further than before. Instead of seeing modding as something added later to the game as a possibility for a few dedicated users, we wanted to integrate it into the heart of the game.

Content vs. Systems

One of the main reasons behind the choice to prioritize modding tools was the choice between content and systems. The core team at Colossal Order started off with only 9 people, and despite growing to 13 during the project and the use of 3rd party graphic studios, there were always hard choices to be made. Our earlier experience with games like »Europa Universalis« and »Cities in Motion 2« told us that even a fairly small player base will create quite an extensive amount of content if you use Steam Workshop. In case of »Cities in Motion 2«, a game that in many ways exemplifies niche, there are at the time of this writing around 1,000 pieces of content on Steam Workshop.

Smaller teams, especially Indies, focusing on systems ahead of content is of course nothing new, but it is maybe not commonly understood how much a file sharing platform such as Steam Workshop can help a game to overcome limitations in content. Games such as city builders are based on the people's love to create, and if you give them good tools and good possibilities to share even a fairly small user base will quickly start creating assets. For a game with a large active community, the sky is the limit, often generating full new game experiences.

This part of modding can often seem problematic to the developer; the more you unlock for the user the harder it will be for the developer to have control of the experience. The developer will often want to focus more resources on bringing their own ideas into the game than facilitating the same opportunities for the players. This is only natural, most developers want to make game experiences, and not feel like a glorified engine maker and want to maintain control of the creative vision of the game.

However, there can be a great and unique experience in working close with a community and developing a game together, blurring the line between creator and consumer. The role of the publisher here should often be to try and help the developer to feel secure in focusing resources on the modding tools rather than adding to core game experience.

Steam Workshop in the game

One of the most important decisions regarding the actual implementation we made regarding »Cities: Skylines«, was to design our modding tools and Steam integration from start. Our previous games had the Workshop added at the end of the project, almost as an afterthought. The games were designed to be full experiences

in themselves and modding seen as a boon for the dedicated few that wanted them.

With »Cities: Skylines« we took another route. We designed all tools and implementation from day one and decided to use a design that would have the mods integrated in the core game loop, even using the mod systems for different game modes and options. We also decided not to use an in-game store but instead integrate Steam to make all the systems as close to each other as possible.

Integrating Steam Workshop into the design was fairly straightforward once the decision was made. As mentioned earlier, Unity has never been the most mod-friendly engine, but the folks over at Colossal Order did an amazing job in opening up the engine, building great tools and taking a holistic view on modding in the game design.

One thing that has proved more successful than we expected was how well the community has monitored itself. Despite having thousands of mods that potentially could be harmful (»Cities: Skylines« allows for .dll-modding which can be used to create all forms of malware), no serious problems have aroused. The community quickly arranged for an approval method based on »Reddit«, and still today the amount of flagged content is in the low tens.

Another thing that worked a lot better than expected was the media pickup. Within weeks the mods for the games got more press than the game itself, continuing to gather pick up for the title and generating great buzz. PC Gamer still has a continuously updated article called »Best mods of Cities: Skylines« that keeps the game in players' mind.

Modding post-release

One of the greatest challenges and maybe the only negative surprise of using Steam Workshop with extensive mod support has been with patching and updating the game as it is live. The most popular mods on »Cities: Skylines« have hundreds of thousands of subscribers, which means that if any of those are broken with a patch, severe player reactions can occur.

We are still working on finding a solution around most of the problems. As a game becomes more popular and the modded experience becomes as common as the vanilla experience, the idea of owning the game and the experience becomes blurred. Paradox's line here has been clear, in the end we are the ones promising to deliver a good product to the player, because of that we also need to make sure that we can take final responsibility for said product. With thousands of mods, there is no way for us to ever ensure that all of them work when a patch is deployed, therefore the most straightforward solution is to take responsibility for none of them. That doesn't mean that we will not try to help modders in quickly patching their content, but the responsibility must lie with them.

In the last part of this article I would like to discuss the future of modding, both in »Cities: Skylines« in particular and gaming in general. But before I get into that, let me summarize the experiences with modding in »Cities: Skylines«:

»Cities: Skylines« has been Paradox Interactive's greatest success to date, and while we had high hopes for the product we never expected what came. It cannot be understated how important modding was in making this happen. The game, however good, had some natural limitations due to the size of the team and Steam Workshop was a key factor in overcoming these short comings. I would recommend any publisher or developer to never hesitate in spending time and resources not only on making modding available, but to also try to develop tools and APIs to facilitate for the creators. Even fairly niche games will get an active community, and it will help your product; end of story.

The Future

So what is the future of modding? First of all, let us start with acknowledging that mods have for a long time defined PC gaming, maybe more than people have been willing to admit. Of the five most played games on Steam, three are based on mods. More and more of the PC's bestselling games have added mod support, including titles like »Skyrim« and »GTA V«.

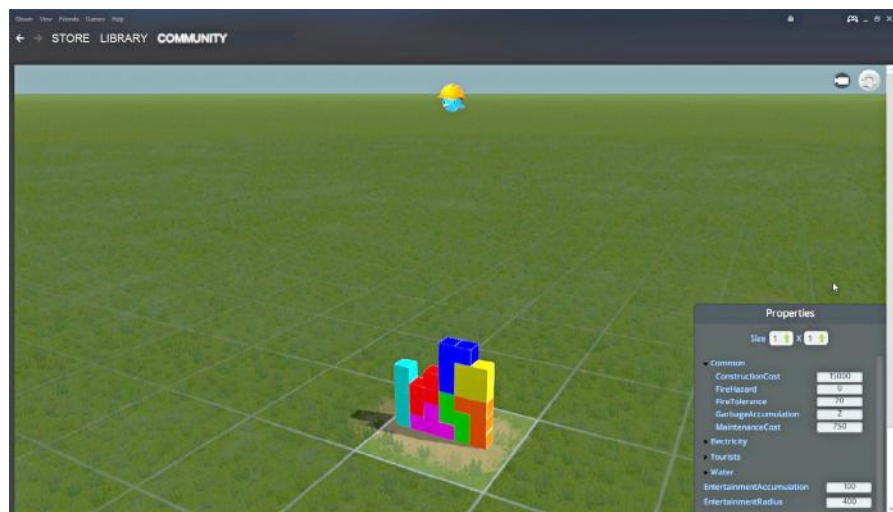
One of the great unresolved issues will be the monetization of mods. The classic way has been making full games out of successful mods, but these will only be for the full mods and not for the smaller content creators that make up the majority of the modders.

Valve has successfully found solutions, »Skyrim« had less luck with theirs. The exact nature of how this will end up remains to be seen, but I am personally convinced that more and more paid-for mods will become the norm. When hundreds of thousands, or even millions, of people use your content it is only fair that some sort revenue flows to you, the modder – the creator.

Jakob Munthe



»Cities: Skylines« features some different themes, like Tropical or European, which have been expanded by modders with real-life buildings, vehicles, etc.



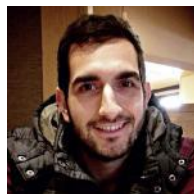
Some players on Steam Workshop published »How to«-guides in which they explain how to create new buildings or vehicles to novice modders.



Making Games 04/2015

FREE WEEKEND PROMOTION OR THE POWER OF FREEBIES

Edge Case Games took their paid Early Access Title and made it free, for keeps, for a long weekend. Here they look at the impact this unusual promotion had on the game as well as the studio.



Sam Faulkner
is Community Manager
at Edge Case Games.

Sam is a Community Manager at Edge Case Games. As such, he splits his time between engaging with and growing the player base and creating content for the studio's various channels. Having recently moved over into the development world after a five-year career as a journalist, he is currently spreading the word about the studio and some of their more unorthodox methods.

Sometimes, the best way to sell a product is to give it away for free. That might sound crazy on first reading, but over here at Edge Case, our recent Steam free weekend promotion seems to have been the catalyst in a successful few months for the studio. It wasn't just a case of throwing the game up on Steam and saying »here you go, help yourself« though. This promotion was months in the making and even with all that planning, mistakes were made and lessons learned.

We've been doing things a little differently over here, stepping away from the usual models of promoting a game, and embracing a fully open development approach was just the first part of that. Giving away *Fractured Space* for a whole five days, with all new players able to keep it forever, was similarly uncharted territory for us. While we are eventually planning on transitioning to a Free2Play model when the game launches in the ever-elusive »1.0« state, we are on a paid Early Access structure currently and opening the gates, if only for a set period of time, was a scary undertaking for us.

Introduction to *Fractured Space*

Just by way of introduction, our game *Fractured Space* is a MOBA-like 5v5 PvP space combat title, featuring huge capital ships duking it out in colossal space battles. Diverse ship designs and a constantly shifting meta give the game an intense tactical edge in amongst the explosions and flying plasma. We're probably about 30 per cent of the way towards being »complete«

at this stage, with the game being constantly tinkered with as new content is added.

Where the traditional behind-closed-doors, publisher-led method of trailing a new game would entail a fairly typical drip-feed to press via a carefully orchestrated PR campaign and structured news beats, we decided to try something a little different. After all, we're just a relatively small team (numbering in the low twenties) operating out of a former bathroom showroom in Guildford – not for us the glass and chrome motherhips of our neighbours EA or Ubisoft. There are legitimate questions about the value in adopting the same practices, and so we chose to open the doors to the public like an Unreal 4-using Willy Wonka.

What is Open Development?

We do things a little differently here at Edge Case, and though we're headed into slightly uncharted waters, the team have gotten fully behind our open development philosophy for this project. We could go on about this for hours, but in a nutshell, it's essentially the process of hiding absolutely nothing from the public as we go through development of *Fractured Space*. Our designers are active on the forums and TeamSpeak servers, the artists live stream their work on Twitch, and our two full-time Community Managers put on several direct interaction events every week. We stream the game every Thursday, taking questions from the playing public, and host a dedicated developer Q&A every other Wednesday to give fans a peek behind the curtain. Well, perhaps it's more appropriate to say there is no curtain,



Fractured Space is a MOBA-like 5-versus-5 PvP space combat game.

as we share pretty much everything we have to show – players can even vote for features in upcoming sprints via our Trello board (a kind of public to-do list/calendar). It's a bit different to the traditional publisher/PR-led behind closed doors approach, but we've thrown ourselves fully into this way of doing things, finding a really rewarding way to work along the way.

Heading into the promotion, we were boasting a manageable (but relatively modest) 1,000 Daily Active Users (DAUs) for Fractured Space, on a pleasing retention rate of around 40 per cent from the previous day. In terms of concurrent users, a pretty vital element for an online-only PvP title, we were peaking between 75-150. This would typically come during our weekly Thursday playtests, during which the community team would stream the game, and our developers would interact with the players over Teamspeak. These sessions would have the team discussing upcoming updates, current issues and proposed features. Early Access was proving a neat fit for our Open Development style (essentially a very transparent and community feedback-driven development ethos, through our forums, streams and an online dev schedule), but what we really needed was more eyeballs on the game to really let it fly.

Obviously, as much as we love our players, we weren't giving away our game purely out of the goodness of our hearts – there were some definite objectives in mind. First and foremost, it was a scalability test. The game's code had been holding up pretty robustly at our current numbers, and the chance to stress test it with a larger pool of users would be of

massive value to the ongoing development of the game. With that said, timing was always going to be key on this one. After all, we are in very Early Access, with the game really still in an Alpha state, so we had to make sure the game was polished enough to show off to vast numbers of new players. Every little piece of data we can Hoover up is useful in one way or another, from player behaviour, to information about match length and the stats of each ship, all go back into designing the game.

Of course, attracting a new flock was always going to be a nice side effect too, especially since without players in a PvP title, there's no game. So with one eye on rolling out our fledgling drafting system ahead of full match-making, we sought to bolster our congregation as much as we could. A rudimentary drafting system was put in place, and extra servers set up in the EU and US to cope with the anticipated increase in traffic. The studio went a little Helm's Deep on that Thursday afternoon.

In a meeting the day before the Steam promotion was about to kick off, we were setting our community goals. The more new players we attracted, the more rewards we would unlock for all those who took part over the weekend. We had put our estimates at around an eight to ten times increase of regular figures, based on similar Early Access titles which had taken the free weekend approach. We decided to set the three unlockables at tiers of five, ten, and 15 thousand new users, and even allowed ourselves to ambitiously talk about our chances of hitting the first milestone within 24 hours – it turns out our projections were pretty far out.



Part of the Open Development philosophy involves sharing artwork on upcoming or potential features, right from the concept stage.



Apart from attracting new players, the free weekend was also a scalability test.

Let Chaos Commence – The Free Weekend Begins

It seems we were a little modest in setting our goals. The Helm's Deep feeling seemed to be on-point, with our servers besieged on all sides. Smashing through the final milestone before we had our first coffee of Friday morning, we ended the weekend having attracted just shy of 270,000 new downloads. Time to get those artists working on the free weekend rewards. About 25 per cent of new players played on more than one day of the weekend, so it seemed we were landing significantly more shots than we anticipated. Concurrent users were the real showpiece stat here though, seeing a 400 per cent increase on the peak CCU for the previous week. Whereas before we had to spend several minutes waiting in a playtest for the server to fill, we were now racing to get into one before the servers filled to the brim. It completely smashed the 80 per cent increase we predicted. More servers were hurriedly ordered in Singapore to cope with our emerging playerbase in that part of the world, and we battened down the hatches to see how the game would hold up. We wanted a stress test, but we had no idea ...

Not only did we see a huge boost in the raw numbers, but for a short time we were in fact in the top fifty trending games on all of Steam. Being up there and rubbing shoulders with giants on the week that Grand Theft Auto V launched on PC was no mean feat, and for the spell during which Fractured Space appeared above the latest Call of Duty, we were in dreamland. Watching the Delta DNA and Steam stats grow was much like the feeling taking a non-professional club all the way to a Wembley final on Football Manager. But after the initial rush of seeing our humble game fly up the charts, it was time to get started on the real work.

All this data was very nice, and certainly put a spring in our steps (being a small team of about 24 in a pretty makeshift studio and appearing in the top trending games on Steam will do that), but what came of all this after the dust settled? Well, though we had occasional server problems over the course of the weekend thanks to the hordes of new players (the hamster powering the wheel just couldn't keep up with the pace as our regular 1,000 Daily Active Users became just shy of 60,000 at the peak) no really disastrous issues reared their heads. Considering the huge step up in scale our game was experiencing, this was encouraging. The worst we had to contend with was a scary period of around 30 minutes of downtime as our third-party servers went kaput, but this was solved in relatively short order. Thanks to careful monitoring of the game's performance over the weekend, disruption was kept to a minimum, with a couple of strategic server resets stepping in to keep lag under control at relatively quiet moments. A few weaknesses were dug up, with a couple of code issues and problems with our third-party account management becoming known, but lessons are

learned through hardship, and all this identified areas for improvement. We also, suddenly, had a massively increased install base to work with – that community suddenly became a lot harder to manage. There was naturally something of a tail off, but numbers remain healthy even now, thanks to the few days in which we gave the base game away for free.

Perhaps the biggest win of all though, was in the reception we got. The Free Weekend generated 64 new articles on our radar, with the likes of PC Gamer and PCGamesN giving the game positive write-ups, as well as a healthy reception in the German-speaking world from sites such as 4players. Our number of Steam reviews shot through the roof too, but the average percentage score didn't drop anywhere near as sharply as you might expect. The chart below shows the number of reviews, and the average scores over the weekend. So while we did drop a little below the mythical 80 per cent positive mark on the Steam store, an average drop in score of around 5 per cent is a small hit to absorb. So while the number of reviews shot skyward in a big way, the average score stayed relatively static, which when considering the game's positioning as a MOBA-like title, with the famously discerning playerbase such a genre typically garners, that is something to be pleased with. Many of the negative customer reviews correlated too, throwing up clear action points going forward – the tutorials on offer were hot topics, with many players not feeling they were communicated boldly enough to new users. With slight issues like this now being worked on, it should hopefully be a matter of time until we see that average percentage creep back up to pre-free weekend levels. The open development mandate no doubt helped to maintain this, with so many of the reviews remarking on the satisfaction players have of being able to talk directly to, say, a level designer, or gameplay programmer, about their ideas or concerns. This was definitely reflected in the feedback we saw, and the feelgood factor was firmly in place despite the non-stop nature of the weekend.

Free Weekends: The Pitfalls

If we had to pick a downside to the method, it would probably be in the sheer amount of work required to keep up. Adopting as we do an Open Development philosophy, we did find ourselves losing a little dev time to dealing with the huge amounts of new players that needed guidance or had burning questions. While we do have two full-time community managers dedicated to this, being a smaller studio meant this really was a case of all hands racing to the pumps. While this is all part of the job – after all, if you live by the sword, you die by the sword – it has to be said that a free weekend isn't something to rush into. Real preparation had to go into the promotion, and if the worst thing you can say about something is that it gave you more work to do, then that's a nice problem to have. Also, if

you're considering following a similar approach, then be ready to buy more servers ...

Press Coverage

Another big win came from our press coverage. While going through Early Access is always a tricky proposition in this regard, the unique angle of our free weekend helped to get more outlets reporting on our game than had been since our initial announcement. The press are traditionally cautious in their approach to Early Access games, what with the element of uncertainty that comes with the territory. Unless you're »DayZ« or »Heroes of the Storm«, it's not easy to get noticed in the wild west of Early Access, so the angle that our free weekend gave the press to work with was a positive influence when looking at the overall reception. When Fractured Space was revealed, the combination of open development and (we like to think) the team's story as a through-line from the Kickstarter success of Strike Suit Zero, gave us a decent splash: 91 mentions arriving on our radar screen, before tailing off for several months as we got our heads down to finish making the thing. We got 64 new mentions regarding the promotion from outlets across Europe, which was a welcome spike – especially given the vastly improved state of the game this much further along production. After the weekend, with more populated servers and the game creating a lot more buzz than before, we began to see more and more positive previews appearing around the specialist press. The game started to appear more and more around the enthusiast press too, with YouTube searches suddenly becoming far more fruitful, and we've even just started to put together our first clan vs clan tournament, to be streamed live on Twitch. For a game that's still in Alpha, we're pleasantly surprised to be this ahead of our own expectations.

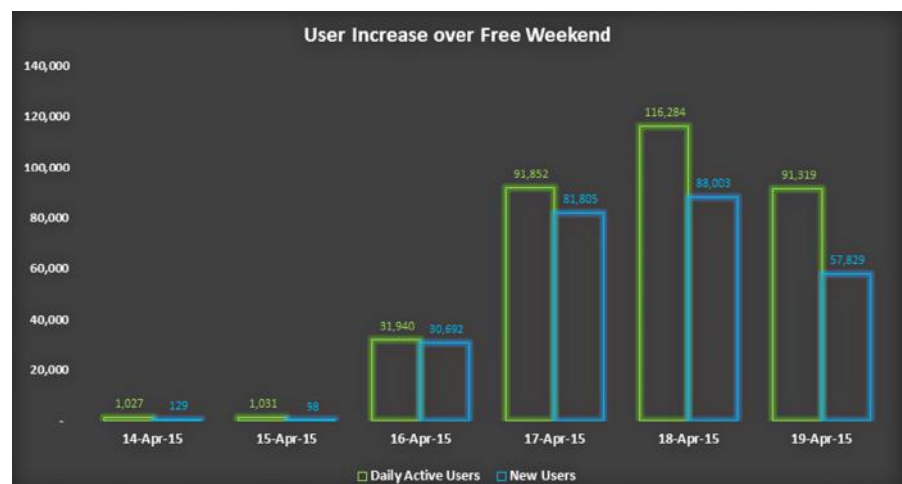
Lessons Learned

While it would be fun to rest on our laurels for a while, it was important to take lessons from what we experienced over this crazy week. Matchmaking and allowing custom teams was a hot topic that shot up the Trello voting, and we adjusted accordingly. Being able to adapt at pace is definitely both one of the advantages and challenges of open development – while the free weekend highlighted priority areas that needed work, we were flexible enough to get straight to work on them for imminent patches. It's all well and good to bask in the warmth of the increased numbers, but if you don't use all this raw data for anything, it can quickly become just a set of gimmicky statistics to show off with. The key thing, especially with an online-only title like ours, is to use every last scrap of data you can get to hone the game into the best possible shape you can. So while putting your early-access, open development, PvP MOBA free on the world's largest distribution platform for the best part of a week may on the

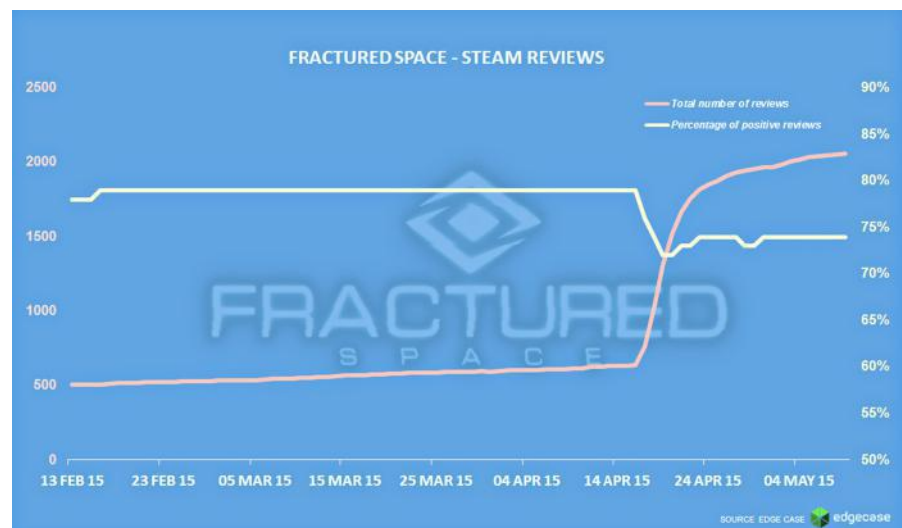
face of it sound risky, we took a real run at it and were pretty delighted with the results. To hit the numbers we did, going toe-to-toe with million-selling franchise game »Endless Legend« on the same playing field, was quite the achievement for our ambitious studio, and the signal boost that has resulted has given us much to be optimistic about for the future. Numbers have stayed healthy, we've seen a decent increase in the amount of paying customers in-game, and Fractured Space is as competitive as ever.

With this enlarged audience, higher profile and plenty of new player data, we have to choose our next steps carefully. We're introducing two new ships in the next update, along with another batch of crew members, doubling the available characters for players to use. This increases the players' customisation options massively, and with the new base system making significant changes to the game mode in the near future, we're providing huge chunks of new content. As the player base has grown, so too will the game – it's an exciting time for us all following the weekend of insanity that is going free on Steam.

Sam Faulkner



Well, that escalated quickly.



Steam reviews went up as expected, though the average score remained surprisingly consistent.



SON OF NORS

STEINIGER WEG AUF STEAM

Erst Kickstarter brachte für Son of Nor den Durchbruch. Die stillalive studios erklären, wie das Portal Einfluss auf den Erfolg auf Steam nimmt und was es nach Release zu beachten gibt.



Chris Polus
ist bei den stillalive studios
verantwortlich für Sound-
design, Marketing und PR.

Chris Polus ist gebürtiger Pole, 36, und Sounddesigner aus Basel. Bei stillalive studios ist er verantwortlich für Sounddesign, Marketing und PR. Für »Son of Nor« produzierte er auch alle Game-Trailer und viele der Dev Diaries. Als freischaffender Sounddesigner macht er aber auch den Sound für Animations-, Dokumentar- und Kurzfilme, spricht Kommentare und Rollen für Dokumentationen und Games, lehrt Sounddesign an der Zürcher Hochschule der Künste (ZHdK) und erstellt Trailer für Games aller Art.

Mehr über seine Projekte: <http://chromonaut.ch>

Das Projekt »Son of Nor« ist in vielerlei Hinsicht ein Erfolg. Unsere stillalive studios haben sich über das Projekt gefunden, zusammengetan und sind von drei Hobbyisten auf 13 Mitarbeiter angewachsen. Wir gründeten eine Firma und absolvierten eine erfolgreiche Kickstarter-Kampagne, in der wir über 150.000 Dollar sammelten. Wir kamen durch Greenlight, als es noch schwierig war, und wurden sogar über mehrere Tage auf der Frontseite von Steam im Karussell gefeurt. Presse, YouTuber, Streamer – wir haben viel Aufmerksamkeit bekommen. Reicht das für einen Erfolg? Was können wir Indie-Teams empfehlen, die ähnliche Stationen auf ihrem Weg planen? In diesem Artikel möchten wir dazu ein paar Antworten geben.

Wieso Kickstarter?

Unsere Kickstarter-Kampagne haben wir lange vorbereitet. Wir wussten, es würde schwierig werden. Vor allem, weil bei Indie-Teams Kampagnen im Bereich von 5.000 bis 15.000 Dollar üblich waren. Wir wollten mit unserer Kampagne aber 150.000 Dollar einfahren! Genug Budget eben, um unserem Team endlich Löhne zahlen zu können. Bis dato war »Son of Nor« lediglich ein Freizeitprojekt. Und es lechzte nach mehr Manpower. Um vorwärts zu kommen, mussten also alle Teammitglieder ihren bisherigen Haupterwerb niederlegen und zu 100 Prozent daran arbeiten.

Wir lasen also sämtliche Artikel, die wir zu Kickstarter finden konnten. Immerhin stand die Existenz des Projekts und damit des Teams auf dem Spiel. Artikel, Analysen und Erfahrungsberichte von Leuten, die erfolgreich eine Kickstarter-Kampagne durchgeführt hatten, sowie auch Beiträge sogenannter Kickstarter- und Marketingexperten.

Folgende Punkte wurden immer wieder aufgegriffen:

- Eine Kampagne muss mit einem großen Knall beginnen und die aufgenommene Fahrt darf nicht abflauen, sonst kommt sie zum Stillstand.
- Der Aufwand darf nicht unterschätzt werden. Die Vorbereitungen brauchen viel Zeit, und wenn die Kampagne erst angelaufen ist, steht alles andere still. Backer wollen Updates und Stretchgoals. Es muss immer was gehen. Selbst nach dem Schlusspfiff geht es weiter: Dankeschreiben, Datenbanken für Adressen, Umfragen etc.
- Die ersten und die letzten Tage sind ausschlaggebend. In den letzten Tagen kommt



»Son of Nor« startete als Hobbyprojekt mit einem Team von drei Entwicklern und durchlief später eine erfolgreiche Kickstarter-Kampagne, Greenlight und Early Access auf Steam und wurde schließlich Ende April 2015 veröffentlicht.

oft noch mal so viel an Spenden rein wie in der ersten Woche.

- Die Struktur der Belohnungen (Rewards) muss gut durchdacht sein. Die Preisebenen sind bestenfalls so zu gestalten, dass man als Backer immer versucht ist, die nächsthöhere Stufe zu wählen.
- Keinesfalls dürfen die Kosten materieller Goodies unterschätzt werden: Herstellungsaufwand, Produktionskosten, Lieferkosten weltweit ...
- Niemand liest den Kampagnentext wirklich, alle gucken nur das Video an. Trotzdem müssen alle Infos schön präsentiert auf die Seite, sonst sieht es unseriös leer aus.
- Da die meisten nur das Video gucken, ist es einer der wichtigsten Schlüsselfaktoren des Erfolgs. Weckt man Neugier und ein »Haben will«-Gefühl, ist ein Großteil der Miete bezahlt.
- Nichts geht über ein Produkt, das ein Bedürfnis deckt. Double Fines »Broken Age« aktivierte alle sich vernachlässigt fühlenden Adventure-Fans. »Pillars of Eternity« richtete sich an eine riesige »Baldur's Gate«-Fangemeinde. Ein »Flappy Birds«-Klon deckt niemandes Bedürfnis.

Gute drei Monate steckten wir bei stillalve studios in die Vorbereitung. Das Projekt legte faktisch die komplette Entwicklung des Spiels lahm. Als unbekanntes Team mit dem ersten Projekt konnten wir keine existierende Fangemeinde aktivieren. Ein Problem, das wohl noch so manches Indie-Team haben dürfte.

In weiser Voraussicht haben wir zu stillalives Anfängen bereits damit begonnen, eine Fangemeinde aufzubauen. Wir erstellten Twitter-Konten und berichteten über unseren Entwicklungsalltag, dazu kam eine Facebook-Seite mit Fotos und Screenshots und jede Woche ein aufwändiges Video auf YouTube, in dem wir die Fortschritte der Entwicklung zeigten. Wir posteten die Videos auf Indie DB, in Foren wie Unity3D, auf TIGSource, GameDev, Spieleprogrammierer.de und weiteren und versuchten, sowohl Entwickler als auch Spieler zu begeistern. Auch das Reisen zu Gameevents wie gamescom, GDC oder QuoVadis half, Kontakte zu knüpfen. Befreundete Indie-Studios können eine große Hilfe sein, wenn es darum geht, Nachrichten zu verbreiten!

Die Kampagne wurde minutiös geplant, die Rewards und das Kampagnenvideo Dutzende Male an Außenstehenden getestet. Wie haben wir die genannten Punkte umgesetzt? Was klappte gut, was eher nicht? Unsere wichtigsten Learnings:

Der große Knall des Kampagnenstarts

Die engsten Verbündeten eines kleinen Indie-Teams sind die drei F: Family, Friends, and Fools. Niemand war sich zu schade, einige Tage vor der Kampagne Familie und Freunde in Kenntnis zu setzen und um Mithilfe zu bitten. Über Brief, E-Mail, Facebook-Freunde und Freundesfreunde

sowie Twitter-Bekanntschaften verbreitete sich die Kunde unseres Vorhabens.

Die Presse durfte auf keinen Fall fehlen.

Über Monate sammelten wir E-Mail-Adressen von Journalisten, die sich vom Konzept des Spiels hätten angesprochen fühlen können. Wir bauten Verbindungen auf, und manche berichteten sogar schon im Vorfeld über uns, so auf Kotaku, »Rock, Paper, Shotgun« und vielen weiteren Gaming-Seiten.

Als Tool setzten wir auf MailChimp. Bis zu 2.000 E-Mail-Empfänger können hier kostenlos angeschrieben werden und man kann tracken, wie viele Mails geöffnet oder gar geklickt werden. Das hat geholfen, weitere Mails zu optimieren und bei den Nichtöffnern nachzuhaken.

Die Trends auf Kickstarter änderten sich jedoch rasend schnell. Der Goldrausch forderte die ersten Opfer. Journalisten wurden dermaßen mit Kickstarter-Mails überflutet, dass eine Müdigkeit, ja schon fast Aversion einsetzte. Kaum ein Outlet mochte mehr über Kampagnen berichten. Es kostete uns viel Extraenergie, doch noch an Coverage zu kommen.

Wir scheuten uns auch nicht, außerhalb des Teams Hilfe anzufordern. Unser enger Partner Remote Control Productions (RCP) half uns tatkräftig bei der Entwicklung und Verbreitung der Kampagne. Zusätzlich verschlossen wir uns auch nicht davor, einen Batzen Geld in die Hand zu nehmen und eine PR-Agentur sowie eine Social-Media-Expertin für die Dauer der Kampagne zu verpflichten. Die PR-Agentur organisierte gar eine Presstour durch Deutschland und Österreich. Wir besuchten über zehn Redaktionen persönlich. In der Tat hat uns dieser direkte Kontakt aus der Masse der Kickstarter-Projekte rausstechen lassen, was sich in zusätzlicher Coverage ausgedrückt hat. Durch die Connections von RCP bekamen wir sogar Artikel in der GameStar und PC Games.

Mit unserem internationalen Team haben wir weiterhin versucht, in allen Ländern an den Nationalstolz zu appellieren. Auf der einen Seite ein Projekt, das mal nicht aus Amerika kommt: Österreich, Schweiz, Ecuador ... Auf der anderen Seite die Presse, die eine Story daraus machen kann. Der Underdog und kleine Entwickler aus Österreich (bzw. Schweiz bzw. Ecuador ...) versucht, ein ambitioniertes Projekt über Kickstarter zu finanzieren! Das kam gut an.

Der Zeitaufwand

Wir haben es in allen Artikeln gelesen, aber richtig erfassen konnten wir es erst, als wir mittendrin steckten. Eine Kickstarter-Kampagne ist ein Vollzeitjob für den größten Teil des Teams. Drei Monate intensive Vorbereitung. Kurz vor dem Start war die Hölle los. Alle rotierten 18 Stunden am Tag, um die letzten Assets, Texte, Videos und Communiqués fertigzustellen.

Doch während der Kampagne fängt die Arbeit erst richtig an! Die Massen strömen auf das Projekt ein. Und sie haben Fragen. Fragen, die im Kampagnentext detailliert abgehandelt



Rewards für die Backer einer Kickstarter-Kampagne sind gerade für ein kleines Team eine knifflige Sache, da sie im Nachgang viel Arbeit bereiten. stillalve studios würde für das nächste Projekt vor allem digitale Belohnungen wählen.

werden. Doch den liest ja niemand. Zudem kommen sie nicht nur über Kickstarter, auch über E-Mail, Facebook, Twitter. Alle Stationen mussten bemannt werden.

Hier blühte unser internationales Team richtig auf. Weil wir Mitglieder in Europa und den amerikanischen Kontinenten haben, konnten wir einen 24-Stunden-Support bieten. Ständig war jemand von uns erreichbar. Das brachte viele Sympathiepunkte. Schnelle, präzise und freundliche Antworten haben die Backer begeistert und manche zu treuen Fans gemacht.

Die Struktur der Rewards

Unsere Kampagne lief zu einer Zeit, zu der materielle Goodies total in Mode waren. Wir dachten, coole Gadgets seien nötig, um Backer zu gewinnen. Spielboxen, Landkarten, Handbuch – wir priesen sogar ein Amulett von Nor an. Die Progression der Rewards haben wir dabei recht gut hinbekommen.

Im Nachgang der Kampagne kosteten uns die Erstellung und die Produktion der Goodies aber mehrere Mannmonate. Ein Amulett etwa musste modelliert und mehrere 3D-Printshops mussten in Preis und Qualität verglichen werden. Von verschiedenen Lieferanten kommen nun die einzelnen Rewards zusammen, die in den richtigen Schachteln zu den richtigen Backern gelangen müssen. Ein unglaublicher Aufwand.

In unserem Team sind wir uns jedenfalls einig, dass für ein nächstes Projekt sinnvolle digitale Goodies zum Einsatz kommen und – wenn möglich –, dass wir materielle Rewards ganz vermeiden.

Das Kampagnenvideo

Wir wussten von uns selbst, dass wir bei Kampagnen meist nur das Video schauten. Viele Transaktionen waren Impulskäufe aufgrund der Videos: »Wow, coole Idee, gebackt!«

Spielszenen dürfen auf keinen Fall fehlen, denn Kampagnen von Studios, die im Video

kein Game Footage zeigten, haben uns eher abgeschreckt. Ausnahmen sind Teams mit einer starken Historie wie Obsidian oder Double Fine. Wir Indies müssen durch Taten überzeugen, wenn wir keine glorreiche Vergangenheit aufzuweisen haben.

Witz und Persönlichkeit dürfen meines Erachtens ebenfalls nicht fehlen. Ein Kickstarter-Video ist der Kampf um Sympathie und schließlich um die Kreditkarte. Man sollte rüberbringen können, dass man ein begeistertes Team von Indie-Entwicklern ist. Die netten Nachbarn von nebenan! So weit zu den »Dos«.

Nun zu den »Don'ts«. Wir wollten so viele Informationen wie möglich ins Video packen, und dieses wurde satte neuneinhalb Minuten lang! Eine Aufmerksamkeitsspanne, die im Internet heute schon episch nahe am Weltrekordniveau liegt, wenn sie denn jemand durchhält (7,77 Prozent der Leute haben das Video zu Ende geschaut). Großer Wert wurde in »Son of Nor« auf die Story gelegt, so sollte dann auch im Video was davon gezeigt werden. Die Wahrheit jedoch ist: »Son of Nor« lebt als Spiel nicht von der Story, was sie damit für die meisten uninteressant macht. Man will Spielgetümmel sehen!

Auch schlechtes Audio war ein No-Go. Personen in hallenden Räumen, die kaum verständlich brabbeln, führen zu sofortigem Wegklicken.

Ein Kickstarter-Video würden wir heute deshalb in Richtung »Elevator Pitch« entwickeln. Kurz und bündig: möglichst sofort zur Sache kommen. Deshalb erst die besten Spielszenen zeigen und sie nicht bis zum Schluss aufheben (7,77 Prozent!). Dann die Backer mit Sympathie, Gesicht, Stimme und Begeisterung über die coolen Features, die man zu bieten hat, auf emotionaler Ebene abholen.

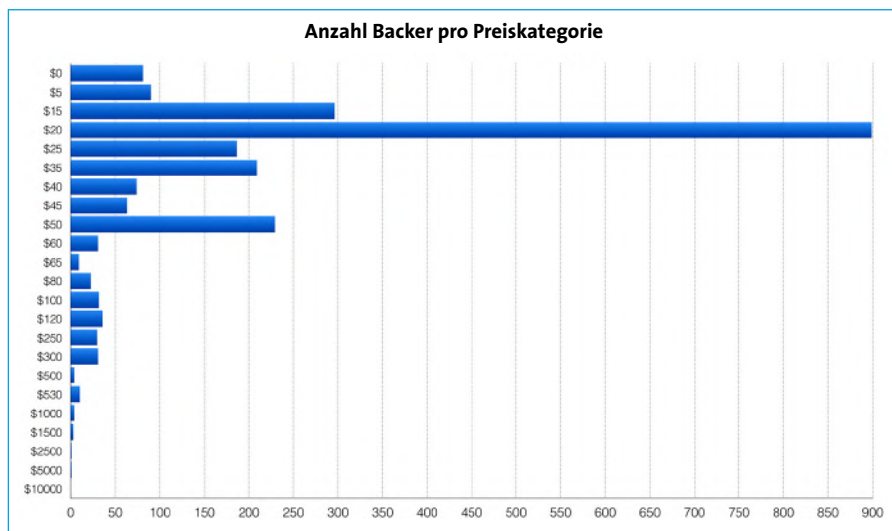
Ein Produkt, das im Trend liegt

Natürlich sind wir alle von der eigenen Idee überzeugt, sonst würden wir nicht Zeit und Geld in ein Projekt stecken. Leider reicht das nicht aus. Der Markt, die Konsumenten scheinen sich in Schwärmen wie Heuschrecken von Trend zu Trend zu bewegen. Liegen gute RPG-Games etwas länger in der Vergangenheit zurück, kann die bloße Ankündigung eines interessanten RPG-Konzepts die Massen bewegen. Nicht umsonst hatten jene Kickstarter-Kampagnen Erfolg, die alte Titel aus der Versenkung holten. Es hilft eben, wenn man eine bestehende Fangemeinde ansprechen kann.

Uns war diese Dynamik nicht bewusst. Mit einer komplett neuen IP und mit einem Spielprinzip, das es in dieser Form noch nicht gab, kämpften wir um jeden Backer einzeln.

Woher kam der Traffic auf die Kampagne?

So viel Presse, Freunde, YouTuber – was hat nun am meisten gebracht? Verwandte und Freunde haben sicher mit den höchsten Beträgen pro Kopf unterstützt. Doch der Großteil der Leute kam überraschenderweise von Kickstarter



Im Falle von »Son of Nor« brachten die hochwertigen Rewards nicht den größten Erfolg. Die meisten Backer spendeten lediglich den nötigen Betrag, um später das fertige Spiel zu bekommen.

selbst, nicht über Berichterstattung der Presse: satte 44 Prozent von innerhalb der Kickstarter-Website. »Son of Nor« war auch ein »Staff Pick« und tauchte so auf der Frontseite auf. Super funktionierte ebenfalls die Mechanik, dass Backer eine Mail an ihr Freundesnetzwerk auslösen. Eine Art vertrauenswürdiger Tipp.

Die Presse war beim Start unserer Kampagne schon dermaßen kickstartermüde, dass wir große Schwierigkeiten hatten, überhaupt erwähnt zu werden. Einen Glückstreffer landeten wir, als wir ein Brain-Computer-Interface aus der Medizinforschung als Steuerungsgerät für »Son of Nor« vorstellten. Ein Spiel, gesteuert mit Gedankenkraft? Etliche hielten es für einen Fake, was viel Aufmerksamkeit erregte, doch die Technik existiert. Das war zahlreichen Game-Seiten eine Story wert und unser Video wurde in kurzer Zeit über 80.000-mal angesehen (vor allem auf Kotaku, PC Gamer und RPS).

Doch auch hier stellte sich bald Ernüchterung ein. Die Headline zog zwar jede Menge Wissensdurstige an, diese wollten jedoch nur ihre Neugier stillen und waren nicht an unserem Projekt interessiert. Gebackt hatten uns an jenem Tag 43 Leute – nur geringfügig mehr als an anderen Tagen in dieser Phase.

Wir suchten nach Wegen, neue Backer-Gruppen auf uns aufmerksam zu machen. Dazu analysierten wir Kampagnen, die parallel zu unserer liefen. Etwa die von Larian Studios für »Divinity: Original Sin«. Sie gaben uns den entscheidenden Hinweis. Sie bestätigten nämlich unsere Erfahrung, dass die Presse kaum noch Backer mobilisieren konnte. Auch Leser waren offensichtlich kickstartermüde geworden, denn unser Top-Referrer war Rock, Paper, Shotgun mit gerade einmal 48 Backern. Larian wendete sich folglich den YouTubern zu, und das hievte die »Divinity«-Kampagne laut Larians CEO Swen Vincke über die Ziellinie.

Mit Eifer haben wir also ebenfalls YouTuber kontaktiert. Let's Player mit großen Abonnentenzahlen spielten unsere Sandbox-Demo im PvP-Modus und hatten dabei sichtlich Spaß. Es kam auch zu lustigen Glitches. Das hat vor allem in den letzten Tagen sehr gut funktioniert, wodurch die Kampagne noch mal an Fahrt aufnahm.

Jede Maßnahme war ein Puzzleteil: Presse, das Brain Device, konstante Tweets und Facebook-Beiträge sowie die YouTuber. Wäre einer der Teile ausgeblieben, hätten wir unser Ziel von 150.000 Dollar wohl nicht erreicht, denn wir landeten schließlich gerade mal so bei 151.000 Dollar. Sauknapp!

Kickstarter-Fazit

Am Ende kamen 75.886 Videoaufrufe und 2.244 Backer zusammen. All diese Tipps sind aber mit Vorsicht zu genießen, denn die Zeiten ändern sich rasant. Wichtig ist, heutige Projekte auf Kickstarter genau zu analysieren. Was machen sie gut, was überzeugt Backer? Diese Fragen sollten in heutigem Kontext gestellt und Strategien

für das eigene Projekt daraus abgeleitet werden. Einfacher ist es nicht geworden, denn immer mehr Teams mit immer kleineren Budgets kämpfen um die Gunst der Geldgeber.

Steam Greenlight

Greenlight ist unserer Meinung nach heute kein großes Thema mehr, seit Valve die Tore geöffnet hat. Games werden in großen Stapeln durchgewunken und es ist nur eine Frage der Zeit, bis man sein Spiel mit etwas Öffentlichkeitsarbeit und Marketing auf Steam bekommt.

Wir haben ungeniert das Rampenlicht unserer Kickstarter-Kampagne genutzt, um auf unsere Greenlight-Kampagne aufmerksam zu machen, und konnten so rasch das grüne Licht verbuchen.

Interesse eines Publisher

Das Erreichen von 150.000 Dollar auf Kickstarter und das Greenlight auf Steam zeigten auch, dass ein öffentliches Interesse an »Son of Nor« bestand. Das fand auch Viva Media reizvoll, ein US-Publisher, den wir für uns gewinnen konnten. Viva schoss weitere Finanzen zu, mit welchen wir den Umfang unseres Spiels erweitern konnten. Unter anderem setzten wir volles Voice-over für alle Charaktere mit AAA-Sprechern um!

Steam Early Access

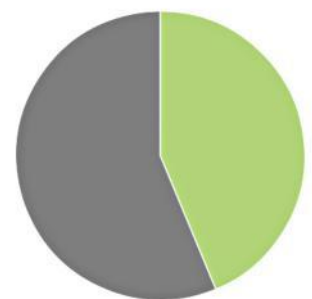
Irgendwann war für uns der Zeitpunkt gekommen, »Son of Nor« als Early-Access-Titel zu veröffentlichen. Dabei ging es nicht um zusätzliche Einnahmen, wir wollten vor allem Feedback zu unserem Entwicklungsstand. Selbst hatten wir den Blick dafür nämlich verloren. Alles wirkte intuitiv und smooth. Es war also Zeit, eine breitere Masse ranzulassen.

Das war eine der besten Entscheidungen für das Spiel! Zwar steckten wir viel Kritik ein: Websites schrieben, wir hätten das »Early« in »Early Access« neu definiert. Doch im Steam-Forum kam Feedback zu spezifischen Teilen der Spielmechanik rein.

Das schlägt anfangs auf die Stimmung. Unsere Playtests auf Messen und mit außenstehenden Personen waren doch überwiegend positiv verlaufen, warum jetzt das? Das muss



Referrers



Pledged via Kickstarter	Pledged via external referrers	Average pledge amount
\$66,094	\$85,080	\$67,37

Obwohl das Team hinter »Son of Nor« viel Interesse seitens der Presse und von YouTubern erreichen konnte, wurde beinahe die Hälfte der Unterstützer direkt über die Kickstarter-Website auf die Kampagne aufmerksam.

Funding progress



Die Kickstarter-Kampagne erreichte am Ende nur knapp das gesetzte Ziel von 150.000 Dollar. Den entscheidenden Hinweis, um die Finanzierungshürde zu erreichen, lieferte Larian Studios: Let's Plays auf YouTube.

man erst mal schlucken. Die falsche Reaktion wäre es dabei gewesen, den Kopf in den Sand zu stecken. Also sammelten wir unsere Kräfte, nahmen jede einzelne positive wie negative Meinung unter die Lupe und versuchten, sie nachzuvollziehen. Wir diskutierten intensiv mögliche Maßnahmen und implementierten neue Ideen. Neun Monate lang! Teilweise wurden ganze Spielabschnitte überarbeitet, weil Spieler sie zu schwer oder zu unlogisch fanden.

Early Access hat unserem Team frische Perspektiven und Impulse gegeben und das Gameplay merklich aufgewertet! Beim nächsten Projekt werden wir Early Access auf jeden Fall wieder in Erwägung ziehen.

Steam Release

Zwei Monate vor unserem Release waren die meisten Feedbacks der Early-Access-Phase implementiert und fast alle erfassten Bugs gefixt.

ziehen, doch kumuliert hatte das Banner mit Sicherheit über 20 Millionen Impressions!

Preisgestaltung

Ein wichtiges Thema ist die Preisgestaltung. Welcher Preis ist für den Spielumfang angemessen? Und sind Spieler überhaupt bereit, zu zahlen? Zu der Zeit unserer Kickstarter-Kampagne schien es üblich, einen Preis von etwa 20 Dollar für einen Indie-Titel zu verlangen. Etwas kleinere Produktionen gingen für plus minus 14,99 Dollar über den virtuellen Ladentisch. Wir fühlten uns also gut aufgestellt.

Nicht einberechnet hatten wir, dass mittlerweile ein großes Überangebot an Games herrschte. Dutzende Indie-Bundles sahen riesige Verkaufszahlen, und im Mobilmarkt wurde es noch dichter, als es ohnehin schon war.

Mehr Presse, YouTuber und PR

Mit dem Release schrieben wir all unsere Pressekontakte wieder an. So zum Beispiel den »Hot Pepper Gaming Fire Sale«. Ein beliebtes YouTube-Format, bei dem Entwickler ihr Spiel einer großen Zahl an Zuschauern anpreisen dürfen, vorher aber eine extrem scharfe Chili essen müssen. 20.000 Views kamen hierbei zusammen, die Kommentare waren außergewöhnlich positiv. Man liebte es, uns leiden zu sehen.

Auch deutsche YouTube-Größen haben Let's Plays von »Son of Nor« produziert. Zombey mit seinen knapp 840.000 Abonnenten machte 53.000 Augenpaare aufmerksam. Ein Russe brachte beinahe 100.000 Views. Und ein amerikanischer Channel holte satte 300.000 Views für »Son of Nor« rein. Unser Release-Trailer konnte ebenfalls 148.000 Aufrufe verbuchen.

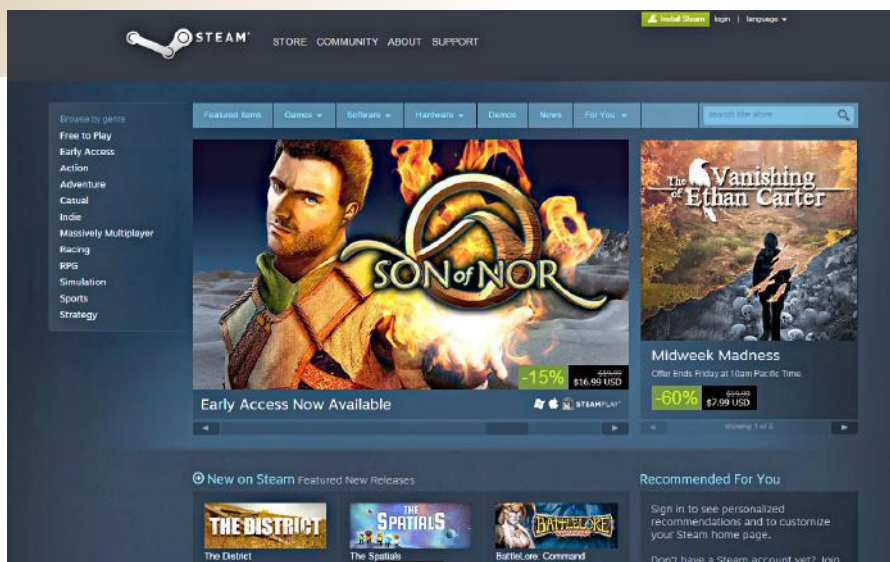
Verkaufszahlen

Am Release-Tag wehte uns trotz allem ein kalter Wind entgegen und wir merkten das schnell. Die Verkäufe blieben am Erscheinungstag, der traditionell eher ein Knaller sein sollte, hinter den Erwartungen.

Verkaufsoptimierung

Woran konnte der verhaltene Start liegen? Gleich nach dem Release stieg die Zahl der Leute, die »Son of Nor« auf ihre Wishlist setzten, sprunghaft von 10.000 (aus der Early-Access-Phase) auf 24.000 an. Offenbar hat das Spielprinzip neugierig gemacht, aber etwas hielt die potenziellen Käufer zurück. Waren 15 Prozent Launch-Rabatt auf den Vollpreis von 20 Dollar zu wenig? Oder warteten Steam-Kunden sowieso stets auf den nächsten Sale? Easter Sale, Midweek Madness, Summer Sale, Halloween Sale. Steam erzieht seine Kunden ja regelrecht dazu, auf Rabatte zu warten.

Auch ist das erwähnte Überangebot an Spielen unserer Ansicht nach ein großer Faktor. Durch zahllose Bundle Sales und konstante Rabattaktionen sammeln sich viele ungespielte Titel in den Steam-Libraries vieler Leute an. Auch in unserem Team ist das ein verbreitetes



Publisher Viva Media schaffte es, dass Valve »Son of Nor« für einige Tage auf der Startseite von Steam platzierte.

Doch ein passendes Release-Datum war schwer zu finden. Wir wollten in einer Woche starten, in der wir als kleines Team keine allzu große Konkurrenz zu erwarten hätten. Schließlich wäre die Arbeit von drei Jahren verpufft, hätten wir neben einem »Assassin's Creed« veröffentlicht.

Ein passendes Datum fand sich dann am 31. März 2015. Es war eine ruhige Launch-Woche mit nur wenigen Titeln. Die meiste Konkurrenz kam aus anderen Genres, sodass wir uns keine Sorgen über Kollisionen machten.

Durch die hervorragenden Beziehungen unseres Publishers Viva Media zu Steam wurde »Son of Nor« nicht nur live geschaltet, sondern erschien auch im Spielekarussell auf der Steam-Startseite. Alle, die Steam über die nächsten Tage öffneten, sahen ein Banner von »Son of Nor«! Auch in den Action-, Adventure- und Indie-Kategorien waren wir mehrere Tage auf den ersten Positionen zu sehen. Die genauen Zahlen können wir nicht mehr nachvoll-

Phänomen. Da überlegt man sich dreimal, ob man ein interessantes Game kauft oder erst mal einen Titel von der Warteliste startet.

Bevor also die aufgenommene Fahrt unseres Launches verpuffte, setzten wir eher früher als später die erste Rabattaktion an. Diese hatte den positiven Nebeneffekt, dass alle Steam-Kunden mit »Son of Nor« auf der Wishlist per E-Mail von der Aktion benachrichtigt wurden. Ein als signifikant wahrgenommener Rabatt von 50 Prozent löste das Impulskaufverhalten aus und die Verkaufszahlen konnten im Vergleich zum Launch um einiges gesteigert werden. Sie blieben aber weiterhin etwas hinter den Erwartungen.

Vergleich mit anderen Indies

Wir hatten nach der erreichten Berichterstattung der Presse, den YouTubern und der tollen Steam-Promotion dank Vivas Kontakten höhere Erwartungen an unsere Verkäufe. Haben wir also irgendeine Abzweigung verpasst?

Im April 2015 lud die Konferenz Quo Vadis in Berlin Entwickler von Indie bis AAA in ihre Hallen. Ein guter Ort, um sich mit anderen Teams auszutauschen. Und auch da hörte man aus vielen Ecken ähnliche Erfahrungsberichte. Dass nämlich trotz guter Pressestimmen, positiver Kritiken, hervorragender User Reviews und zahlreicher Let's Plays die Verkäufe ausblieben. Auch im Internet fanden wir bestätigende Erfahrungsberichte. Zum Beispiel von Trite Games, deren Spiel »Of Carrots And Blood« von niemandem geringeren als PewDiePie gespielt wurde. Drei Millionen Views später präsentierte sich folgende Statistik: 595 »pay what you want«-Downloads (entspricht 0,02 Prozent). 3 von diesen 595 bezahlten was, nämlich kumuliert acht Dollar!

Wir bildeten keine Ausnahme. Mehrere Vorträge auf der Quo Vadis handelten von dem Crash des Spielemarktes. Erleben wir diesen nun aus erster Hand? Die Vortragsräume waren voll, doch Antworten, was man diesem »Crash« entgegensetzen sollte, konnte niemand geben.

Alternative Strategien

Im Wirtschaftsunterricht bekommt man zu hören, dass es mehrere Strategien gibt, um sich am Markt zu behaupten.

- Man bietet etwas, was andere nicht bieten können, etwa höhere Qualität in bestimmten Bereichen (Story, Grafik oder Gameplay).
- Man ist der Erste, der ein komplett neues, innovatives Produkt offeriert bzw. der Erste in einem komplett neuen Markt.
- Man hat die tiefsten Preise.
- Und dann gibt es noch den Sechser im Lotto: Man hat einen Hit!

Meist beinhalten Hits auch einen oder mehrere der obigen Elemente. Sie haben es aber an sich, dass sie nicht planbar sind. Hits sind irgendwie zur richtigen Zeit am richtigen Ort. Und trotz übersättigter Märkte gibt es sie immer wieder, diese Ausbrecher! Das macht doch Hoffnung!

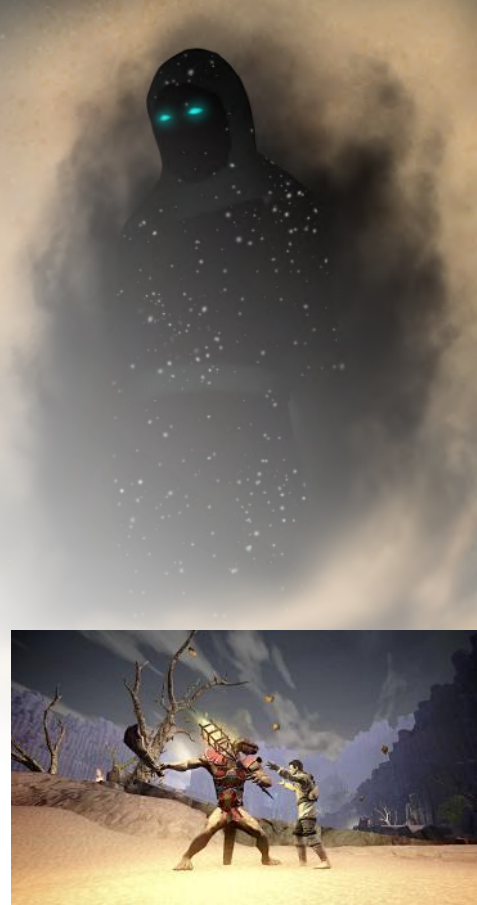
Wir wussten, dass »Son of Nor« nicht mit den qualitativ besten Games mithalten konnte. Eine neue, innovative Spielmechanik hätten wir zwar gehabt, doch diese war offenbar ein zu kleines Alleinstellungsmerkmal, was Punkt 2 auch ausschloss. blieb uns also nur noch der Weg über einen Preis, bei dem auch Leute mit überquellenden Steam-Bibliotheken einen Kauf aus Impuls heraus tätigen? Abschließend lässt sich das nicht sagen, zu viele Faktoren spielen hier mit rein. Sagen lässt sich hingegen, dass sich kommende Games gegen diese vollen Game-Libraries werden behaupten müssen – mit einer dieser Strategien oder einer Kombination aus diesen.

Können Indie-Spiele denn in einer anderen Kategorie außer dem tiefen Preis konkurrieren? In puncto Qualität können sie sehr wohl überzeugen, auch visueller Qualität. Es muss nicht immer AAA-Grafik sein. Ein spezieller, künstlerischer Stil kann ebenso anziehend wirken wie Ultrarealismus. Leicht erstarren möchte man trotzdem, wenn man liest, dass auch »Monument Valley« in der Entwicklung beinahe eine Million Dollar gekostet hat. Welches Indie-Spiel hat schon solch ein Budget?

Indie-Spiele können auch ganz trefflich eine innovative Spielmechanik bieten. Wo AAA-Spiele wegen exorbitanter Kosten praktisch keine großen Risiken eingehen, machen Indies freudig verrückte Dinge wie »Dear Esther« oder »The Stanley Parable« und sind erfolgreich damit.

Insofern, ja, es ist massiv schwieriger geworden, kommerziellen Erfolg zu haben, wenn man sich nicht substantiell in einem der Bereiche abhebt (Qualität, Neuheit oder Preis). Vor allem ist es kostspielig, da das »sich Abheben« einen größeren Aufwand oder im Falle von tieferen Preisen weniger Einnahmen bedeutet. Doch während bestehende Märkte sich sättigen, kommen glücklicherweise neue hinzu! Erst waren es die Konsolen, dann PCs. Nach Sättigung des PC-Marktes stand ebenfalls lange wie in Stein gemeißelt, dass nur ein großes Team Spiele herstellen könne. Allein die Lizenzen für Software und Engines waren zu kostspielig. Dann kamen Handhelds, später Mobilgeräte auf iOS- und Android-Basis. Durch immer günstiger werdende Tools (3D-, Sound- und Game-Engines) wurde das Erstellen von Games demokratisiert. Plötzlich konnte man als One-Man-Show wieder was bewegen! In diesem Jahr drängen Smartwatches auf den Markt und Virtual-Reality-Helme, und auch Hololinsen sind in aller Munde. Eine dieser Plattformen könnte durchaus einschlagen. Wer sich jetzt in Position bringt, der hat gute Chancen auf einen der vorderen Plätze in einem nicht übersättigten Markt, in welchem verrückte Ideen möglich sind und die Kundschaft experimentierfreudig ist. Solche Gelegenheiten wird es geben, solange neue Sachen erfunden werden. Also, ran den Speck, entweder über bessere Gamekonzepte oder verrückte Experimente auf neuen Plattformen.

Chris Polus



stillalive studios wusste, dass »Son of Nor« nicht mit den qualitativ besten Games mithalten konnte. Eine innovative Spielmechanik als Alleinstellungsmerkmal alleine half im hart umkämpften Markt jedoch nicht, wie sich schnell in den Verkaufszahlen zeigte.



»Son of Nor« als Early-Access-Titel zu veröffentlichen war eine der besten Entscheidungen, da das Team viel Feedback bekam: Teilweise wurden ganze Spielabschnitte überarbeitet und neu gebaut, weil es Spielern zu schwer oder zu unlogisch war.



Spiele, die einen Trend bedienen oder bereits eine Fangemeinde besitzen, haben es auf Kickstarter deutlich einfacher als eine komplett neue IP oder neuartige, unerprobte Spielmechaniken.



Making Games 04/2015

HOW TO CHANGE FROM AAA TO FREE2PLAY ON STEAM

Vostok Games decided to publish their shooter *Survarium* as a Free2Play game on Steam and quickly realized that there are a lot of pitfalls to be aware of.



Oleg Yavorsky is PR and Marketing Director at Vostok Games.

Oleg Yavorsky currently is PR and Marketing Director for Vostok Games, a game development studio based in Ukraine. Oleg started his games industry career at GSC Game World back in 2000 after graduating from Taras Shevchenko National University in Kiev (Master's degree in linguistics). He was responsible for PR and marketing campaigns for titles like «Cossacks: European Wars» and its two add-ons, «Codename: Outbreak», «American Conquest», «Cossacks II», «Heroes of Annihilated Empires», the «S.T.A.L.K.E.R.» series and other titles released and published by GSC. Currently he works on the online first-person shooter «Survarium».

Survarium« is an online Free2Play FPS in a post-apocalyptic setting; our team started development from scratch in March, 2012. Previously, our team worked on the development of the «S.T.A.L.K.E.R.» series at GSC Game World. After closing the studio, we managed to keep the team together and to find financing to start developing our own project «Survarium». At present the game is in the Open-Beta testing stage. While our PvP mode is the only one available at the moment, we are continuing to actively develop and improve the project. In the future we plan to complete the PvP mode and implement the long-awaited and greatly anticipated Freeplay (PvE) mode. Generally speaking, the sky's the limit!

We'd like to share a little bit about how we came to launch on Steam. Last year in July, we gave access to the Beta of «Survarium» to

players from the former USSR (Russian and Ukrainian versions). European players joined the testing beginning of November, 2014 (Polish version), and in January, 2015 testing started on the English and German versions of PvP. All this time the game has been in deep, active development and we have grown our online player base only by using organic traffic, without buying advertising. In this condition, when the game is not yet ready and retention is not ideal, it's necessary to maintain a sufficient level of concurrent users on the server, it is important to find traffic channels that can give a constant influx of new players.

Why going for Free2Play on Steam?

With this in mind we considered Steam Early Access. Of course, today Steam is the largest digital gaming platform with tens of millions of players from around the world. So our hope was to raise the number of concurrent users of the whole game and attract more players from North America in particular (since it is a significant part of the Steam audience). However, the launch of a «traditional» single-player game (such as «S.T.A.L.K.E.R.») on Steam was a known path for us, in case of a Free2Play game many issues arose, especially from the point of view of business strategy, because it is difficult to predict the results of the launch. I'll explain why ... There are many lasting opinions on Steam around F2P:

- Steam Players treat Free2Play games lightly (not taken as seriously).
- A project collects most of the money at the launch, including through the sale of DLC packs.
- There's not much traffic from Steam. More precisely there are a lot of players, but due to clause one, a very small part of them play the game.



«Survarium» is a Free2Play first-person shooter set in a post-apocalyptic world that started on April, 2nd 2015 on Early Access at Steam.

- It's difficult to attract new players because there are no convenient advertising tools in Steam.

Now we have to test these assumptions in our own experience.

How we prepared for launch on Steam

We integrated »Survarium« into the system. Initially, the access to the integration tools and settings page in your store will be closed until you complete all the required information about the company, bank details, etc. An U.S. tax identification number is also required. All these formalities took about three weeks. After that we got the opportunity to begin integrating the game into Steam. Everything went quickly and took less than two weeks. The only point which led to some internal disputes was if we should use SteamID to authorize players, or our own authorization system. Because the second choice would mean the need for additional registration on our website, after some hesitation, we made a choice in favor of the SteamID and greater conversion in player numbers as a result. The downside of this solution is that we do not have any email contact. We cannot for example, send separate notifications about new updates for the game to a Steam player. However, in practice, Steam is quite a self-sufficient social platform with its own set of tools that allow you to convey important information to the players. So, in our opinion, the choice was correct.

If we talk about the technical side, our programmers retained the opinion about the high quality of the Steam API, security systems, the authorization system, how tickets work, etc. The documentation fits in two pages, and everything works well. You can also note the high quality CDN (content distribution network), which certainly contributes to the high level of comfort of the players. Among the shortcomings we can note the insecurity of the process of giving achievements inside the game that makes cheating easier.

Also we had problems with the installer and Steam auto updater – we had to upload our own installer, which is executed before the first run of the game and abandon the Steam auto updater.

The effects of the game's store page

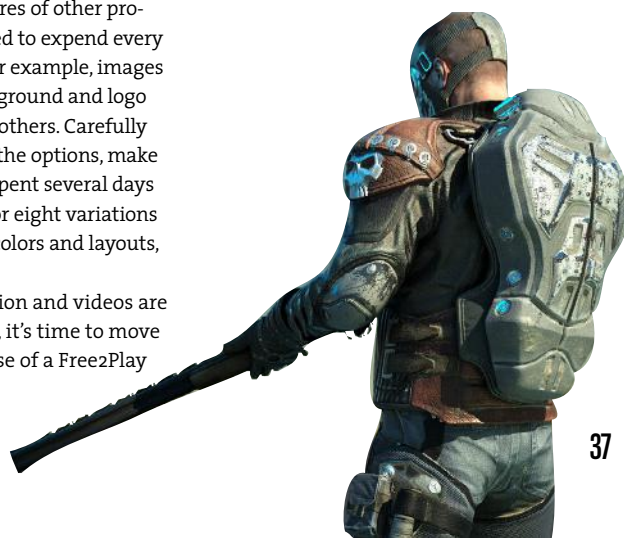
Next we started decorating the game page in the store. It should be noted that Valve did a really good job on the documentation and prepared samples of all types of images (about ten for each product, including DLC) and has done everything possible to simplify adding a new game to Steam. Therefore, the design process was easy and fast. It took more time to prepare graphic materials.

We did pay a lot of attention to the title pictures with the game's logo, such as main picture, small picture, etc. They are very important, because these are the images players see in the

Vostok Games started to sell DLC in three different price categories but quickly realized that the cheapest ones did not sell very well. Today there are only mid- and high-price categories available that range from 24.99 Euros to 45.99 Euros.

various sections of Steam, in search results, recommended games and so on. And often those banners are quite small. Therefore, the chance of a player clicking on your banner depends on how well you picked a background image and whether the name of the game is clearly visible when you reduce the image, for example to 120 by 45 pixels. It should also be noted that there will always be plenty of pictures of other projects next to yours, so you need to expend every effort into standing out. For example, images with a good contrasting background and logo will be more noticeable than others. Carefully consider this aspect, think of the options, make mockups and test them. We spent several days in discussion and had seven or eight variations of the picture with different colors and layouts, until we found the right one.

After the pictures, description and videos are uploaded to the project page, it's time to move on to sales matters. In the case of a Free2Play



game, in addition to monetization, you can also sell separate in-game bundles (DLC). We prepared three exclusive DLC-bundles for the launch. If you analyze any Free2Play game on Steam, you will find out that they all offer at least one paid set of items or in-game items as separate DLC. Our estimation was that players who want to get the maximum benefit »here and now« not wasting time on leveling will be ready to buy such bundle.

Prior to the Steam release our players did have the opportunity to purchase a variety of premium-items and sets of items in our own shop on the website, and we tried to make DLCs for Steam exclusive and different from what players could buy in the game previously. For example, we made an exclusive Steam-camouflage, unique weapons, etc. Thus, these DLC became interesting for new players, who learned about the game via Steam, and for older players who have decided to try out the Steam version.

Setting the right price for DLC bundles

Regarding the issue surrounding bundle prices, we proceeded from the assumption that non-paying players are not going to spend money on a bundle anyway and a paying audience will want to buy something that would distinguish them from the crowd. Therefore, all the DLC was made based on medium and high price category 15.99 dollars, 24.99 dollars and 49.99 dollars. For the medium bundle we also gave a discount of 20 percent at the start, which allowed it to be featured in the Specials category. After the first two weeks after launch, we realized that the medium-priced bundle

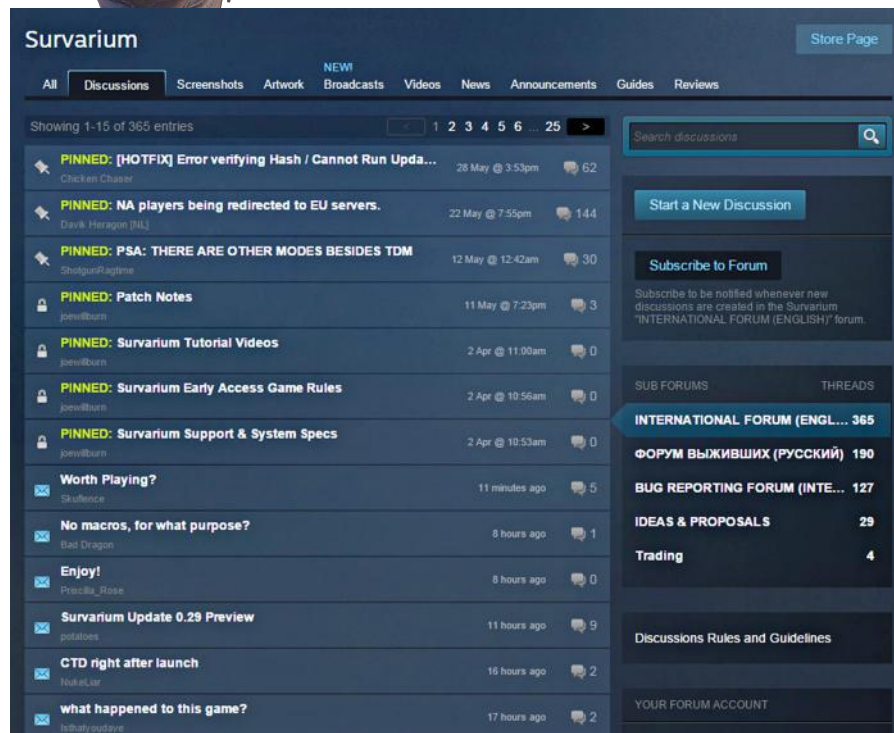
with a discount sold best, on the second place (and with a small difference in the number of copies sold) is the most expensive pack. After four weeks we added two new bundles (and also gave them a discount of 10 percent and 15 percent) and removed the old bundles from the store. It is important to refresh the range and offer players new content from time to time. It's better to time the release of new DLC bundles to certain important updates that attract the attention of a large number of players.

Genres and tags

We would like to pay attention to genres and tags. As Steam is a very big platform, it's important to appear in the greatest possible number of sections of the site. This is especially crucial for Free2Play games, where you need a constant influx of new players. In the framework of Steam this can be achieved by the envisioned featuring of the important updates, but in addition, you can maximize performance through proper design of the game in the store. For example, if you create a Free2Play shooter with RPG elements and it is in Early Access, don't forget to mark all the categories (and consider what else may be relevant to your product from the list) when putting in the store, pay attention to the tags and specify all relevant ones so your game appears in several sections of the website, so that more players find it, etc. In addition to tags that you define by yourself, there are also tags which are put by the players themselves. Moreover, the status of these tags is higher than yours – they are located right below the short description of the game in the most conspicuous place. According to our observations, Steam gives the search results based on these custom tags, so let's hope that your vision of the game will coincide with the opinion of the users.

The importance of the community

And since we're talking about users, we would like to add a few words about the influence of the community: The players community is a very important component of success! First, the opinion of the players has an impact on the sales and popularity of the game in general. High ratings and players reviews will allow your game to shine on the top of the popular games list, which in turn opens up opportunities to get more featuring from Steam. In our experience – players on Steam are rather critical in their assessments and will lead you to all the shortcomings and problems that exist in the game. So the worst thing you can do immediately after launching a game on Steam is »to go home and go to bed«. Our recommendation is to take more people from the team and put everyone to communicate with players. We launched »Survvarium« at 9 p.m. and went home at 6:30 in the morning, when we were sure that all the basic questions were answered, problems resolved, the



One very important thing is to care for the community. The team spent three days after the launch just helping the players with problems.

servers worked consistently, etc. And this was followed by three days of communication with the community around the clock. Community Hub should be the second home for you and your support team. To affect the evaluation of players is extremely difficult, but your open communication with them and hard work on the game will make players your allies and not opponents.

Additional measures to gain more players

Also as the game has progressed, we have prepared special Steam Trading Cards, badges and emoticons. Many players like to collect and exchange cards. And adding them to Steam does not require game developers. A pleasant surprise for us was the fact that in the case of Free2Play games, collectible cards on Steam are »random drops« only to those players who have purchased something. This is another argument in favor of the adoption of the instrument.

The next step we plan to introduce is various in-game achievements (we just add an achievement system to the game). It is a powerful tool that allows you to better keep players and we plan to use it on Steam.

Conclusion

Let's come to conclusions. A little over a month has passed since our launch, so it's hard to talk about long-term conclusions, but we can reach some now:

1. Steam provides more random traffic, because the first game session is lower than of players from other sources. However, the behavior of those who came for a second time is similar to the behavior of those from other sources. It turns out that there is just a large share of those who have installed the game, played it, then quit, and this greatly affects the overall statistics of beginners. Why is it so? It is difficult to say, perhaps, the proverbial »disregard« of Free2Play.
2. Steam players who returned to the game a second time subsequently behave like all the other players, and pay exactly the same. So we don't see that Steam brings money only at the start. Steam players buy in-game gold, just like other players.
3. The expensive DLCs on Steam are selling well. Don't be afraid to form expensive sets for Free2Play games. Periodically update the stock of DLC bundles and remove the old ones. Discounts are a useful tool that will allow you to not only increase sales, but also appear in the Specials category which is an additional traffic channel for your game.
4. With regard to the traffic of Steam, at the start we received two times more than we expected. Now it's less than we predicted for the second month. However not the quantity is important but the problem of predicting this quantity. Given the short time since release, we can say our guys from PR are just begin-

STEAM

COMMUNITY MARKET

Buy and sell items with community members for Steam Wallet funds.

Steam > SCAVENGER

SCAVENGER

Steam

Survarium Trading Card

Scavengers, not great fighters. Scouts perform discreet infiltrations into the territory doing their best not to be seen. Snipers are used for combat at long distance, Artifact seekers, aside from the necessary knowledge, possess special equipment to search the often strange and undoubtedly unique formations at anomalous locations.

Some of the survivors, while traveling over land were halted by the elements and were left with no choice but set up camp in the vicinity of a former settlement. There they have seen other in need of help. Thus, in this safe area, people formed a separate group where they had more chance of survival. They became known as Scavengers.

This item is a commodity, where all the individual items are effectively identical. Individual listings aren't accessible; you can instead issue orders to buy at a specific price, with the cheapest listing getting automatically matched to the highest buy order.

After purchase, this item:

- will not be tradable for one week
- cannot be listed on the Steam Community Market for one week

22 for sale starting at \$0.21

Buy...

Price	Quantity
\$0.21	2
\$0.23	2
\$0.25	1
\$0.36	2
\$0.39	2
\$0.42 or more	13

50 requests to buy at \$0.12 or lower

Sell...

Price	Quantity
\$0.12	4
\$0.10	10
\$0.08	2
\$0.07	1
\$0.06	7
\$0.05 or less	26

Trading cards and achievements are vital measures to increase the number of players or to keep those invested who are already in.



Vostok Games was founded by former GSC Game World employees therefore »Survarium« reminds a lot of their best known game »S.T.A.L.K.E.R.«

- ning to »dive into the turbulent currents« of Steam and prepare new experiments.
5. Use tools for featuring important updates to get more traffic.
 6. Technical support will need to actually work with the Steam community because the majority of users will not send tickets to your helpdesk or discuss the game on the official forum.

Oleg Yavorsky

SUPER CARL VON ZÄUNEN UND MÖCHTEGERNHelden



Was von Klippen stürzende Schafe und auf dem Papier entworfene Levels mit dem ersten Mobile Game von Mooneye Studios zu tun haben.



Marco Apel
ist Community Manager
bei Mooneye Studios.

Marco ist dem Team von Mooneye Studios kurz nach dessen Gründung beigetreten. Seitdem ist er als Allrounder für Social Media und PR sowie als unberechenbarer Kickerspieler bekannt.

Mooneye Studios wurde im September 2014 aus der Taufe gehoben. Die vier Gründer hatten sich während des Studiums kennengelernt und dabei festgestellt, dass sie sowohl als Team harmonieren als auch eine identische Philosophie vertraten, was das Entwickeln von Games angeht. Zwei wichtige Aspekte, wenn man beschließt, den Traum von der eigenen Firma in die Tat umzusetzen. Und so wurde aus einer eher zufällig entstandenen Gruppe von Game-Enthusiasten Mooneye Studios geboren.

Jeder Held hat eine Vorgeschichte

Der Grundgedanke für das spätere »Super Carl«-Spiel entstand schon recht früh aus einer Studienaufgabe heraus. Die Anforderungen waren simpel, Vorgabe für das Spiel war nur die Verwendung eines 5x5-Grids. Wir haben uns damals für ein recht simples Spielprinzip entschieden. Als Inspiration diente uns das Flash Game »Cat Trap«. In dem Spiel besteht die Herausforderung des Spielers darin, auf zufällig generierten Spielfeldern eine Katze durch taktisches Platzieren von Blöcken einzufangen. Ein Prinzip, das uns gut gefallen hat und sich auch gut auf die 5x5-Grid-Anforderung umsetzen ließ. Was jetzt noch fehlte, war eine passende Spielfigur.

Dabei kam uns der Zufall zu Hilfe. Zum damaligen Zeitpunkt ging ein Bericht über sich von Klippen stürzende Schafe durch die Medien. Davon inspiriert entstand »Suicide Sheep« – ein durch und durch liebenswertes, aber auch etwas dämliches Schaf. Mit diesem Hauptcharakter war ebenso der Projekttitel gefunden.

Die technische Umsetzung war nicht sehr zeitaufwändig, eine spielbare Version für Android und HTML5 benötigte nur ein paar Tage Arbeit. Sie lief flüssig und hat damaligen Testern bereits großen Spaß gemacht. Als wir uns dazu entschieden, dass Mooneye Studios weit über ein Studentenprojekt hinausgehen sollte, war klar, dass »Suicide Sheep« einen Relaunch brauchen würde. Es sollte als eines



»Suicide Sheep« war die Grundlage für »Super Carl« und entstand während des Studiums.

unserer ersten Spiele professionell gepubliziert werden. Das stellte uns natürlich vor einige Herausforderungen.

Die Geburt von Super Carl (Development)

Mit der Absicht, das Spiel professionell aufzubauen und zu vermarkten, war einer der ersten notwendigen Schritte, sich vom Titel »Suicide Sheep« zu verabschieden. Dies hatte unterschiedliche Gründe. Zum einen gab es bereits eine App mit dem gleichen Titel, zum anderen wollten wir eine sehr breite Masse mit unserem Spiel ansprechen. Der Name hat zwar einige Lacher produziert, allerdings erschien er uns auf lange Sicht nicht ideal. Insbesondere, da wir mit dem Spiel auch eine jüngere Zielgruppe ansprechen wollten. Zusätzlich befürchteten wir durch das Wort »Suicide« eine negative Vorverurteilung unseres Spiels. Außerdem war es uns wichtig, eine Storyline in das Spiel einzubauen, um es attraktiver zu gestalten. Und so wurde aus dem kleinen Schaf, das völlig unmotiviert in den Abgrund läuft, Super Carl – das Schaf ohne wirkliche Superkräfte, aber mit der Mission, die Welt zu retten. Was den Spieler wiederum vor die Aufgabe stellt, Super Carl vor sich selbst zu schützen. Wir entwarfen einen kleinen Comicstrip, der am Anfang des Spiels gezeigt wird und die Hintergrundgeschichte von Super Carl erläutert.

Mit der Entwicklung des neuen Hauptcharakters wurde auch klar, wie die Überarbeitung des Spieldesigns auszusehen hatte. Aus der teilwei-



In einem 5x5-Grid muss der Spieler das Superheldenschaf Carl einfangen, bevor er sich von der Klippe stürzt.

se etwas dunkleren Levelgestaltung des Vorgängers sollte ein Comic-ähnlicher Look entstehen. Bunt, witzig, niedlich und casual – passend zum Comicstrip und dem neuen Erscheinungsbild der Spielfigur. Ein Design, das nach unserer Vorstellung ideal mit unserer Absicht, eine breitere Masse anzusprechen, funktionieren würde.

Die Mechanik hinter einem Superhelden (Game Design und Mechanik)

Was nach der Überarbeitung des Designs noch übrig blieb, war die technische Seite des Relaunchs. Ein Bereich, der im Vergleich zum Design deutlich mehr Arbeit gemacht hat als erwartet. Einer der ersten Schritte war der komplette Relaunch des Spiels mit Unity, um mehr Plattformen mit »Super Carl« erreichen zu können. Dadurch haben wir uns die Möglichkeiten offengehalten, das Spiel zu einem späteren Zeitpunkt auch für iOS und Microsoft-Betriebssysteme zu veröffentlichen.

Daneben gab es noch andere Gründe, warum wir uns für Unity anstatt für HTML5 entschieden haben. Es gibt eine Vielzahl an vorgefertigten Plug-ins für Unity, die uns eine Menge Arbeit erspart haben (zum Beispiel für den Ingame-Shop). Außerdem hatte ein Großteil des Teams bereits Erfahrungen mit Unity sammeln können, sowohl im Rahmen des Studiums als auch durch erste Auftragsarbeiten. Damit war die Wahl von Unity nur logisch.

Es sollte sich allerdings herausstellen, dass das Arbeiten mit vorgefertigten Plug-ins ein zweischneidiges Schwert sein kann. So hatten wir große Probleme mit dem von uns verwendeten Facebook-Plug-in, denn bei zweimaligem Einloggen stürzte es regelmäßig ab. Der Versuch, diesen Fehler zu beheben, brachte uns an die Grenzen unserer Möglichkeiten. Wie sich jedoch herausstellte, lag der Fehler nicht bei uns, sondern beim Anbieter des Plug-ins. Nachdem wir Kontakt mit ihm aufgenommen hatten, konnte er uns mit einem externen Bugfix weiterhelfen.

Zum damaligen Zeitpunkt erschien es uns am sinnvollsten, »Super Carl« vorerst nur für Android zu veröffentlichen. Eine Entscheidung, auf die wir heute immer noch häufig angesprochen werden. Die Frage nach dem Grund ist allerdings schnell beantwortet: Wir wollten das Spiel zeitnah veröffentlichen und unsere Kernkompetenzen lagen ganz klar bei Android. Das lag daran, dass Android ein fester Bestandteil des Studiums war. Daneben hatten wir schon bei verschiedenen kleineren Projekten Erfahrung mit dem Android-System gesammelt. Mit iOS dagegen hatte keiner von uns zum damaligen Zeitpunkt nennenswerte Erfahrungen machen können. Es wäre zweifelsfrei möglich gewesen, »Super Carl« auch für iOS zu optimieren, aber dies hätte Zeit gekostet, die wir nicht bereit waren zu opfern.

Insgesamt hat die Überarbeitung der technischen Komponenten mehr Zeit in Anspruch genommen als anfangs geplant. Vor allem das



Der Comic am Anfang des Spiels erzählt die Geschichte des Heldens Carl ohne Superkräfte.

Balancing der einzelnen Levels war zeitintensiv. Weil sich die Levels bei »Super Carl« nicht zufällig generieren (wie zum Beispiel bei »Cat Trap«), sondern von uns entwickelt wurden, musste jeder Level einzeln auf eventuelle Fehler und auf die generelle Lösbarkeit überprüft werden. Erschwerend kam hinzu, dass wir in jedem Level drei verschiedene Lösungswege (drei mögliche Medaillen) durchtesten mussten.

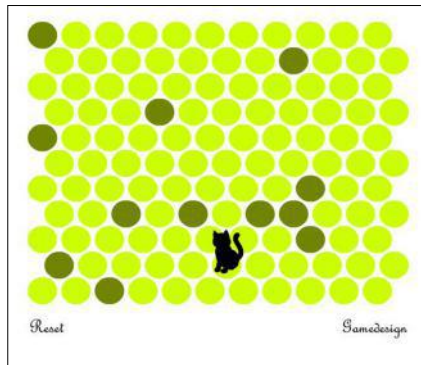
Neben dem Balancing hat auch das Erstellen des Tutorials viel Zeit gekostet. Die Mechanik



Bei der Überarbeitung der Grafik entschied sich das Team für einen bunten Look, der an Comics erinnern sollte.

hinter dem Bewegungsverhalten von »Super Carl« ist relativ komplex. Carl sucht sich im Uhrzeigersinn immer den kürzesten Weg von seiner aktuellen Position zum Abgrund. Nach jedem Zug des Spielers verändert sich dieser Weg. Dadurch wird ein gewisses taktisches Denken gefordert. Wir hatten bereits in frühen Tests festgestellt, dass nicht jedem sofort klar war, wie und warum sich das Schaf bewegt. Insbesondere die Tatsache, dass Carl seine Richtung nach jedem Zug ändern kann, schien den Testern Schwierigkeiten zu bereiten. Wir mussten die Spielmechanik folglich so kurz, aber gleichzeitig auch so erklärend wie möglich in einem Tutorial verpacken.

Weil wir uns bei der Monetarisierung für das Free2Play-Modell entschieden hatten, mussten wir noch einen Ingame-Shop implementieren. Weil wir bisher keinerlei Erfahrungen mit Shop-Einbindungen hatten, mussten wir auch hierfür etwas mehr Aufwand betreiben als gedacht. Es gelang uns einfach nicht, eine funktionierende Shop-Version einzubauen. Wie sich herausstellte, hatte sich im Cache ein altes Passwort zur Shop-Anmeldung festgesetzt. Die Lösung unseres Problems war also ein simples Löschen des besagten Caches. Bis wir allerdings darauf kamen, waren drei Tage vergangen und unser Programmierer war mehrmals kurz davor, seinen Rechner aus dem Fenster zu schmeißen. Die einfachsten Lösungen sind manchmal die richtigen – eine Lektion, die wir unter »Learning by Doing« abgespeichert haben.



Das Flashspiel »Cat Trap« mit zufallsgenerierten Leveln diente als Inspiration für »Super Carl«.



Die Implementierung des Ingame-Shops sorgte durch ein im Cache gespeichertes Passwort für Probleme.



Der Trailer zum Spiel zeigt den umhangtragenden Carl in klippenspringender Aktion.

Mechanik. Weil wir kein Freund der Pay2Win-Methode sind, haben wir sehr darauf geachtet, dass unser Monetarisierungsmodell keinen Einfluss auf das Gameplay hat. Egal ob sich der Spieler entscheidet, Geld im Spiel auszugeben oder nicht, die grundlegende Spielmechanik wird dabei nie beeinflusst. Wir haben bewusst in Kauf genommen, dass der Spieler zu einem gewissen Zeitpunkt aufhören muss zu spielen, damit sich seine Leben regenerieren können. Allerdings haben wir die Möglichkeit integriert, dass sich der Spieler Werbevideos ansehen kann, um damit Ingame-Währung oder direkt Leben zu bekommen.

Ein Superheld wird berühmt (Marketing)

Wie bereits vorher angesprochen, steht jungen Indie-Teams eher selten ein großes Marketingbudget zur Verfügung. Wir sind da sicherlich keine Ausnahme. Es stand von vornerein fest, dass wir zumindest am Anfang »Super Carl« nur durch unsere eigenen Kanäle bewerben und verbreiten würden. Wir steckten einige Stunden in die Produktion eines Trailers, den wir danach über unsere Social-Media-Kanäle verteilt haben.

Neben Facebook, Youtube und Twitter haben wir versucht, »Super Carl« so vielseitig und oft wie möglich im Netz sowie verschiedenen Foren zu platzieren. Zusätzlich hatten wir einen wertvollen Kontakt zum amerikanischen Indie Game Magazine, das monatlich über unser Hauptprojekt »Lost Ember« berichtet hat. Dort konnten wir »Super Carl« als Zwischennote einbringen. Daneben hatten wir die Möglichkeit, die Quo Vadis 2015 in Berlin zu besuchen und dort unsere Projekte zu präsentieren. Trotz unseres No-Budget-Plans mussten wir an verschiedenen Stellen Geld in die Hand nehmen. Zum einen haben wir für die Quo Vadis Unmengen an Flyern mit Promotioncodes für »Super Carl« drucken lassen, zum anderen haben wir gezielt Werbeanzeigen auf Facebook geschaltet. Das Ergebnis lag allerdings weit unter unseren Erwartungen. Wir konnten keine signifikante Veränderung unserer Downloadzahlen feststellen. Den größten Erfolg hatten wir durch ein Youtube-Review, das eine Steigerung der Downloads nach sich zog. Allerdings blieben die meisten Anfragen für Reviews unbeantwortet oder uns wurde

direkt eine Preisliste zugeschickt. Natürlich spielt man immer mit dem Gedanken, sich zu-
mindest einmal irgendwo »einzukaufen«, um
dann die entstandene Aufmerksamkeit nutzen
zu können. Allerdings hatten wir dafür einfach
nicht das Geld und wir wollen bei unseren
Games durch Qualität überzeugen, selbst wenn
das der härtere Weg ist.

Das mit Abstand erfolgreichste Marketing-
mittel war unser Stand auf der Quo Vadis.
Neben dem direkten Kontakt zu potenziellen
Spielern haben wir dort auch erste Kontakte
zu Publishern knüpfen können. Wir werden
auch in Zukunft versuchen, an so vielen
Messen wie möglich teilzunehmen. Beim per-
sönlichen Gespräch bekommt man einen viel
besseren Eindruck davon, wer die tatsächliche
Zielgruppe für das eigene Spiel ist. Obwohl
unser erster Versuch, Werbeanzeigen bei-
spielsweise über Facebook zu schalten, eher
schlecht gelaufen ist, gehen wir davon aus,
dass wir mit unserem jetzigen Kenntnisstand
bei einem zweiten Versuch deutlich bessere
Ergebnisse erzielen werden. Eine der wichtigs-
ten Lektionen, die wir bereits jetzt gelernt ha-
ben, ist, dass es ganz ohne Marketingbudget
verdammt schwierig ist, die Aufmerksamkeit
auf das eigene Spiel zu lenken.

Springt er oder springt er nicht? (Fazit)

Wenn man uns fragen würde, was unsere
abschließende Zusammenfassung für »Super
Carl« ist, dann würden wir sagen: Es gibt
keine. Allein schon, weil »Super Carl« erst
knapp einen Monat auf dem Markt ist. Wir
können bislang nicht abschätzen, welchen
Weg unser kleines Schaf gehen wird. Dazu ist
noch zu viel ungeklärt. Wir sind im Gespräch
mit verschiedenen Publishern und hoffen na-
türlich, dass dabei am Ende etwas rumkommt.
Was wir aber sicherlich teilen können, sind
Erkenntnisse, die wir gewonnen haben.

Egal wie begeistert alle im Team von einer
Idee oder Mechanik sind: Es ist wichtig, dass
die Leute, die hinterher das Spiel in der Hand
halten, verstehen, wie es funktioniert. Wir
sind immer davon ausgegangen, dass wir
einen guten Weg gefunden haben, die recht
taktisch ausgerichtete Mechanik hinter
»Super Carl« zu erklären. Aber nachdem
das Spiel jetzt veröffentlicht ist und wir auf
Messen beobachten konnten, wie Leute unser
Spiel spielen, müssen wir zugeben, dass noch
Verbesserungspotenzial besteht. Wir haben
unterschätzt, wie viele Spieler nach dem Trial-
and-Error-Prinzip vorgehen. Bei einem Spiel,
dessen Mechanik mit dem Verlust von Leben
funktioniert, kann dies zu Frustrationen
führen. Wir werden zu einem späteren Zeit-
punkt versuchen, das Tutorial noch verständ-
licher zu gestalten und die erforderliche takti-
sche Herangehensweise zu verdeutlichen.

Außerdem sollte man nie davor zurück-
schrecken, einen eigenen Level Editor zu



Der Stand auf der Quo Vadis brachte dem Team von Mooney Studios erste Kontakte zu Publishern.

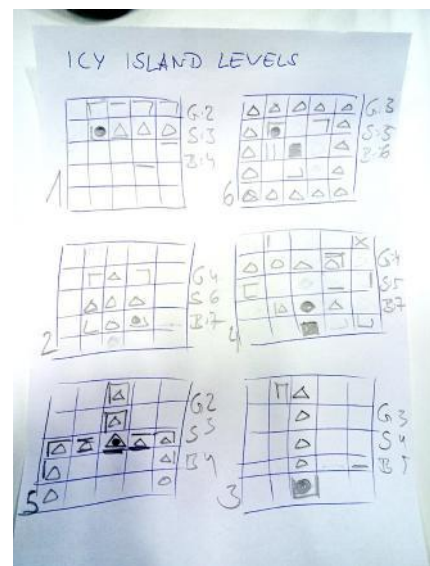
schreiben. Bei »Super Carl« haben wir die
einzelnen Levels noch mit Stift und Papier
entwickelt. Ein Aufwand, den wir so niemals
wieder betreiben würden.

Generell gilt: Der Markt für Mobile Games
ist riesig, jeden Tag werden Hunderte Games
von nicht weniger talentierten Indies ver-
öffentlicht. Um sich dort positionieren zu
können, braucht es Vorausplanung, ein Budget
und nicht zuletzt auch etwas Glück. Trotz-
dem haben wir nicht das Gefühl, in einem
sprichwörtlichen Haifischbecken zu schwim-
men. Der Zusammenhalt unter den vielen
Indie-Studios ist großartig und man kriegt
hilfreiches Feedback oder nützliche Tipps.

»Super Carl« war unser erstes eigenes
Projekt, das wir veröffentlicht haben. Es war
absehbar, dass dabei nicht alles nach Plan
laufen würde, aber wir haben aus unseren
Fehlern gelernt und hoffen, dass auch andere
aus unseren Fehlern lernen werden. Bei Mo-
oney Studios arbeiten wir zurzeit an unserem
Hauptprojekt »Lost Ember« und zwei weiteren
Mobile-Titeln und sind so motiviert wie nie
zuvor. Wir machen Games, denn das ist unse-
re Leidenschaft!

Wir haben nie den Punkt erreicht, an dem
wir keinen Spaß mehr daran hatten, was wir
machen. Und das wird hoffentlich auch nie
der Fall sein.

Marco Apel



Alle Level wurden ohne Level Editor, dafür aber mit Stift und Papier entwickelt.

UNREAL VS. UNITY

WHICH ENGINE IS BETTER FOR MOBILE GAMES?

The team from OnlineFussballManager is currently working on a new mobile game. But before the development could start full steam they had to decide which engine to base their game on - a hard decision that made some prototyping necessary.

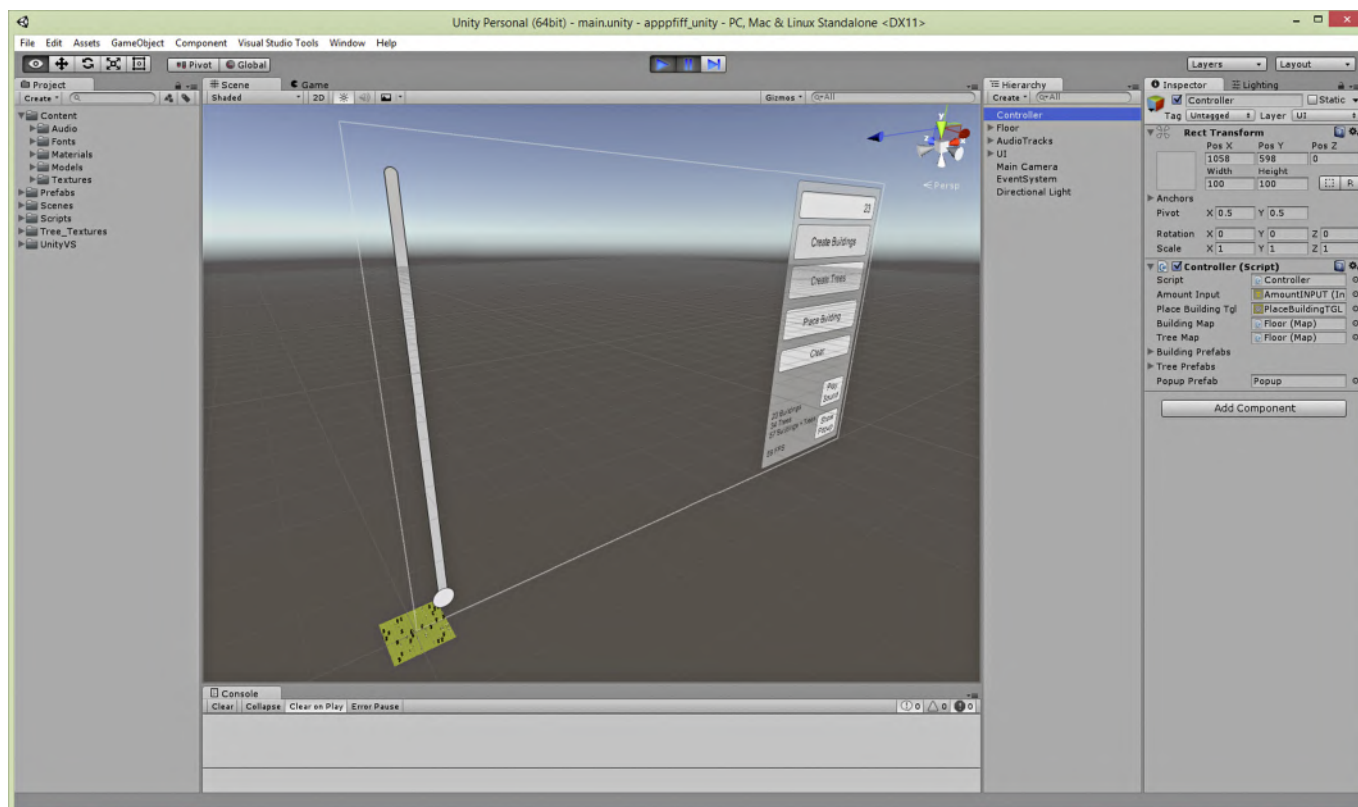
Starting a new project is always an exciting time. It is when your idea finally starts to take shape. We (OnlineFussballManager GmbH) are now commencing development on our new project for mobile. The vision is to create an exciting and unique combination of a building-up strategy and a football management game.

Players will be able to plan and create their own club grounds and facilities and manage tactics and training of their team. The team's players will not just be jerseys with numbers but real personalities with faces and traits to remember. Everything will take place in an

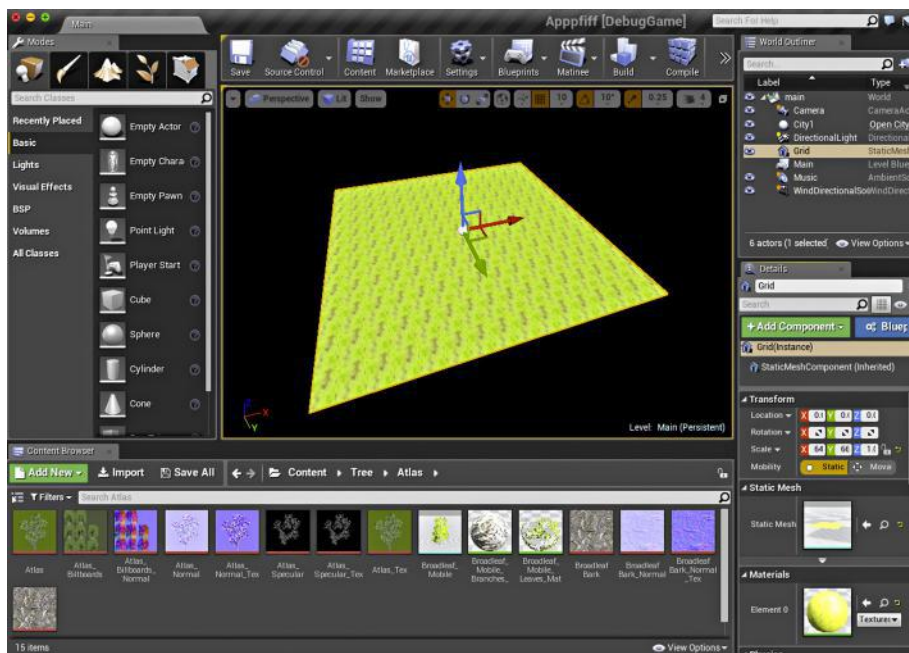
isometric view of the club's premises, featuring animated buildings on a defined grid.

So, when developing the game's vision, we arrived at a crucial point: Given our game design requirements and the fact that we are going to develop this title for iOS and Android - how are we going to set up the project technically?

One of the first considerations was whether to create an in-house engine or to use an existing one. Creating an engine yourself is of course the best way to go if you want to retain 100 percent control of what is happening in your source code, your toolchain and your earnings from the game. The downside, however, is that it will take a significant amount of effort to make everything work. This idea



The image shows the project directly inside the Unity editor. One big advantage of the editor is that you can extend it by your own scripts.



This is how the project looked inside the Unreal editor, which has a lot of specialized editors inside of which some can replace full programs.

was quickly dropped, as we wanted to keep the time and costs within reasonable boundaries.

The next step was to decide which existing engine to use. There were numerous possibilities, ranging from Stencyl, GameMaker over Cocos2d and Marmalade to Unreal and Unity.

Obviously there are a lot of opinions out there whether this or that engine is better and for what reasons. When discussing with the team, there was a lot of love for the Unreal Engine. However, every time we dug deeper in whether this actually WAS a good idea, no one could answer this without resorting to pure opinions. So we decided to get real facts on what would be the best engine for our project, and started to research some of the more prominent engines.

GameMaker was soon dropped as it was more geared towards beginners of game development, thus offering a great and easy-to-learn framework but being way too restricted for what we wanted to do.

Next up was Cocos2D, which was promising at first. As the name suggests, Cocos2D is specifically geared towards games in 2D. It rose the question whether we wanted to create our isometric building grid in true 2D or in actual 3D with a fixed viewpoint. After a bit of extra research in this side-ways direction we decided to go for a 3D implementation. This, apparently, made Cocos2D obsolete for our project.

We turned our attention to Marmalade. Games like »Plants vs. Zombies« and »Godus« have been created with it. And though we found a lot of positives about this engine there were some issues which lead us to consider other engines instead. One of the main issues was the comparatively small community and the resulting dependence on the Marmalade team, which would prove too much of a risk for us.

So the two biggest names remaining were Unreal and Unity. And at this point, we couldn't reach any reasonable conclusion by just research alone. Luckily, GDC San Francisco was just around the corner. So we took the chance to fly over there and to talk to some of the guys directly. We met with Epic, took a direct look at their engine and tools and discussed with them what their thoughts were on which engine we should use. The guys were really nice and basically told us: »Unreal is great stuff, but so is Unity. You guys just have to try it out.« We then went on to talk to the guys from Unity we took a look at Unity 5. In the end we asked them the same question and got the same answer.

At this point we decided to do a proper technical research, deciding once and for all which engine would be better suited for our project. We planned a prototype which should resemble the project in a reasonable manner and to check out the build process and the performance on a couple of mobile devices. We wanted to try for ourselves and see where all the pitfalls and issues lay when using both engines.

Since all of our in-house programmers were bound in the development of our other project we had to look outside for help in creating and checking our prototype. This is when we reached out to Bit Barons and started a successful collaboration. Their work on the prototype and the results they created were extremely enlightening and we will now share their report.

Bit Baron's Engine Research

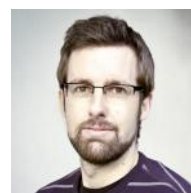
Up until now OFM's sole development focus had been on the creation of web based browser games. We at Bit Barons knew we would be able to help them find what they were looking for. The plan was to create a test application in

Salomon Zwecker
is Programmer
at Bit Barons.



Salomon studied Game Design at the Media Design Hochschule in Munich. He worked at Realmforge Studios as programmer on two retail games. After experiencing the difficulties with own game companies, he joined Bit Barons in 2012. At Bit Barons, he worked on several mobile games utilizing the Unity engine and before that, he worked with XNA and MOGRE. He always preferred to use the beautiful programming language C#.

Daniel Wilkins
is Programmer
at Bit Barons.



Daniel Wilkins has been programming and developing games on the original Unreal Engine since he was 13. Although he initially had his heart set on a music conservatory, he abandoned this path to develop games professionally. After interning at NovaLogic, he did freelance development in Unity and Unreal and programmed and released several games with his own company, DAJMAG. After realizing that programming was much more engaging than the business side of games, he left his company and moved to Munich and joined Bit Barons as a programmer in early 2015. He is eternally thankful that Epic Games integrated C++ directly into their engine.

Guido Schmidt
is Lead Game Designer
at OnlineFussballManager.



Guido Schmidt has been in the industry for over 12 years. He started out as a programmer for mobile games, but quickly transitioned into a more design specific role. He was working for HandyGames and created games like »Townsmen« and »Frozen Front«. His greatest successes include the browser game »The Settlers Online« and the PC game »Anno 1404« when working for Related Designs and later Ubisoft Blue Byte. After over seven years of working on the »Settlers«- and »Anno«-franchise he decided to turn his attention to the mobile market again and join up with the OnlineFussballManager GmbH to create new and unique gaming experiences for players.



Guido Schmidt just recently joined OnlineFussballManager and is now designing the mobile game.



As all team members at OnlineFussballManager were involved in projects already the company decided to have Bit Barons do the evaluation of Unreal Engine and Unity.

OnlineFussball-Manager

OnlineFussballManager GmbH is the developer and publisher of the browser game OnlineFussballManager that was released in 2003 and has currently about 150,000 active players (April 2015).

In the beginning the game was a project by some students and a lot of features from those days are still in the game today. Over time OnlineFussballManager became the most successful soccer manager in German-speaking countries and with that the company OnlineFussballManager GmbH was founded in 2006.

Since its launch the game won several awards and was even nominated as the best browser game at Deutscher Computerspielpreis 2009.



OnlineFussballManager GmbH

those two competing engines so that they knew which one would be best for their exact needs. To be honest, we were excited and curious ourselves to see what the results would show about the current state of mobile development on these engines. There are often reports written detailing the differences between them, but ours would be different in that we were creating and benchmarking a nearly identical test application. Creating the same app twice allowed us to test both on an even playing field and to share results with OFM which engine had performed better and highlight problems we had during development. Salomon Zwecker headed up the Unity project development. He had quite substantial experience with Unity, having used it for several years. He would program his application in C#, a powerful language that is integrated seamlessly into Unity. Daniel Wilkins was in charge of the Unreal development. Daniel had been using Unreal since 1999. He would be using mainly C++ and integrate the game functions with blueprints. Salomon's and Daniel's knowledge in their respective engines ensured that again the results would come from an even playing field.

Test Case

We wanted to create a benchmarking test that would provide OFM with results that were directly related to the type of game they were making. They had specified that they wanted several animated buildings and trees in their prototype so we built a 3D scene where those objects could be placed onto a map by the user. The grid was 11 by 16 and had space for 176 buildings. Each grid space supported up to six trees for a total of just over 1,000 trees. We added a simple UI with which you could add a specified number of trees and buildings to the scene and additionally, we added functionality to add buildings to a specified location by clicking on the map. For organizational purposes, we built the system on a flat plane with a grid and used an overhead camera to display the scene. We added camera functionality to allow the user to zoom and pan the camera. Because this prototype was being used to decide which engine to develop with, we made every effort to keep the appearance of the scene the same. This took a bit of tweaking as Unreal and Unity handle some things differently.

To keep the performance testing uniform, we wanted to use identical tree and building models on both systems. For the trees, we used a SpeedTree mobile model which had right around 1,000 polygons and allowed us to get a good feeling for how small increments in the displayed triangles affected the frame rate. After having a hard time finding a model for the animated buildings which worked in both engines, we ended up using two different ones. Both models had just over 16,000 polygons and had a nearly identical material setup. We disabled any Level Of Detail (LOD) to ensure

both engines would display the same amount of triangles at any camera distance. The test was designed to not only show the performance differences but also the difference in rendering quality. We also had to pay close attention to the default shaders in the Unreal Engine. After noticing that Unreal looked remarkably better, we realized that by default the camera had enabled a number of performance expensive shaders, which we disabled. The lighting was another area in which we had to spend some time getting it right.

To give a sense of the development process we also compared things like build time, available documentation, the difficulty of mobile deployment and in code iteration. These are areas we knew where there would be differences.

Unity Project

The Good

- Unity is all about Entities («GameObject») and Components («MonoBehaviour»). Once you understand this concept you can work well within Unity. This system can add a great deal of organization to your project.
- There are many components included in Unity that provide everything you need to make a game – except the game logic itself. This can be something as small as a Plane (not included in Unreal), which was used for our grid system. The latest additions to the engine are the «UI» and «Layout» components for powerful and scale-able GUIs.
- The editor can be extended by your own scripts and there are tons of assets for any purpose available in the Asset Store.
- Unity was one of the first public game engines with mobile support. Therefore, mobile deployment works pretty well and the visual result look very much like it does inside the editor. The systems have been refined and deployment to devices is painless.
- Unity has probably the biggest community of all game engines so it is very likely you get an answer if you have a question. Even though Unity supports multiple languages for scripting, the documentation for each is incredibly robust. Furthermore, even if you find an answer that's in another language, you can still understand the logic behind it.
- Unity did a very nice job of optimizing rendering for multiple objects of the same type. To get similar performance in Unreal, we had to use Instanced Rendering and this is often less flexible than Unity's rendering.

The Bad

- The engine is closed source. If you are not willing to negotiate a price with Unity for the source code, you will have to deal with that. This can be problematic if something is broken and you have to wait for an update.
- The new UI system is pretty good. But rather than having an extra editor for the UI, it is directly inside the scene. When you open the scene and want to edit the UI, you have to

zoom out first unless you want to edit things on the bottom left corner of the UI.

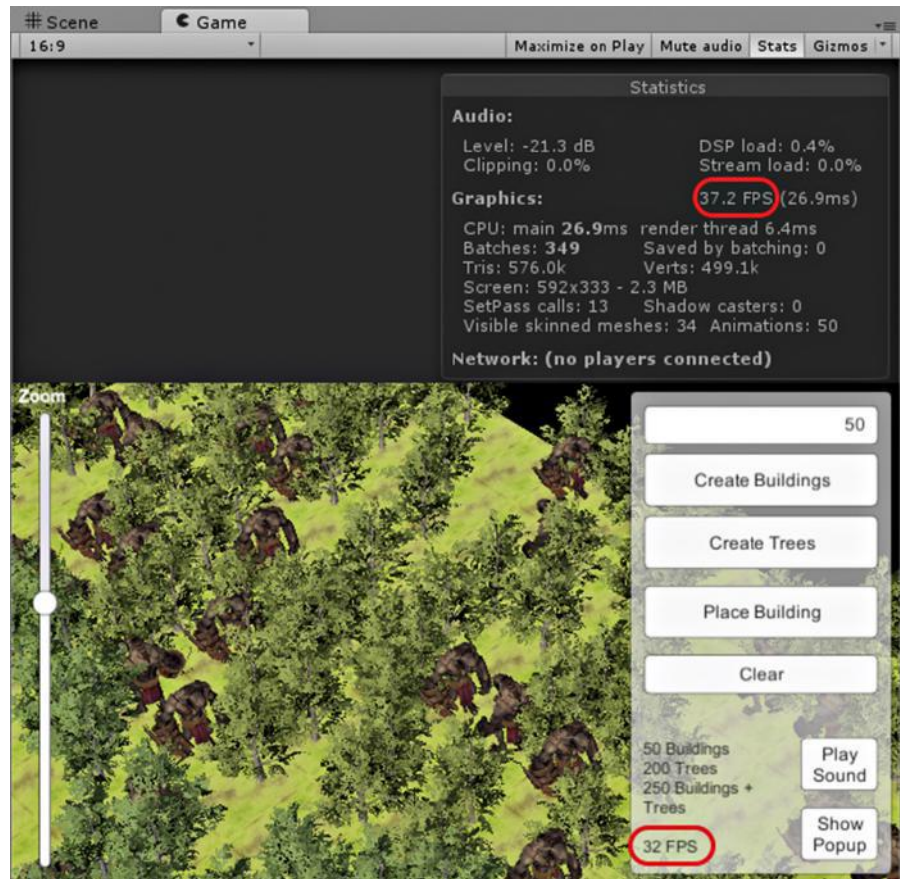
The Ugly

- There were two problems with the new UI system of Unity. When tapping somewhere on the screen, it wasn't possible to determine whether an UI element has been clicked or not. However, if the GraphicsRaycaster class is extended, all required data can be accessed and exposed publicly. After replacing the GraphicsRaycaster in the canvas object it is possible to check if the UI has been clicked.
- The second problem with the Unity UI is about scaling for different display sizes. The canvas component actually has scaling options with some parameters. But it was really hard to preview them and so we ended up deploying to the device several times until we found a satisfying configuration.
- Another odd thing is Unity's stats panel. After implementing an ingame FPS counter we compared the displayed number with the FPS displayed in the stats panel of the unity editor. They were different, because the stats panel shows an estimation of how many frames the game would have if it was stand alone and not running inside the editor. So our FPS counter logic was totally fine.

Unreal Project

The Good

- Unreal is completely free to try. You can have the full editor at absolutely no cost. Unity has a free version, but you need to spend some serious money to access the Pro-version.
- Unreal has a mighty editor with many specialized editors inside. As soon as you are familiar with those sub editors, they support your workflow nicely and often expose information you couldn't see with Unity. Some editors are even powerful enough to be replacements for full programs.
- The engine is shipped with full source code. With that it is possible to dig deeper and understand how certain parts of the engine work. Additionally, you can even fix bugs within the engine or add functionality directly. The community and Epic are constantly improving and updating the codebase.
- The overall visual quality inside the editor is great. The amount of rendering view options (e.g. lighting only or shader complexity) is incredible. There are tons of state of the art shaders shipped with the engine. Unreal's renderer is really one of the best on the market that creates amazingly beautiful scenes.
- Blueprints are a quick way to create something simple and to implement basic game logic. Their integration with C++ is fantastic and you get the sense that Epic really made this decision deliberately as it supports both beginners and expert developers and furthermore, allows them to collaborate.
- The C++ integration in general is great. Hot reloads are fantastic.



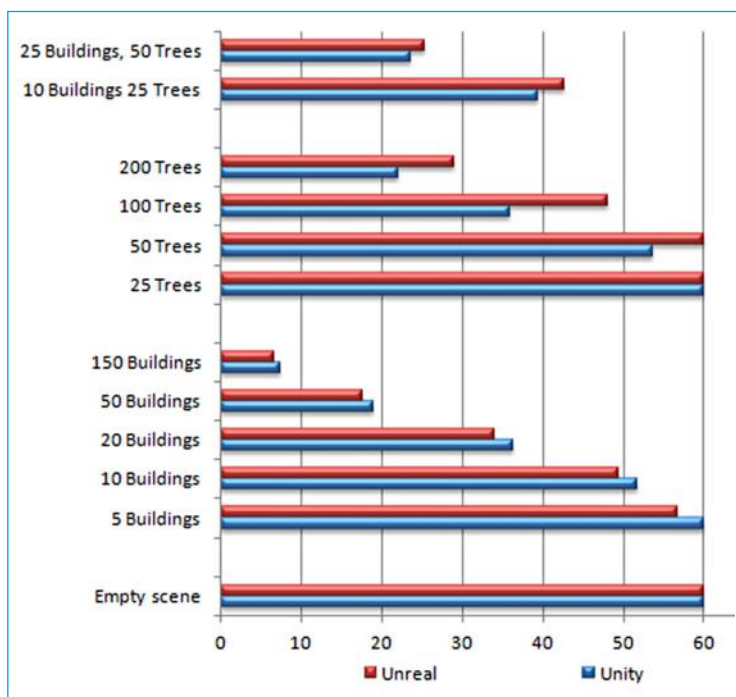
37 vs. 32 FPS: The stats panel in Unity displays the estimated FPS for the stand alone application, instead of the actual data inside the editor.

The Bad

- The Unreal Engine is difficult to get up to speed with. Even if you are an expert in C++, you will need to spend a lot of time learning the various macros and functions.
- Blueprints can very quickly get very messy. A logic consisting of dozens of blueprint nodes that can often be simplified to one or two lines of equivalent code in C++. This isn't usually a problem as you can use whichever you choose, but for some things blueprints are required and can become cluttered.
- The mobile simulator was very inconsistent. It warned us when we would use a shader that was not supported on mobile but even when it validated a shader, it sometimes didn't work. It's a nice thing to have but its inconsistencies leave much to be desired.
- Although Unreal has a big community, we rarely got any answers to our questions. Additionally, nearly all of the provided support is for blueprints. You get the sense that the community is eager to learn and to help but just isn't as comfortable with the engine as is the community with Unity.

The Ugly

- The C++ documentation is seriously lacking. Epic has acknowledged this and is prioritizing their improvement. Due to constant updates you have to be careful when looking at help videos as they may be using an older version of the engine.
- For the GUI, we used the new integrated UI-system, »UMG«. It is based in blueprints and can be very useful. With some work, we



This chart shows the average benchmark FPS of all tested Android devices.

were able to extend a C++ class to reduce the blueprint clutter. However, the system is still buggy and lacks some controls such as toggle buttons. Additionally, documentation for the C++ side is virtually non-existent. Furthermore, we experienced a number of editor crashes while developing the UI. Epic continues to develop this system and we have no doubt it will be fantastic soon.

- Mobile development with Unreal is slow. It takes a long time to deploy to a device. For Android there were some visual issues like blurry textures. But for iOS the issues were much bigger. UE4 supports the ability to build to an iOS device if your project is blueprint only. This is Apple's fault, but we had gone through the entire process of importing the iOS developer keys before it told us this. It's not that Unreal's mobile support is bad, it's just that Unity's is much better.

Benchmark Results: Framerate

Surprisingly, the Frames Per Second (FPS) tests provided very different results on different

devices. On some devices, Unreal performed better in any configuration. On others, Unity beat Unreal in tests that had lots of buildings. Overall, Unreal won the tight race, but the victory came at a price. The visual quality and consistency was much better in Unity. The textures in Unreal looked blurry on some devices and the trees did not look nearly as good. The difference in quality was in part due to the difference between what was displayed in the Unreal editor and mobile preview, and what was actually displayed on the phone. Unity was far more consistent in this regard and what was displayed on each phone was identical to what appeared in the editor preview.

On iOS, both engines scored about the same when it came to displaying animated models. The tests with trees turned out to be worthless however on this device, because the Unreal build did not show any textures on the speed tree models for some reason. We should also point out that you have to have access to an Apple computer if you want to deploy from these engines. OFM also wanted us to provide them with engine benchmarks on Windows Phone. Unfortunately, Unreal does not have the ability to deploy to Windows phones, yet. Therefore Unity did win here by default. Epic Games has said that Windows devices are likely to be added in the future, but they are not a priority due to their small market share currently.

Ultimately, the rendering benchmarks showed just how close these two engines are. This put extra importance on our reports about the pros and cons of each engine.

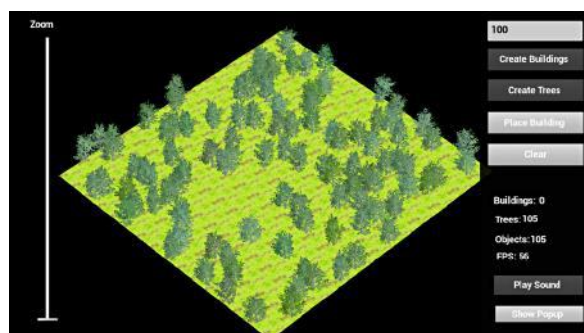
Other Benchmarks

Here are some additional benchmarks we got from our tests:

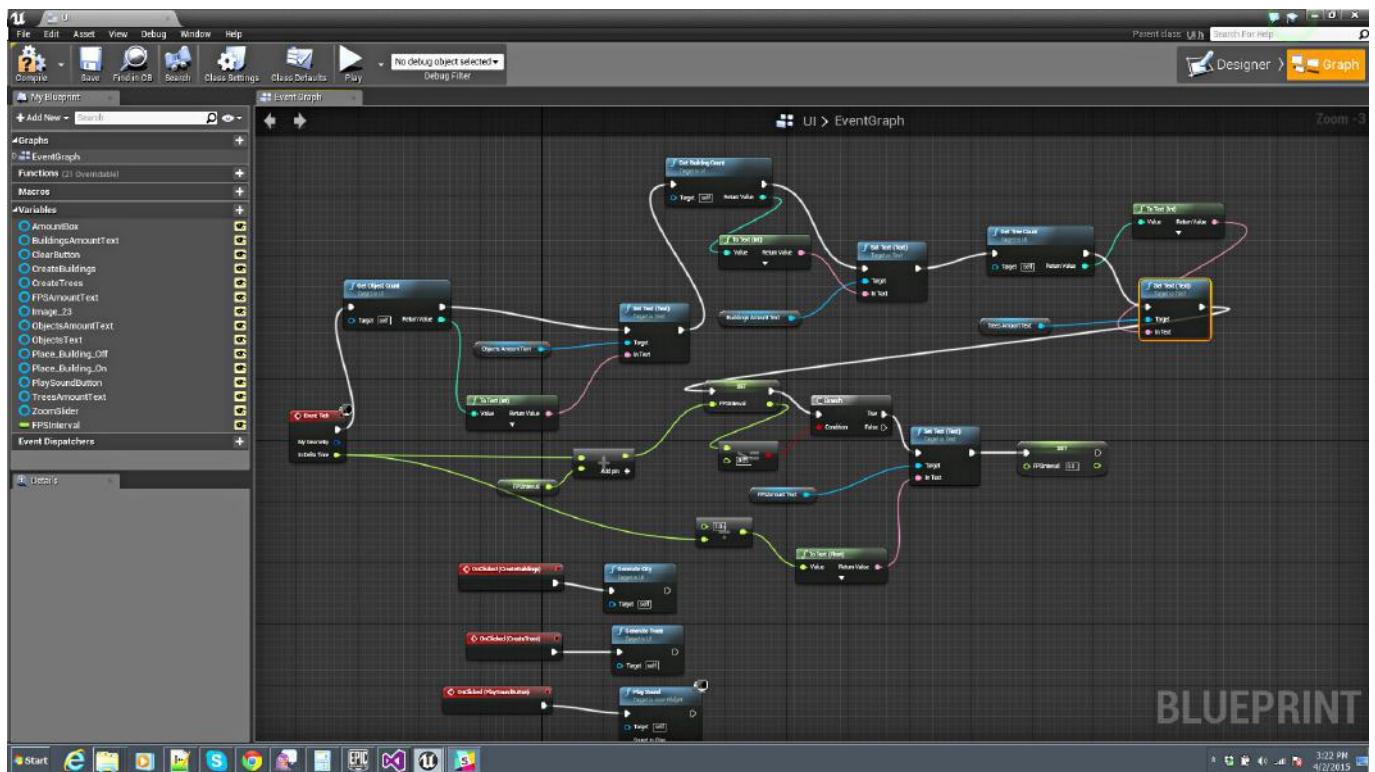
- There was not much of a difference between both engines when it came to memory usage. On Android, Unity was a bit better, on iOS Unreal was better.
- The Unity project had a significantly smaller file size (Android: 42 MB / iOS: 100 MB) than the Unreal project (Android: 101 MB / iOS: 173 MB).
- Unity is about three times faster when deploying to a device. Also compilation of source code is much faster in Unity.



This screenshot shows the prototype project in Unity, displaying 200 trees on the map.



The same shot in Unreal Engine with default mobile settings without lighting for the trees.



Blueprints are cool for simple things and their integration with C++ is fantastic. For more complex logic they quickly become messy and hard to understand.

- In case of battery drain, Unreal was the winner. After two hours with 150 animated models on the screen, the battery drain with Unreal was about 42 per cent on Android and 36 per cent on iOS. Unity consumed about 72 per cent on Android and 47 per cent on iOS.

Conclusion

In review, the results of our testing showed us a lot of the things we liked about each engine. It also uncovered a lot of areas in which these engines can improve to provide a better experience for developers. No engine was remarkably better than the other one and with the ever changing features and addition of support, our results could very well be obsolete in months. The rendering tests did show that the performance is remarkably similar. Again, this puts extra importance on the user experience and how important user friendly and intuitive development can be. Despite our initial prediction that Unreal Engine would be the winner, Unity turned out to be the better choice currently for mobile game development. The major factors that contributed to this decision were:

- Unity was visually more consistent on each platform and provided problem free one-click deployment to all the devices.
- Unity had a much smaller footprint on each device. A smaller install is important on the Google Play Store where APKs are split if they are larger than 50mb.
- Unity is much easier to learn and to understand. and has a large community.
- There is much less iteration time in Unity (faster deployment and faster compilation of source code).

We want to make sure that our results are understood in the context of this prototype. What works for one company might not work for another. There are also many more game engines, some which might work for your specific purpose. For some companies with the time and resources, it even proves beneficial for them to write their own game engines. Do yourself a favor and experiment with as many as you can.

Wrapping Up

Interestingly enough, even the guys at Bit Barons would have advised us to use the Unreal Engine for our project before the actual prototype. Combined with the initial in-house favor of the Unreal Engine we might have taken a sub-optimal decision there. It is, of course, some effort to define, create and test a prototype just to answer the question whether to take this or that engine. But it is a crucial one.

Finally, we checked some additional factors which would influence our decision to one or the other direction. Namely, how easy is it to find experienced people for either engine and what are the available business models?

In terms of people we contacted experienced recruiters to get some real numbers. It turned out that, at the moment, you will find four Unity programmers for every single Unreal programmer. And in terms of business model, the Unreal engine doesn't take an initial, fixed fee, but royalties while Unity is the other way round. Those were both important factors, which made the choice for Unity in our case favorable.

So in this light, we were very happy to have made this prototype and base such an important decision on the solid foundation of facts.

Daniel Wilkins, Guido Schmidt,
Salomon Zwecker

Bit Barons

Bit Barons is an independent game development studio from Munich, Germany. Amongst others the team made »Astroslugs« for PC, Mac, iPhone and iPad or the tablet version of the board game classic »Scotland Yard« for Ravensburger Digital. Currently the team is working on the survival game »Submerge« that is developed by Ice Bird Studios, a collaboration of Bit Barons and Aesir Interactive.



VOM KINDERBUCH ZUR SPIELE-APP



Mixtvision und Mimimi Productions haben sich daran gewagt, eine der bekanntesten Janosch-Geschichten als Spiele-App umzusetzen und berichten davon, wie schwer es ist, Kinder und Eltern gleichermaßen zu unterhalten.



Benjamin Feld
ist Geschäftsführer
von mixtvision Digital.

Benjamin ist bereits seit über einem Jahrzehnt in der Medienbranche tätig. Unter anderem arbeitete er als Produzent für Super RTL und EM.Entertainment. Zuletzt war er Leiter der digitalen Geschäftsentwicklung bei Studio 100 bevor er 2014 zu der in München ansässigen mixtvision Mediengesellschaft wechselte und dort die Geschäftsführung des Digitalbereichs übernahm. Schwerpunkte von mixtvision Digital sind die Entwicklung, Produktion und der Vertrieb von hochwertigen Apps, E-Books und Bilderbuchfilmen.



Dominik Abé
ist Gründer und Game Director
bei Mimimi Productions.

Dominik ist mit der Tigerente aufgewachsen als er noch mit Holzspielzeug spielen musste. Nach über zwanzig Jahren hat er endlich die Chance bekommen, sie mit in die neue digitale Welt zu nehmen. Als Gründer und Game Director bei Mimimi Productions hat er alles dafür gegeben, dass sich die Tigerente in ihrer neuen Welt wohlfühlen kann.



Angelika Achter
ist Studio Managerin
bei Mimimi Productions.

Angie unterstützt das Mimimi-Productions-Team seit Oktober 2014 und übernimmt Aufgaben in den Bereichen Studio-Management, PR und Biz-Dev. Zuvor war sie als Sichterin bei der USK tätig und achtet nun unter anderem darauf, dass die Spiele von Mimimi Productions keine kleinen Kinder verschrecken.

Making Games Wie kam ursprünglich die Idee auf, ein Tablet-Spiel zu »Oh, wie schön ist Panama« zu entwickeln?

Benjamin Feld Als transmedialer Verlag entwickeln wir nicht nur eigene Themen, sondern suchen auch immer nach interessanten Marken und Stoffen, die das Potenzial für Umsetzungen haben. »Oh, wie schön ist Panama« ist ein Klassiker und besticht durch Janoschs außergewöhnliche Illustrationen sowie eine wunderbare Geschichte. Beides ist originell und hat universelle Gültigkeit, sodass das Werk heute noch zeitgemäß erscheint.

Die größte Herausforderung sahen wir darin, die Geschichte adäquat auf ein neues Medium zu übertragen und sie anzureichern, ohne dabei den ursprünglichen Charme zu verlieren.

Making Games Hattet ihr nicht einen gewissen Respekt vor der Marke Janosch beziehungsweise vor der Vorlage?

Benjamin Feld Die Umsetzung bekannter Stoffe gehört quasi zu unserem Tagesgeschäft. Um Erwartungen nicht zu enttäuschen, ist es immer wichtig, Respekt vor der Marke und dem, was die Menschen damit verbinden, zu haben. Gleichzeitig geht es aber eben auch darum, Stoffe zu modernisieren und dadurch neue Nutzergruppen zu erschließen.

Dabei arbeiten wir stets eng mit den Lizenzgebern zusammen.

Angelika Achter Bei uns überwog die kindliche Freude ganz klar im Vergleich zum erwachsenen Respekt (lacht). Eine Vorlage hat vor allem einen tollen Vorteil:

In der Regel ist es ja so, dass die Charaktere in einer Spieleproduktion erst nach und nach entwickelt werden. Bei dieser Lizenz ist das nicht so. Man kennt die Protagonisten schon, verbindet sogar eigene Kindheitserinnerungen damit. Klar, dass da die Begeisterung von Anfang an da ist. Dass jeder das Kinderbuch kennt, fand einer unserer Game Designer

besonders klasse: »Endlich mal ein Spiel, von dem ich meiner Oma erzählen konnte. Und sie hat's auch noch verstanden!«

Making Games Welche Faktoren haben am Ende den Ausschlag dafür gegeben, dass Mimimi Productions den Zuschlag bekommen hat?

Benjamin Feld Für uns waren insbesondere zwei Dinge wichtig: die Qualität des Konzepts, das auf unseren Ideen aufbauen sollte, und das Gefühl, in Mimimi Productions einen Partner gefunden zu haben, bei dem wir der Meinung waren, dass er zu uns und zur Vision des Projektes passt. Dass sie außerdem genau wie wir in München ansässig sind und man sich jederzeit auch persönlich treffen konnte, war trotz Skype und ähnlicher Tools ein nicht zu unterschätzender Pluspunkt.

Das Konzept an sich hat uns überzeugt, weil es am besten unsere Vorstellungen aufgegriffen und sogar an einigen Stellen weitergeführt beziehungsweise konkretisiert hat. Außerdem brachten sie auch eigene Ideen mit ein und äußerten, wo sie Bedenken hatten.

Making Games Inwieweit haben euch, Mimimi Productions, die Erfahrungen aus euren vorherigen Projekten beim Pitchen geholfen?

Angelika Achter Prinzipiell gilt das Übliche: erst genau analysieren, was sich der Auftraggeber wünscht – manchmal kann das sogar etwas sein, von dem der Auftraggeber selbst noch nichts weiß. Dann, unter Abwägung von Aufwand und Nutzen, so anschaulich wie möglich erstes Material

und den Entwicklungsplan präsentieren. Damit kann man dem Auftraggeber auch am besten vermitteln, wie ernst es einem mit dem Projekt ist. Das Ausschlaggebende ist, Vertrauen zu schaffen und mit tollen Ideen überraschen zu können. Bei einer so aufwändigen Herangehensweise gibt es natürlich auch einen Nachteil:

»Die kindliche Vorfriede hat den erwachsenen Respekt ganz klar übertroffen.«



Die größte Herausforderung war, »die Geschichte adäquat auf ein neues Medium zu übertragen und sie gleichzeitig anzureichern, ohne dabei den ursprünglichen Charme zu verlieren.«



Die Minispiele orientieren sich häufig an der Buchvorlage: »Im ersten Kapitel geht der Bär fischen, und da hat es sich beispielsweise angeboten, ein Minispiel zu machen.«

Man hat keine Garantie, den Zuschlag wirklich zu bekommen, und hat damit im schlimmsten Fall viel Zeit und Geld für nichts investiert.

Für das Spielkonzept haben wir die vorgegebene Grundidee übernommen und bewusst kleine Teile herausgeschnitten, um den Kern des Spiels stärker zu fokussieren. Dabei ist wichtig, immer klarzumachen, warum etwas Teil des Konzepts ist oder eben nicht. Alle Ideen haben wir aufwändig visualisiert, um zu zeigen, dass wir die Kompetenzen und das nötige Feingefühl für die Umsetzung der Marke und deren Stil haben. Schließlich wurde hier ja auch ein 2D-Buch in ein 3D-Spiel verwandelt, was nicht ohne Herausforderung war.

Making Games Später haben drei Parteien am Projekt mitgearbeitet. Wie genau hat diese Dreiteilung funktioniert?

Benjamin Feld Bei drei Parteien kommt es natürlich immer wieder mal zu unterschiedlichen Auffassungen. Das war bei »Panama« nicht anders, auch weil alle Parteien mit viel Herzblut an dem Projekt gearbeitet haben. Allerdings waren es fast nie die großen Themen, bei denen man unterschiedlicher Meinung war. Dafür wurden die Feinheiten oft intensiv diskutiert.

Die Rollenverteilung war relativ eindeutig: mixtvision fungierte als Produzent und Publisher; die Janosch film & medien AG war Lizenzgeber und agierte als Koproduzent, indem sie auch in den Entwicklungsprozess aktiv integriert waren; und Mimimi war der Entwickler. Ehrlicherweise muss man aber auch sagen, dass sich das Rollenverständnis im Alltag durchaus immer wieder mal verschiebt, wenn alle Parteien mit Elan an dem Projekt arbeiten und versuchen, ihre Vorstellungen mit ein- und auch durchzubringen.

Dominik Abé Wenn drei Parteien zusammenarbeiten, bedeutet das zum einen, dass verschiedene Vorstellungen und Wünsche irgendwie zu einem Ergebnis kommen müssen. Aber es bedeutet ebenso, dass man aus drei Kompetenzbereichen schöpfen kann.

So viel gebündelte Erfahrung ist toll, aber es ist auch nicht ganz einfach, alles unter einen

Hut zu bekommen. Zum Beispiel dann, wenn Gameplay und Geschichte in einem Widerspruch stehen. Im Buch »Oh, wie schön ist Panama« gibt es eine Flaschenpost, die in der Geschichte zweimal vom Erzähler erwähnt wird, die aber beide Male vorbeischwimmt.

Der Spieler dagegen klickt drauf. Bis auf ein leichtes Zappeln der Flasche passiert da aber nichts. Dadurch entsteht eine Diskrepanz zwischen der Erwartung des Spielers und der Geschichte. Der Spieler versteht nicht, warum die Flasche wegschwimmt, obwohl er doch mit ihr interagiert hat. Aus Game-Design-Sicht ist das doof, aber aus Sicht der Marke ist es eben wichtig, die Originalstory nicht zu verändern.

Ansonsten haben wir viele Freiheiten bekommen, neue Inhalte hinzuzufügen, solange die vorgegebene Geschichte unverändert bleibt. Beispielsweise ist Günter Kastenfrosch neu in die App gekommen. Der lustige kleine Frosch kommt in einem Minispiel vor, wo er mithilfe des Spielers einer Kuh hilft, lästige Fliegen loszuwerden.

Making Games Wie haben sich euer Workflow und Abnahmeprozesse mit der Zeit verändert?

Benjamin Feld Man steckt eigentlich immer irgendwo zwischen zwei Extremen. Rein inhaltlich kann man sich zu Tode diskutieren, wenn der Lizenzgeber das Gefühl hat, dass seine Marke nicht bestmöglich behandelt wird. Auf der anderen Seite muss man mit dem Entwickler immer auch die technische Machbarkeit im gegebenen Kosten- und Zeitrahmen im Auge behalten. Beides lässt sich vorher nicht bis ins letzte Detail vertraglich regeln. Und ehrlich gesagt, wenn es im Laufe des Projekts dazu kommt, dass man auf Vertragsinhalte pochen muss, steht das Projekt vermutlich sowieso schon kurz vor dem Scheitern.

Zu berücksichtigen ist allerdings, dass der Lizenzgeber quasi ein Vetorecht hat, wenn er seine Marke gefährdet sieht. Auch hier ist ein gemeinsames Verständnis der Marke und der Vision des Endprodukts sehr hilfreich, um zu vermeiden, dass man sich bei den Entscheidungen gegenseitig blockiert.

»Wenn man auf Vertragsinhalte pochen muss, steht das Projekt vor dem Scheitern.«



Mimimi hatte während der Entwicklung trotz bekannter Marke viele Freiheiten, kreativ zu werden: »Beispielsweise ist Günter Kastenfrosch neu in die App gekommen.«





Vom Konzept (oben) zum finalen Look in der App (unten): »Das Konzept an sich hat uns überzeugt, weil es am besten unsere Vorstellungen aufgegriffen und sogar an einigen Stellen weitergeführt beziehungsweise konkretisiert hat.«

Dominik Abé Besonders am Anfang hatten wir mit einem sehr langsamen Abnahmeprozess zu kämpfen. Wenn ein Teil des Spiels fertiggestellt wurde, musste dieser von beiden Parteien das Okay bekommen. Das hat zeitweise so lange gedauert, dass wir auf Risiko begonnen haben, an diesen Teilen aufbauend weiterzuarbeiten. Schließlich sind wir dazu übergegangen, mit mixtvision direkt über Skype zu kommunizieren.

Aber selbst bei dieser einfachen und direkten Kommunikation ist es wichtig, dass neue Ideen präsentabel vorbereitet und Ergebnisse sowie Entscheidungen festgehalten werden. Wir glauben, dass es schwieriger ist, für eine Idee zu begeistern, die »auf dem Papier« im Konzeptstadium vorgestellt wird. Aus diesem Grund haben wir Ideen oft prototypisch bis zu einem besser vorstellbaren Stadium ausgearbeitet.

Es gab aber ein Beispiel, bei dem das nicht geklappt hat: Wir fanden, dass ein Nordpol gut auf die Weltkugel passen würde, also haben wir diesen ausgearbeitet, inklusive eines Pinguins. mixtvision hat das auch gut gefallen, aber der Lizenzgeber fand, dass Schnee und Eis nicht in die Welt passen. Wir mussten also produzierten Content wieder streichen. Dem Pinguin trauert das Team immer noch hinterher.

Making Games »Oh, wie schön ist Panama« spricht eine sehr junge Zielgruppe an. Wie seid ihr die Entwicklung des Spiels aus dieser Perspektive angegangen?

Benjamin Feld Einer unserer Schwerpunkte liegt im Bereich der Kinder- und Jugendmedienproduktion, sodass wir auf langjährige Erfahrungen in dieser Zielgruppe zurückblicken können. Tatsächlich besteht der große Spagat darin, Themen zu entwickeln, die bei Kindern und Eltern gleichermaßen gut ankommen.

Ein wichtiger Punkt, damit die App auch entsprechend gut bei der äußerst divergenten Zielgruppe ankommt, ist sicherlich das Playtesting. Für jede App laden wir einen wechselnden Pool an Kindern zum Testen zu uns ein. Wir beobachten sehr genau, wie die Kinder mit den Apps interagieren und wo sie Probleme bei der Bedienung haben. Anschließend vertiefen wir unsere persönlichen Einsichten noch im Gespräch mit ihnen, was viel Fingerspitzengefühl bedarf. Statistisch ist das natürlich nicht repräsentativ, aber es hilft uns in der Entwicklung enorm.

Im Falle von »Panama« war es außerdem sehr hilfreich, dass auch die Entwickler von Mimimi direkt bei den Playtests mit vor Ort waren, sodass sie die Eindrücke und Ergebnisse aus erster Hand bekamen.

Dominik Abé Aus Game-Design-Sicht gibt es allein bei einer sehr jungen Zielgruppe schon ein großes Problem: Ein 2-Jähriger hat ganz andere motorische Fähigkeiten als ein 5-Jäh-

riger. Für das reine Abspielen der Geschichte wäre das okay, aber dann gibt es da noch die Minispiele. »Oh, wie schön ist Panama« spricht ja auch Kinder an, die eventuell noch Hilfe bei der Bedienung eines Smartphones oder Tablets brauchen. Deswegen gibt es beispielsweise zusätzlich einen Film-Modus, in dem die Geschichte automatisch und ohne Minispielunterbrechung abgespielt wird. Für uns war das die erste App mit einer so jungen Zielgruppe. Das ist besonders deswegen schwierig, weil dieser Gruppe die Spielerfahrung fehlt und gewisse universal gültige Symbole und Spielmechaniken nicht greifen. Das ist beispielsweise bei Medaillen so, die man als Belohnung für seine Leistung bei den Minispielen bekommt. Die ganz kleinen Kinder haben noch nicht gelernt, dass Silber besser ist als Bronze. Es fliegen Partikel, es leuchtet, es ist bunt, die Musik ist fröhlich – also schlussfolgern die Kleinkinder, dass Bronze genauso gut sein muss wie Silber oder Gold. Generell bewährt es sich bei Kleinkindern – völlig anders als im regulären Game Design –, keine Bestrafungen anzuwenden.

Making Games Die App ist ja sehr Story-zentriert. Wie seid ihr vorgegangen, um passende Minispiele zu finden – oder gab es hier bestimmte Vorgaben?

Benjamin Feld Uns war es extrem wichtig, die Geschichte und ihre Moral in der digitalen Umsetzung zu erhalten. Wir wollten dabei von Anfang an einen Flow in der Kombination zwischen Globus, Kapitel und Spiele kreieren, der die Geschichte nicht unnötig unterbricht. Deshalb sollten trotz aller Interaktivität in erster Linie die Kapitel möglichst filmisch umgesetzt

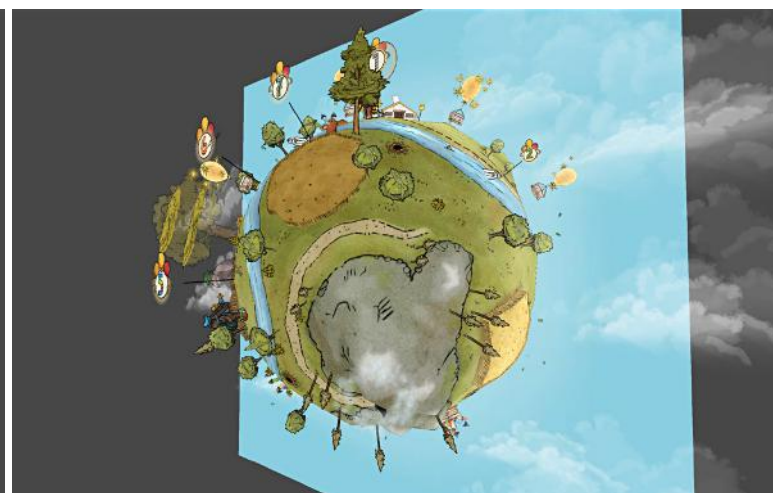
werden, ohne den Charme eines Bilderbuches mit der Möglichkeit des Vorlesens zu gefährden.

Die Spiele waren dabei nicht zuletzt ein Zugeständnis an die heutige Mediennutzung. Allerdings haben wir ganz besonders darauf geachtet, dass die Spiele Teil der Geschichte sind und nicht

einfach beliebige Minigames ohne jeglichen Bezug zum Kern der Marke.

Angelika Achter Durch enge Vorgaben entstehen oft ganz tolle Ergebnisse, weil man sich darauf konzentrieren kann, das Beste innerhalb der bestehenden Einschränkungen zu schaffen. Dass die Struktur des Spiels durch die Geschichte festgelegt war, hat uns die Möglichkeit gegeben, die Minispiele ganz gezielt in die Story einzubinden. Wir haben uns angesehen, welches Spiel eine gute Ergänzung zur Geschichte wäre, und haben mit einfließen lassen, welche Spielmechaniken für die junge Zielgruppe zu bewältigen sind und Spaß machen. Im ersten Kapitel geht der Bär fischen, und da hat es sich beispielsweise angeboten, ein Minispiel zu machen, in dem der Spieler aus dem Wasser auftauchende Fische angeln muss.

»Bei kleinen Kindern greifen universal gültige Spielmechaniken noch nicht.«



Ein Blick in den Unity-Editor zeigt, wie aufwändig es war, die Originalzeichnungen von »Oh, wie schön ist Panama« als interaktive Elemente zu arrangieren.

Making Games Gab es in der Nachbetrachtung Spielinhalte, die bei Release anders aussahen, als sie ursprünglich geplant waren?

Dominik Abé Der Kern des Spiels ist ja die Geschichte, daher blieben die Texte von Anfang an genau so wie im Buch. Daran haben wir bis zur Fertigstellung festgehalten. Wirklich anders umgesetzt als ursprünglich geplant wurden die Minispiele. Die Vorgabe war anfangs auf eine jüngere Zielgruppe gerichtet, und das Durchspielen der Geschichte sollte immer die gleiche Erfahrung bieten. mixtvision hat dann aber viel Potenzial in den Minispielen gesehen und uns die nötigen Ressourcen zur Verfügung gestellt, um die Zielgruppe auszuweiten und einen Wiederspielwert ebenso für ältere Kinder bieten zu können.

Für uns als Spieleentwickler ist das natürlich eine tolle Nachricht – wenngleich zweischneidig, weil wir mit dem Spiel ja zeitig fertig werden mussten, um unser nächstes Projekt pünktlich zu starten. Also haben wir überlegt, wie man Minispiele einbauen kann, die sowohl 3-Jährigen als auch 10-Jährigen Spaß machen könnten. Die zufriedenstellendste und effizienteste Lösung war, zwei verschiedene Versionen der Minispiele einzubauen. Einmal ohne Zeitdruck und Bewertung der Spielleistung in der Geschichte und einmal eine zeitkritische Version mit Highscore, die man in einem separaten Minispiel-Menü auf der Weltkugel auswählen kann. Natürlich war das ein ungeplanter Mehraufwand, es hat sich aber im Sinne der Wiederspielbarkeit sehr gelohnt.

Making Games Wie seid ihr bei der Vermarktung vorgegangen? Eine bekannte Marke kann hier ja Fluch und Segen gleichermaßen sein.

Benjamin Feld Wenn man die Erwartungen nicht enttäuscht, entsprechende Qualität abgeliefert und vor allem den Kern der Marke nicht verfälscht, dann ist eine gute, bekannte Marke eigentlich immer ein Bonus. Das große Thema in der Vermarktung ist heutzutage ja die Sichtbarkeit, und da hilft eine bekannte Marke schon enorm.

Angelika Achter Wie mittlerweile jeder weiß, sind Apps kaum klassisch zu vermarkten. Unse-

rer Erfahrung nach sind hohe App-Qualität und bekannte Markennamen starke Faktoren, um ein Produkt erfolgreich zu machen und Features in Stores zu erlangen. Marken erhöhen auch die subjektive Wertigkeit einer App und ermöglichen eine bessere Preisgestaltung.

Making Games Könnt ihr abschließend zusammenfassen, was die wichtigsten Erkenntnisse bei der Entwicklung von »Oh, wie schön ist Panama« waren?

Benjamin Feld Das Projektmanagement ist für uns ein immer wiederkehrendes Thema, das wir kontinuierlich versuchen zu optimieren. Dabei stehen nicht einmal die Faktoren Zeit und Geld im Vordergrund, sondern insbesondere die Frage, wie man Kommunikation und Prozesse optimieren kann, um mehr Zeit für die eigentliche inhaltliche und kreative Arbeit zu haben.

Eine Herausforderung ist und bleibt natürlich die Vermarktung. Lokale Märkte sind heute aufgrund von Preisverfall, Überangeboten und zunehmender Monopolisierung einfach zu klein, weshalb eine Internationalisierung unabdingbar ist. Diskutieren kann man auch über die Art der Monetarisierung. Wir haben uns angesichts unserer Zielgruppen bewusst für eine Premium-Variante entschieden. Übrigens halten wir das nicht nur im Hinblick auf die Zielgruppe für eine sinnvolle Entscheidung, sondern glauben, dass Free2Play-Modelle der Branche langfristig grundsätzlich schaden. Außerdem passen sie nicht zu unserem Storytelling-Ansatz.

Dominik Abé Das Projekt war ein riesiger Spaß, weil wir allesamt Fans von Bär, Tiger und Tigerente sind. Wir haben viel darüber gelernt, was es bei einer sehr jungen Zielgruppe zu beachten gibt. Wir finden es zudem gut, dass die App sehr clean und fokussiert geworden ist. Das Drehen der Weltkugel mit dem Finger und das Entdecken der Details darauf macht sehr viel Spaß, das Erleben der Geschichte sowieso.

Natürlich haben wir auch bei der Entwicklung dieses Spiels wieder dazugelernt, wie wir Arbeitsabläufe innerhalb unseres Teams optimieren können. Aber prinzipiell würden wir alles wieder genau so machen.

Interview: Sebastian Weber

»Wir glauben, dass Free2Play-Modelle der Branche langfristig schaden.«



THE FUTURE OF FAST AND EASY UI-DESIGN

HTML, CSS and JavaScript are getting more and more popular to create user interfaces. Stoyan Nikolov explains why HTML is a great choice to work with.



Stoyan Nikolov
is Co-Founder
and Software Architect
at Coherent Labs.

Stoyan Nikolov is Co-Founder and Software Architect at Coherent Labs where he works on the Coherent GT and Coherent UI game UI middleware. Previously he was a senior graphics programmer at Masthead Studios where he worked on the MMORPG »Earthrise«.

Game user interface (UI) is an aspect often neglected in development-related literature and is expected by novice developers to just »appear« during the game creation process. Everyone who has started with such attitude quickly finds out that game UI is both very important for the quality of the final product and more challenging than anticipated.

The challenges of creating UI

A »user interface« represents a convenient abstraction over a concept. UI is tightly integrated with the user experience (UX) of the player. The UI is one of the systems in a game that players interact with the most. All good games have great UX that is usually directly correlated to a coherent and user-friendly interface. Looking for an option in endless menus can ruin the experience of an otherwise great title. A good user interface is unobtrusive and »just works« the way players expect it to.

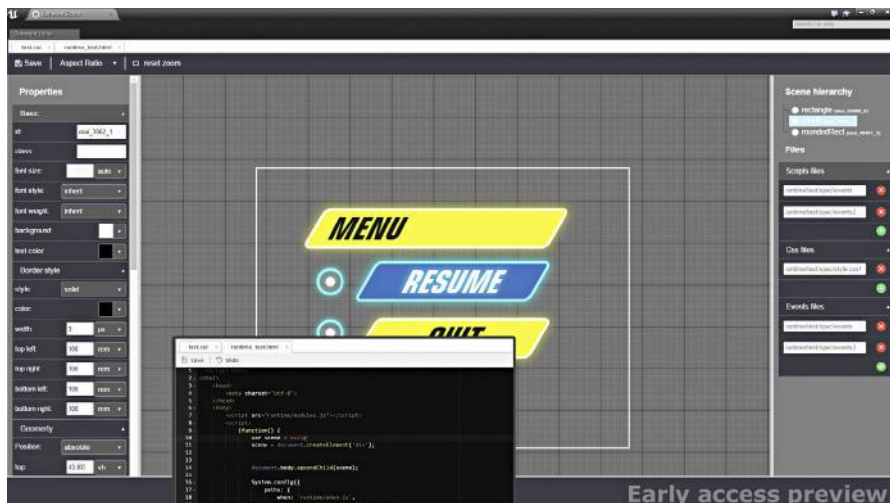
There are games that remove the UI completely and substitute it with in-game elements in order to try to achieve realism. Such games have been few, the reality is that a UI is convenient and helps users. Few people in the real world have a »user interface« aid in their life and this has prompted developers to look for additional realism at the expense of user convenience. Things however have been going the other way

around lately. Having a clever abstraction as a UI is what will become more and more common in our »real« lives. Just think about the advent of augmented reality. Most AR applications actually create a sort of user interface for the world we live in. We will soon move and interact similarly to the way we do in an FPS game.

A game UI is created in phases by multiple people, as most other components in a game. The development involves at least three roles – an artist who draws the 2D images, a designer who arranges them and cares for the user experience and the final layout, and a programmer who connects the UI interactions with the game engine. If we go with this »classic« workflow, we clearly have a bottleneck. The UI creation is not a pipeline, it doesn't flow in one direction. If there is no way for any of the parties involved to change any element in the UI easily, the process quickly becomes very slow and tedious. Imagine a request for moving a button in a menu to the right. If the UI is entirely coded in the engine with C++, this will involve a change from a programmer and a recompile of the game. Same goes for the work of the artist – a good interface system would allow fast re-styles, we wouldn't want to re-draw every single image just to change the theme to a darker hue or to add some effect. UI creation, as any design-oriented activity, requires very fast iterations. An ideal system allows developers to author the UI directly while the game is running and everyone involved should be able to do tweaks even in areas that are not strictly her responsibility.

The evolution of UI

User interfaces follow the general trend in video games to become more complex, dynamic and polished. Very often they blend the classic 2D style with advanced 3D and image effects. The influence by Hollywood movies like »Minority Report« and »Iron Man« is obvious in games like »Watch Dogs« and »Remember Me«. Another trend involves beautiful interfaces that are cleaner and follow the recent trends in the UI of major operating systems like iOS 8, Mac OS 10.10, Windows 8+. Good examples are the menus in »Destiny« and »Dirty Bomb«. All those games have the high dynamicity of the elements in common. The actions of the player cause the interfaces to



A Visual HTML Editor in action that helps designers a lot to quickly iterate on the look and feel of any UI element.



A lot of games today use clean interfaces that follow recent trends of major operating systems. One example for this is »Dirty Bomb« by Splash Damage.

flow from one state to the other instead of just cutting to another screen.

The effects in the classic game heads-up-display (HUD) have also become quite complex. Interface elements in the 3D world are now the norm. Many games choose slightly skewed elements in the HUD that improve the immersion and mimic the visors of combat air pilots. Think of »Crysis 3«, »Battlefield 4«, and in-development games like »Heavy Gear Assault«. The HUD no longer appears splatted on-top of the 3D world but integrates better than ever with it.

New challenges for interface design come with the advent of virtual reality (VR) and augmented reality (AR). No longer is the user looking at the virtual world through a small »window« (the monitor) but is either completely immersed in it or it blends with the real world. The classic HUD can't be just composed on-top of the screen but has to be integrated in the 3D scene. The distinction between the HUD and menus and the interface elements in 3D is effectively removed.

The complexity of modern interfaces comes with an increased cost of development. In my experience, UI development can account for 10 to 20 percent of the whole development time and budget. Genres that traditionally are very UI-heavy like RPGs and MMOs can hit up to 25 per cent. It is no surprise that game companies constantly look for ways to improve the interface development process, reduce cost and improve the quality of the final product.

The era when the whole interface was coded in the engine's language (usually C++) is gone. Virtually all game developers either create their own tools for the UI or go with middleware solutions. The current most popular game engines – Unreal Engine 4 and Unity –

also have their own integrated user interface systems. Unfortunately most stock UI tools require special training, don't implement some standard technology or well-known way of authoring. This makes past experience of developers unusable and closes the UI development in the limited ecosystem of the engine itself.

Misconception about HTML

A technology that fulfills all the requirements for good UI authoring and is gaining popularity among developers is HTML. By HTML I mean the whole package of related technologies – HTML, CSS and JavaScript. They have been created decades ago, have been evolving rapidly and what's more important are entirely geared towards rich user interface experiences. Unfortunately they are also shrouded with a cloud of myths. The first myth is that the content created is mostly static. When people hear about HTML, they often imagine the old websites full of text, a couple of pictures and links. This has changed dramatically with the introduction of HTML5 and CSS3. They provide dynamic, animated environments that feature-wise far exceed custom game UI system, even the ones integrated in large engines like Unreal Engine and Unity. Another popular misconception is that HTML-based rendering technologies are slow. There is some truth in this one if you look at open solutions. The specifications for HTML/CSS/JS are large and very complex and few people try to implement them. Most developers try embedding the popular open source renderers like WebKit, Blink or Gecko. These projects however were not created with games in mind. The performance and feature requirements of games are quite different than the ones in a generic browser. This makes vanilla open source

References and further reading

For further reading, the before mentioned talk from GDC or some tools and software solutions, please visit:

<http://www.makinggames.biz/HTML5UI>

There you will find:

- Yee Cheng Chin on »How to Implement AAA Game UI in HTML and JavaScript« (GDC 2015)
- libRocket user interface middleware for games
- Chromium embedded framework
- EAWebKit resources for NBA Live 14, FIFA 14, Madden NFL 25, Battlefield 4, Plants vs. Zombies: Garden Warfare, and Need for Speed Rivals
- UI middleware Coherent GT
- Blog post »Create animated game HUD with Adobe Edge Animate and Coherent UI« by Hristo Klisurov
- Blog post »What should developers consider when using Chromium Embedded Framework (CEF) in their games?« by Stoyan Nikolov
- Michael Antonov (Oculus) on Asynchronous Timewarp

HTML renderers unsuitable for AAA games, some of them have architectures that are not even compatible with game consoles. To really have a chance to create an amazing high-performance UI with HTML, you need a renderer that is especially designed to work in games.

Why HTML is great for developing UI

There are many benefits that HTML brings in game development. Out-of-the box it solves the most difficult issues encountered when developing an UI system. First and foremost is the layout of the interface. When the resolution changes or an element is added, the rest should re-organize itself in a correct way. CSS defines one of the most powerful layout systems around – just look at professional web pages. If done right, they look great on any resolution or aspect ratio – from a phone to a 27 inch monitor.

Another great benefit is the easy styling. Suppose the UI artist wants to change the hue of all menus in a game – our previous example. This is one of the cases where CSS excels. With a couple of clicks developers can make changes to the overall design. The need for independent styling is universally recognized and many UI system provide some way of doing it. But none is as feature-rich as CSS, it was designed just for that.

If the HTML UI renderer and integration are done right, they improve the iteration time. When the interface is created, the author won't need to re-compile the game to see the changes. Some libraries even allow live editing. The interface can be worked on while the game itself is running. JavaScript is a dynamic scripting language with some functional features, which makes it ideal for modifying logic in »running« environments. Modern games are large software projects that, especially in a development environment, might be slow to start. Even a delay of 20 to 30 seconds required for restarting a game quickly becomes a productivity killer. The HTML family of technologies potentially removes this delay.

When adopting an HTML renderer, developers gain another important feature: access to web services. All games today have some kind of integration with the web. Such integrations might come essentially »for free« if the whole UI uses the same technologies and plays well with the web services. The alternative of writing the integrations by hand is less than appealing. Many companies integrate some kind of in-game browser within their games just to have them.

Everybody knows HTML

HTML has the features and potential but we haven't yet talked about a very important aspect for any development environment – knowledge of the technology by professionals and ease of authoring. It's safe to say that there aren't more popular languages in the IT-industry than HTML, CSS and JavaScript. This means that developers are no longer confined to

the relatively small ecosystems of a game tool but have access to a huge pool of documentation, libraries, peers and professionals. HTML as a game UI allows many talented developers that might not have strict game experience to become UI designers or programmers. Even smaller companies could potentially leverage their web programmers as UI developers. JavaScript comes with a myriad of libraries than can be used to create better UI. Unfortunately this great selection of libraries often confuses developers and the unique requirements – especially performance-wise – mean that not every library has the characteristics to work well in a game.

Authoring HTML and CSS by hand is a preferred way for web gurus but UI is mostly about great design and fast iterations, so good visual tools are required. Fortunately such tools already exist. Many large companies are betting on HTML. Even Adobe the creator of Flash is slowly abandoning it in favor of standard HTML/CSS/JavaScript: »Adobe Edge Animate« is already being used by game companies to create interfaces. Other companies that work towards visual editing tools are Sencha or Google. The major downside of all those tools is that they are designed for the web and require occasional hands-on modifications in order to optimize something or add a game-specific feature. There are already tools in development that will be game-only editors for HTML interfaces.

Current solutions for HTML-UI

If developers want to give a try on HTML as a game interface, there are several alternatives. Theoretically they can create their own HTML renderers. But this is very difficult and time consuming. Major renderers and layout engines (e.g. WebKit, Blink etc.) have hundreds of thousands of lines of code. Arguably not everything in the standard is required for a game and there are libraries that provide HTML rendering but cut many of the more advanced features. One such library is »librocket«. Unfortunately in this case most of the »cut« features are also the most interesting – animations, JavaScript support, etc.

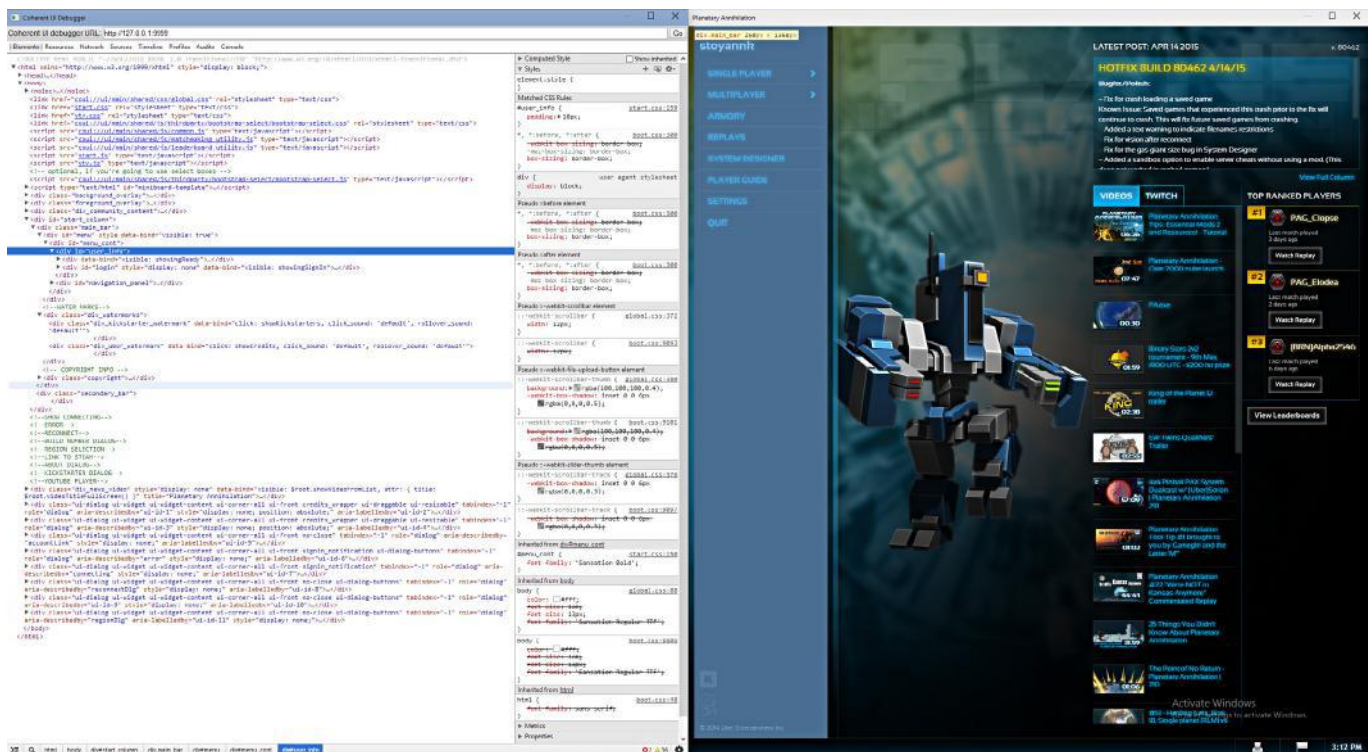
For something that is much more feature-rich and closer to what modern browsers can do, the simplest (and cheapest) is to go with a wrapper around a popular open source HTML renderer. Such a wrapper is the Chromium embedded framework (CEF), which wraps Blink – the rendering engine used in Chrome. CEF is used in many applications and game launchers – Steam is based on it and UE4 recently added an integration to the engine. In both cases however they show a web page within an application or game. Things get much more complicated when a whole UI has to be made. CEF, being based on Blink, has a multi-process architecture. This is great for a generic browser, as it adds security, but might really take its toll on a game performance-wise. The many layers of abstraction and communication really don't cut it when every millisecond counts. All multi-



Some games that are currently in development use HTML for their UI, even more action-oriented ones like »Fractured Space«.



In »Planetary Annihilation« not only the developers profited from the easy to work with HTML code. In addition the players could simply modify the UI.



HTML allows developers to live edit and debug the UI as this example screenshot shows. This makes iteration much easier than solutions where any change has to be done by three different departments.

process solutions suffer also from a problem we could call »GPU starvation«. CEF draws the image in a separate process and then sends it to the game for display. Graphic drivers are not very good at preemption and what often happens is that all the GPU resources get dedicated to the main application (the game) and very few to the Blink renderer. The result is a lagging UI. Even Oculus have encountered this problem, though in a completely different context (see references). Multi-process architectures are also not portable on consoles.

Some HTML renderers are geared towards games and try to solve all of the above issues. One solution is EAWebKit, which was used for the complete UI of »SimCity«. Coherent GT and Coherent UI are commercial solutions that provide HTML rendering for games and are designed with their requirements in mind.

Games need a fast HTML renderer but need also good control over it. Especially on consoles, control over the memory budgets can make a huge difference.

Conclusion

More and more developers are currently embracing HTML as a technology for the UI. There are dozens of titles on the market that shipped with whole interfaces or parts of them made with HTML. This is great proof that it is a viable technology one must be aware of.

»SimCity« shipped with the interface done entirely with HTML and a GDC presentation gives a recollection of the team's experience. »Planetary Annihilation«, another strategy game allowed users to modify the UI, which was made very easy by the use of HTML and JS. For »Guild Wars 2« the developers moved their Trading post to web technologies. More action-oriented games that are still in development

like »Dirty Bomb« and »Fractured Space« have also chosen to go with HTML interfaces.

HTML is deemed to substitute Flash as the most popular technology for UI in both the web world and in games. In game development there are still challenges to face. Designers are still getting used to it and trying to better fit it in their workflow. The current commercial visual editing tools are good but still too much geared towards the web. New game-oriented editors will really leap forward the adoption. They will free developers from having to eventually tweak the generated content. One is already in development by Coherent Labs.

The runtimes for successfully using HTML as a game user interface are already available and proven. Even console developers can now use them. The next frontier for HTML, after it has conquered the web world, is to lend its power to game developers.

Stoyan Nikolov



Maxis used a complete HTML-based interface for the last »SimCity«, today a lot of big games use the so called EAWebKit.

MODEL VIEW CONTROLLER MIT UNITY

Spaghetti-Code ist fast unmöglich zu entwirren und muss im Zweifelsfall neu geschrieben werden. Nurdogan Erdem erklärt, welche Vorteile ein Architekturmuster wie MVC bietet.



Nurdogan Erdem
ist Senior Client Developer
bei Goodgame Studios.

Nurdogan lebt in Hamburg und arbeitet als Senior Client Developer bei Goodgame Studios. Vorher war er bei InnoGames als Senior Software Developer und bei Bigpoint als Tech Lead tätig.

Jeder hat es einmal getan und einige tun es immer noch: Spaghetti-Code. Source Code, der diese Bezeichnung verdient, ist unübersichtlich, voller Abhängigkeiten und schwer erweiterbar. Einen solchen Code zu entwirren, ist meist unmöglich, es sei denn, man schreibt ihn komplett neu. Hat man sich zu diesem Schritt entschieden, sollte man in Betracht ziehen, ein Architekturmuster wie MVC einzusetzen. MVC ist die Abkürzung von Model View Controller und hat im Wesentlichen zum Ziel, Abhängigkeiten im Code zu reduzieren. Die Vorteile von MVC sind vielfältig. Richtig eingesetzt kann es die Wiederverwendung, Erweiterbarkeit und Testbarkeit des Codes erhöhen. Mit Strangeloc (Strange) haben wir ein MVC Framework, das für Unity und C# verfügbar ist. Strange ist stark von Robotlegs inspiriert, einem MVC Framework für ActionScript. Aufgrund der sprachlichen Unterschiede zwischen C# und ActionScript ist es jedoch kein direkter Port. Robotlegs-kundige Entwickler dürften sich bei Strange aber dennoch wie zu Hause fühlen.

Dependency Injection

Eines der Hauptmerkmale von Strange ist Dependency Injection (DI). Im Grunde ist DI etwas, was wir jeden Tag nutzen. Jedes Mal, wenn wir Objekte im Konstruktor einer Klasse als Parameter mitgeben, erzeugen wir eine DI. In Strange müssen wir das nur noch bedingt tun. Denn die Idee hinter DI ist, dass sich eine externe Instanz – nämlich der Injector – um diese Aufgabe kümmert. Das heißt, jedes Mal, wenn eine Klasse etwas möchte, ist es die Aufgabe des Injectors, es ihr zu geben. Damit jedoch der Injector weiß, was er bringen soll, muss vorher klar definiert werden, was die Klasse braucht. Dies geschieht mithilfe des »injectionBinders« und wird demzufolge Binding genannt. Es gibt noch weitere Binder, wie ich im späteren Verlauf zeigen werde. Allen gemeinsam ist, dass sie in der Regel in einer Klasse definiert werden, die von der Klasse »MVCContext« des Strange Frameworks erbt. Die Abhängigkeit, die wir binden, sollte

möglichst ein Interface implementieren. Dies macht es später einfacher, sofern erforderlich, die Abhängigkeit auszutauschen.

```
injectionBinder.Bind<IMyClass>().  
To<MyClass>();
```

Wir können nun überall dort, wo wir auf den Injector respektive das Strange Framework Zugriff haben, die Abhängigkeit »injecten«. Hierzu wird das [Inject] Tag genutzt.

```
[Inject]  
public IMyClass {get; set;}
```

Wird das Interface von mehreren Klassen implementiert, kann man auch die Methode ToName() nutzen. Dies nennt man »named injection«.

```
injectionBinder.Bind<IEnemy>().To<Dwarf>().  
ToName(EnemyType.ADVANCED);  
injectionBinder.Bind<IEnemy>().To<Rat>().  
ToName(EnemyType.BASIC);
```

Das [Inject] Tag wird daher um den entsprechenden Namen erweitert.

```
[Inject(EnemyType.BASIC)]  
public IEnemy enemy {get;set;}
```

Auch ein polymorphes Binding ist möglich, indem man mehrere Interfaces mit der konkreten Klasse verknüpft.

```
injectionBinder.Bind<IHittable>().  
Bind<IUpdateable>().To<Dwarf>();
```

Überall dort, wo entweder IHittable oder IUpdateable injectet wird, bekommt man nun die Instanz der Klasse Dwarf beziehungsweise das entsprechende Interface. Wie wir sehen, definiert der injectionBinder vorab, was der Injector liefert, wenn er nach einer bestimmten Klasse gefragt wird. Die Klasse, in die injectet wurde, hat daher keine Kenntnis darüber, ob die Abhängigkeit ein Singleton oder eine neue Instanz ist. Bei diesem einfachen Binding wird immer eine neue Instanz der Klasse MyClass injectet. Dies nennt man, analog zum Factory-Designmuster, »Factory Mapping«. Oft hat man aber auch zentrale Klassen, von der nur eine einzige Instanz erwünscht ist. Denkbar wäre etwa ein Objekt, das Konfigurationsdaten vorhält. Hier bietet sich die ToSingleton()-Methode an. Das Singleton Pattern dürfte vielen

als Designmuster bekannt sein. Es injectet eine Instanz, die nur einmal existiert. Auch hier nutzen wir wieder ein Interface.

```
injectionBinder.Bind<IConfiguration>().
To<Configuration>().ToSingleton();
```

Was ist aber, wenn wir ein Objekt injecten wollen, das wir von einer externen Instanz bekommen haben? Beispielsweise ein Objekt, das Konfigurationsdaten vorhält? Hier kommt das »Value Mapping« ins Spiel.

```
Configuration config = loadConfiguration();
injectionBinder.Bind<IConfig>().
ToValue(config);
```

Wenn ich eine Klasse injecte, sorgt der Injector dafür, dass alle Injections innerhalb dieser Klasse zur Verfügung stehen. Wenn ich ein Objekt allerdings mit »new MyClass()« instanziiere, werden keine Abhängigkeiten injectet. Wie auch? Das Strange Framework bekommt von einer Instanziierung mit dem new-Operator nichts mit. Möchte man daher manuell eine Klasse injecten und gewährleisten, dass deren Abhängigkeiten ebenfalls injectet werden, muss man die Methode `GetInstance<Classname>()` nutzen.

```
IMyClass myInstance = injectionBinder.
GetInstance<IMyClass>() as IMyClass;
```

Möchte man anschließend direkt im Konstruktor der Klasse MyClass auf Injections zugreifen, wird man eine Überraschung erleben. Die Injections sind gar nicht verfügbar, weil sie zu diesem Zeitpunkt schlichtweg noch gar nicht injectet worden sind. Um dem abzuweichen, kann man das `[PostConstruct]` Tag nutzen. Jede Methode, die mit diesem Tag versehen ist, wird unmittelbar aufgerufen, sobald alle Injections verfügbar sind. Das vermeintliche Problem mit dem Konstruktor ist daher eigentlich keines. Da alle Abhängigkeiten vom Injector zur Verfügung gestellt werden, wird in der Regel auch

kein parametrisierter Konstruktor benötigt. In dieser Hinsicht muss man also umdenken.

Das MVC-Komponenten-Modell

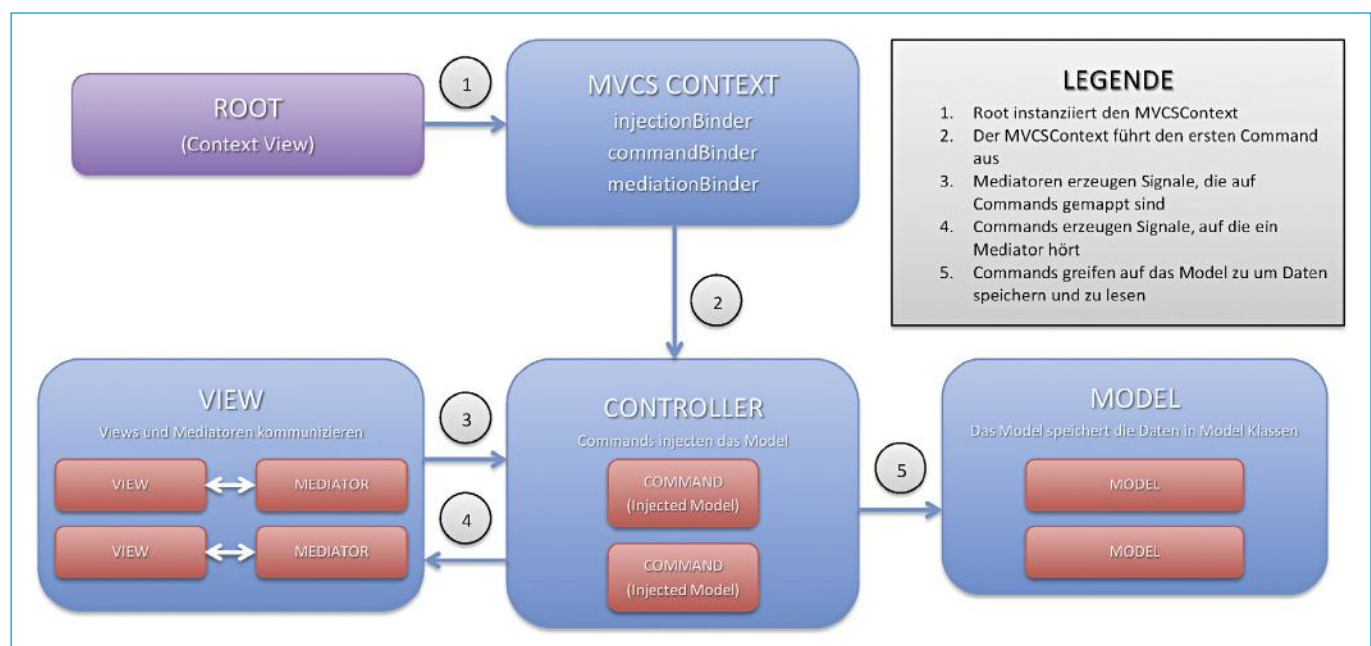
Die drei unterschiedlichen Komponenten von MVC sind das Model, der View und der Controller. Im Model werden die Daten gespeichert, der View kümmert sich um die Darstellung dieser Daten, und im Controller findet die Geschäftslogik statt.

Das Model

Stellen Sie sich beispielsweise ein Rollenspiel wie »Diablo« vor: Es hat eine Unmenge von Items, deren Daten irgendwo untergebracht werden müssen. Ein solches Item kann viele Attribute beinhalten, wie etwa den Schaden, den es anrichtet. Abgesehen von grafischen Daten, die in den View gehören, speichert ein Item-Objekt daher alles, was irgendwie relevant für ein Item ist.

Um die Item-Objekte unterzubringen, erzeugen wir zum Beispiel die Klasse `ItemModel` und speichern diese in einem Dictionary. Ich empfehle, nur der Klasse `ItemModel` direkten Zugriff auf das Dictionary zu geben. Ein Client, der Item-Objekte im `ItemModel` speichern oder löschen möchte, sollte daher mit öffentlichen Methoden wie etwa `AddItem()` oder `DeleteItem()` Vorlieb nehmen. Benötigt ein Client dennoch die komplette Liste der Item-Objekte, kann das `ItemModel` das Dictionary in ein List-Objekt konvertieren und einen Getter zur Verfügung stellen. Kurzum: Die Verwaltung des Dictionarys sollte nur dem `ItemModel` vorbehalten sein. Um keine Abhängigkeiten zum Client zu erzeugen, sollte die Model-Klasse natürlich wieder ein Interface implementieren.

Das Model ist insofern für die anderen Komponenten interessant, als möglichst alle darauf



Eine Veranschaulichung des MVC-Komponenten-Modells, bestehend aus Model, View und Controller.

zugreifen wollen. Daher möchte ich an dieser Stelle einige Tipps aufzählen, die vorrangig darauf abzielen, die Geschäftslogik nicht an den falschen Stellen zu implementieren.

- Schreibzugriffe auf das Model sollten nur vom Controller beziehungsweise einem Command erfolgen. Lesezugriffe sind natürlich ebenso erlaubt.
- Ein Mediator kann das Model injecten, um darauf zuzugreifen. Allerdings sollte sich der Zugriff auf Leserechte beschränken. Zwar erzeugt man hier eine Abhängigkeit zwischen Mediator und Model; weil wir allerdings mit Interfaces arbeiten, ist dieser Umstand nicht schlimm und kann daher einiges an Schreibarbeit ersparen.
- Ein Model sollte keine Nachrichten in das Strange Framework schicken. Stattdessen sollte ein Command diese Aufgabe übernehmen, nachdem er Daten im Model geändert hat.
- Ein Model sollte selbst auf keine Nachrichten hören.



Ein Rollenspiel wie »Diablo 3« hat eine Unmenge von Items, deren Daten irgendwo untergebracht werden müssen.

Der Controller

Im Controller findet die Geschäftslogik statt, die im Wesentlichen auf dem Command Pattern aufbaut. Im Grunde nutzt man hier eine eigene Klasse, die von der Command-Klasse des Frameworks erbt. Diese Klasse überschreibt die »Execute()«-Methode seiner Superklasse, die bei der Instanziierung aufgerufen wird. In diese Methode kommt dann die eigentliche Geschäftslogik. Ein Command sollte im vollen Umfang Zugriff auf das Model haben, also Lese- und Schreibzugriff. Die Instanziierung des Commands kann manuell, aber auch indirekt von einer externen Instanz erfolgen. In letzterem Fall wird das Nachrichtensystem in Strange genutzt. Der Sender einer Nachricht kann unterschiedlichen Ursprungs sein. Dies kann eine Benutzerinteraktion oder ein Service sein, der bestimmte Daten fertig geladen hat. Manchmal kommt es aber auch vor, dass der Controller dem View etwas mitteilen möchte. Beispielsweise wenn sich Daten im Model geändert haben und der

View aktualisiert werden muss. In diesem Fall schickt der Command eine Nachricht in das Framework, auf das ein Mediator hören kann. Commands sind in der Regel kurzlebige Objekte. Wurde ein Command instanziiert und dessen Execute-Methode abgearbeitet, wird das Command-Objekt zerstört. Man kann jedoch die asynchronen Methoden Retain() und Release() nutzen, falls man dies selbst steuern möchte. Das kann hilfreich sein, falls der Command auf die asynchrone Antwort eines Services wartet.

Der View

Der View ist für die grafische Präsentation unseres Spiels zuständig und besteht normalerweise aus einer View-Komponente, die von MonoBehavior erbt. Ist es ein komplexer View, wie etwa eine grafische Benutzeroberfläche, kommt zur View-Komponente auch ein Mediator hinzu. Warum ein Mediator? Eine View-Komponente sollte nicht direkt mit dem Strange Framework kommunizieren. Diese Aufgabe übernimmt der Mediator, der als Schnittstelle zur View-Komponente dient. Eine der Aufgaben des Mediators ist es daher, Nachrichten aus dem Framework zu empfangen, zu verarbeiten und anschließend seine View-Komponente entsprechend anzupassen. Möchte wiederum der View dem Mediator etwas mitteilen, beispielsweise eine Benutzeraktion, geschieht dies erneut über Nachrichten. Der View schickt eine Nachricht, sein Mediator empfängt diese und entscheidet hernach, was er damit anstellt. In der Regel erzeugt der Mediator nun eine weitere Nachricht, die an einen Command gebunden ist. Jede View-Komponente hat ihren eigenen Mediator. Ein Mediator wird immer nur dann erzeugt, wenn sein View instanziiert wird. Daher muss das Binding des Mediators zu seiner View-Komponente möglichst früh im MVCContext erfolgen.

```
mediationBinder.Bind<DashboardView>().
To<DashboardMediator>();
```

Kleiner Tipp: Man sollte versuchen, den Mediator so sauber wie möglich zu halten, sprich möglichst wenig Logik im Mediator zu implementieren – was in der Regel immer anfällt, wenn man etwas im View ändert. Zu diesem Zweck können wir wieder Commands nutzen. Diesmal erzeugen wir aber kein Binding zwischen einem Command und einer Nachricht. Stattdessen injecten wir den Command mit injector.GetInstance<Commandname>() und referenzieren dabei die View-Komponente. Anschließend rufen wir die Execute-Methode im Command auf. Alle Änderungen, die an der View-Komponente notwendig sind, werden nun in der Execute-Methode des Commands durchgeführt. Zur besseren Übersicht kann man Commands, die View-Komponenten referenzieren, auch Behavior nennen.

Das wären also die drei Komponenten: Model, View und Controller. Allein für sich stehend richten diese Komponenten natürlich noch nichts aus. Wie kommunizieren diese Module also miteinander?

Das Nachrichtensystem

Um im Strange Framework Nachrichten zwischen den einzelnen Komponenten auszutauschen, können entweder Events oder Signals genutzt werden. Beide Varianten bieten die Möglichkeit des Transports von Objekten zum Empfänger. Wobei von den Entwicklern des Frameworks empfohlen wird, lieber Signals statt Events zu nutzen. Einfach aus dem Grund, weil Events deutlich unflexibler sind, da sie im Vergleich nur ein einziges Datenobjekt transportieren können und nicht typensicher sind. Zudem werden Events immer neu instanziiert und belasten daher den Garbage Collector. Eine Kombination von Events und Signals ist übrigens nicht vorgesehen. Im MVCContext der Applikation muss eindeutig festgelegt werden, welche Variante man nutzen möchte. Ein einfaches Signal mit einem Parameter sieht folgendermaßen aus.

```
public class PositionUpdatedSignal :  
Signal<string> { }
```

Das Signal muss im MVCContext gemappt werden, bevor es injectet werden kann.

```
injectionBinder.Bind<PositionUpdatedSignal>().  
ToSingleton();
```

Anschließend können wir das Signal injecten und dispatchen.

```
[Inject]
```

```
public PositionUpdatedSignal signal {get; set;}  
signal.Dispatch(value);
```

Dies ist ein einfaches Signal, das beispielsweise vom Controller oder von einer View-Komponente erzeugt werden kann. Die Instanz, die auf das Signal hören möchte (z.B. ein Mediator), muss das Signal ebenfalls injecten und eine Methode referenzieren.

```
PositionUpdatedSignal.
```

```
AddListener(OnPositionUpdated);
```

Ein Signal kann auch mit einem Command verknüpft werden. Hierzu nutzt man den »commandBinder«.

```
commandBinder.Bind<KeyPressedSignal>().  
To<HandleKeyboardCommand>();
```

Der Erzeuger dieses Signals wäre beispielsweise ein Mediator, der den Controller ansteuern möchte. Einen dazugehörigen Command zeige ich im nächsten Abschnitt. Beachten Sie, dass Parameter (wie etwa der keyCode), mit signal.Dispatch(keyCode) übergeben werden und im Command injectet werden müssen.

```
public class HandleKeyboardCommand: Com-  
mand{
```

```
    [Inject]
```

```
    public KeyCode keyCode{get; set;}
```

```
    public override void Execute (){  
        //Programmlogik  
    }  
}
```

Nehmen wir nun ein detailliertes Fallbeispiel für den Einsatz von Signals. Angenommen, der Benutzer klickt auf ein Icon auf dem Bildschirm, das im View einen Pfeil auf den Gegner werfen soll. Der Ablauf gestaltet sich im Code wie folgt:

- Der View reagiert auf eine Benutzerinteraktion und sendet ein Signal, auf das der Mediator hört.
- Der Mediator hört auf das Signal vom View und erzeugt seinerseits ein Signal, das an einen Command gebunden ist.
- Der Command prüft im Model, ob die Voraussetzungen für das Erzeugen eines Pfeils gegeben sind. Falls ja, reduziert der Command im Model die Anzahl der Pfeile.
- Der Command erzeugt anschließend ein Signal mit den Ziel-Koordinaten als Transportobjekt.
- Der Mediator empfängt das Signal und leitet die Koordinaten an den View weiter.
- Der View zeichnet einen Pfeil auf den Bildschirm und bewegt diesen zu den Ziel-Koordinaten.

In diesem Fall hat der Mediator mit dem Controller und dem View kommuniziert. Ein anderes Szenario wäre beispielsweise das Bewegen eines Objektes, das bereits auf dem Bildschirm ist.

- Der Command prüft im Model, ob die Voraussetzungen für das Bewegen des Objektes gegeben sind. Wenn ja, ändert der Command die Koordinaten des Objektes im Model. Dies müssen nicht zwingend echte 3D-Koordinaten sein. Es können auch Koordinaten auf einem Raster sein, die der View dann in echte 3D-Koordinaten umrechnet.
- Der Command schickt nun ein Signal an den Mediator mit der Objekt-Id als Parameter.
- Der Mediator, der das Model injectet, empfängt das Signal und greift anhand der Id auf das Datenobjekt im Model zu. Die ermittelten Koordinaten werden in echte 3D-Koordinaten umgerechnet und anschließend an die View-Komponente weitergegeben.

Performanz

Frameworks, die Dependency Injection nutzen, haben meist den Ruf, langsam zu sein. Schuld daran ist Reflection – in diesem Fall System.Reflection in C#. Performanztests, die diese These untermauern, findet man auf einschlägigen Seiten im Netz. Das Problem: Viele dieser Tests sind Labortests, die in der Praxis nahezu keine Anwendung finden. Natürlich ist es nicht gerade performant, in einer Schleife 10.000 Abhängigkeiten zu injecten. Meist ist daher die Implementation das Problem. StrangeIoC bietet zudem einige Mechanismen, um Reflection zu beschleunigen, beispielsweise indem es jede Reflection in einen Cache schreibt. Man sollte daher abwägen, ob man zugunsten einer sauberen, erweiterbaren Architektur einen minimalen Geschwindigkeitsverlust hinnimmt.

Woher bekomme ich es?

StrangeIoC ist kostenlos und kann im Unity Asset Store heruntergeladen werden. In dem Paket sind diverse Beispiel-Projekte enthalten, unter anderem auch eines, das den Einsatz von Signals demonstriert.

Nurdogan Erdem



Das noch unfertige Indie-Spiel »Ironshade« nutzt Strangeloc/MVC für eine flexiblere Code-Architektur.

RAPID GAME DEVELOPMENT WITH V-PLAY

Creating mobile 2D games is all about speed. Time of development, content creation, balancing and change of gameplay: all of it should be as quick as possible. For a successful free-to-play game, you also need engaged players and the right monetization strategy. This article shows how the V-Play Game Engine helps to solve these issues and how it was created.



Christian Feldbacher
is CEO & Co-Founder
at V-Play.

Christian has more than 15 years of experience in software development with a strong focus on game and mobile app development for all kinds of platforms including iOS, Android, Windows Phone, native Symbian, Java ME, MeeGo, and BlackBerry. His passion for games and experience in Qt, game, and mobile development led to the creation of V-Play Game Engine. He is speaker of international conferences like GDC or Qt Developer Days, loves game jams and teaches university students how to develop games.

As we saw in the first article, a major benefit of V-Play is the customizable in-game level editor, which works with any kind of 2D game. In this second part, you'll find out just how the in-game level creation components work internally, as well as which technology was used for the V-Play Game Engine and why.

The Beginning of V-Play

V-Play is a cross-platform game engine specialized for the rapid development of 2D games. It was designed initially with the highest priority for games targeting mobile platforms. Since its launch in 2012, little has changed in terms of the requirements of mobile 2D games:

- Fast development cycles: quick change of gameplay, balancing settings and fast testing on PC and mobile devices.
- Fast content creation.
- Support for scripting language while supporting native C++ code in edge cases.
- High performance.
- Simple (animated) UI creation.
- Target multiple screen sizes easily.
- Social capabilities to increase visibility and lower user acquisition costs.
- Plugins to monetize Free2Play games.

In 2011, after testing available engines and even writing three engines in the past, the team behind V-Play felt that there was still a missing link to fulfill all these requirements and evaluation of technologies for the engine began. At that time, Qt was chosen as a clear winner for the foundation of V-Play.

Why is Qt so Cute?

Qt (pronounced »cute«) is well known as a C++ framework that simplifies cross-platform development because it abstracts platform differences like timers, threads, storage, networking, rendering, and multimedia. Traditionally, Qt is most

popular for native desktop applications. Some examples of applications that are built with Qt include VLC Media Player, Skype and Autodesk Maya. All of them make use of the GUI module, which allows the creation of native-looking applications with a single source code base. From the viewpoint of a game developer however, the biggest advantage of Qt is its scripting language on top of the C++ framework, called QML.

QML: A Powerful Language

Qt uses QML (Qt Modeling Language) for scripting support and for defining user interfaces. QML is a declarative language that can be mixed with JavaScript. It has two key advantages over HTML5-based technology: readability & performance.

Listing 1 shows QML code that does the following in just 30 lines of code:

- Adds a button element that changes its color from gray to blue when pressed.
- The button is draggable.
- The button tracks the number of presses and displays it in its Text element.
- The button becomes invisible after it has been pressed 10 times.

The QML components like Rectangle or MouseArea are internally written in QML or C++. Although QML was originally intended as a UI definition language, components were added which also allow the writing of application logic in QML files, or in separate JavaScript files for better code separation.

Compared to imperative languages such as C++ or JavaScript, QML results in far fewer lines of codes, while still having native performance because the scripted code is translated to C++ code. One of the reasons for code-savings is the powerful concept of signals and properties. Each property of a component yields a changed-signal once it changes its value. This also allows properties to be connected with each other, meaning that connected properties get re-

The series at a glance

- **Part 1**
User-generated content in the V-Play Game Engine
- **Part 2**
Rapid Game Development with V-Play

evaluated at every change (also called property bindings.) In [Listing 1](#), this concept is used for the visible property of Rectangle, which is connected to the pressCount property. Only if pressCount is smaller than 10, the visible property is true, otherwise it is set to false. In imperative languages, an if-check in the setter source code would be needed or a polling approach checking the pressCount status, leading to dependencies in independent code parts. With these QML features, your code is not only easier to read but the different components remain independent.

Coming back to game development, this property-binding feature also enables the balancing changes to be done at runtime in a clean way. It is probably THE most time- and code-saving feature of QML and extremely useful for game development.

To measure how big the difference in terms of code saving is, we developed an app with Cocos2d-x in C++ and then made the same app with V-Play in QML. For the same game, more than 60 per cent of source code could be saved in comparison to C++. With V-Play we brought 369 lines of code down to just 150. You can find the full comparison at www.v-play.net/c2dx.

Keep in mind this is only a simple game – for projects that are more complex the difference gets even bigger. What's more, far fewer characters are needed per line of code in QML. Compared to other frameworks with scripting like Corona, an average of 20 to 30 per cent of code could be saved (you can also find the source code comparison of the same projects on the previous link).

A Note About Performance

So, not only can you use QML for UI design but also to add logic in JavaScript. The property bindings and QML elements are all implemented in C++. Also, the renderer is based on a scene graph that optimizes automatically for a minimum amount of draw calls and texture swaps, which yields a much better performance compared to other scripted languages.

However, for existing source code you may have written in C++ or for existing C++, Objective-C or Android code or frameworks, you can easily connect QML with a C++ layer with a clean scripting interface. The same interface that Qt uses internally. One great thing about Qt is that it's open-source: this means you can take a look at the source code of all the components and learn from some of their best practices.

Qt Creator IDE

Qt comes with its own IDE, called Qt Creator. Qt Creator is written in Qt itself and available for the Windows, Mac OS X and Linux desktop platforms. It has many powerful features, including:

- **Code Completion:** The code editor supports the editing of C++, QML and JavaScript code with code completion of functions and classes, refactoring and auto-completion of filenames in the project directory.

```
1 import QtQuick 2.0 // for accessing all QML elements like Rectangle, Text or MouseArea
2 import VPlay 2.0 // for accessing all V-Play elements like GameWindow or Scene
3
4 GameWindow {
5     Scene {
6         Rectangle {
7             id: button
8             color: "lightgrey"
9             x: 100
10            y: 50
11            width: 100
12            height: 40
13            // after the button being pressed more than 10 times, it gets invisible
14            visible: pressCount < 10
15
16            property int pressCount: 0 // initialize with 0, increase in onPressed
17
18            Text {
19                // this will automatically update when pressCount changes
20                text: "Pressed " + button.pressCount + " times"
21                color: "black"
22                anchors.centerIn: parent
23            }
24
25            MouseArea {
26                anchors.fill: parent // make the touchable area as big as the button
27                drag.target: button // this forwards the dragging to the button
28
29                onPressed: {
30                    // make the button blue when pressed
31                    button.color = "blue"
32
33                    // increase pressCount property by 1 each time the button is pressed
34                    button.pressCount++
35                }
36
37                // reset to initial color when the button is released
38                onReleased: button.color = "lightgrey"
39            }
40        } // MouseArea
41    } // Rectangle
42 } // Scene
43 } // GameWindow
44
```

Listing 1: Full source code of a QML example with a draggable button that becomes invisible if pressed 10 times.

- **Debugging and Profiling:** Qt Creator includes a C++ and QML/JS debugger and profiler. In debug mode, variables can be changed on the fly. The debugger even allows debugging on mobile devices.
- **Live Issues:** Spelling errors in function names or paths to files are marked before running the program.
- **Quick Help:** All V-Play and QML components and all C++ classes are accessible within the SDK by pressing the F1 key. Also an index search in all properties, methods and classes is available.
- **Integrated UI Designer:** The UI designer allows you to rapidly design screens without the need to code. It also allows the creation of multiple states of a QML element visually.
- **Version Control:** Qt Creator includes built-in support for versioning tools like Git or SVN from within the IDE.

In addition to these features, the Qt SDK also contains a dedicated tool for internationalization: Qt Linguist simplifies translating texts into multiple languages. It works by scanning all the texts in your project and shows the translation side-by-side within the context of where the text is used. You can also add notes to the translated texts, making life a lot easier for translators. You can then read the system locale where your application is running and have the option to let the user change the language at runtime.

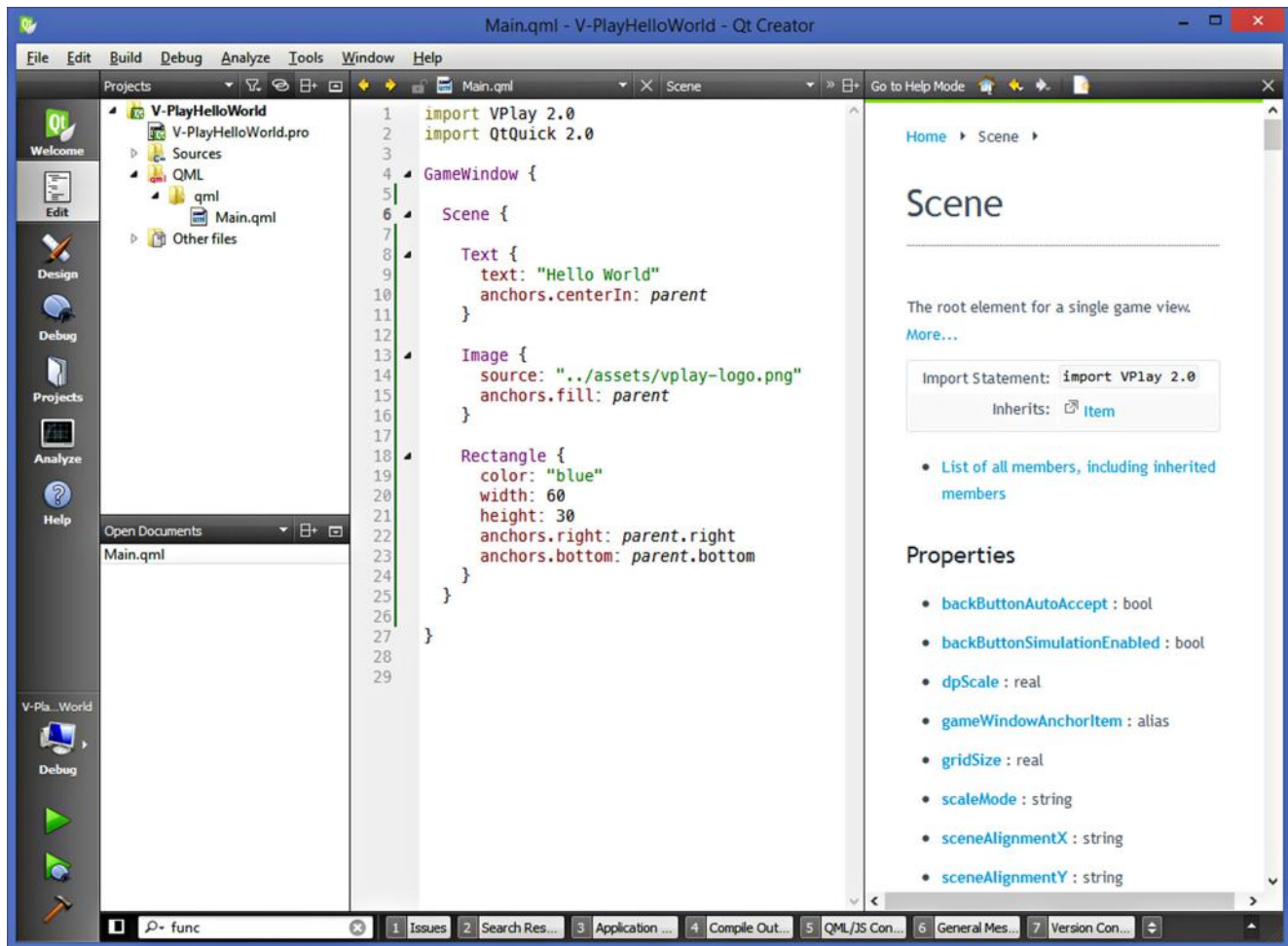
Qt and V-Play

To summarize: Qt is a proven, well-established application framework that was barely used for games before. This was surprising, not only because the scripting language QML was perfectly suited for component-based game development.

About Qt

Qt is a cross-platform application framework based on C++ and the foundation of V-Play. Its development started 1991 and it is used by popular apps like VLC Media Player, Skype, Autodesk Maya or the Panasonic in-flight entertainment system. It is used by more than 800,000 developers and available in open-source and commercial licenses.





Qt Creator IDE with integrated help with the default V-Play project template.

The power of QML, plus its native performance and cross-platform capabilities, was the reason why we chose Qt as the underlying core of the V-Play Game Engine. After making the decision on technology, we added gaming-specific components on top of Qt: sprites, particles, Box2D physics, level editor, audio, video, AI, multi-touch, gestures, and sensor support.

The goal was a clean API design of the gaming components whilst keeping 2D games in mind. To achieve that, we created more than 13 games of very different genres that are most often used in 2D games: platformers, side scrollers, falldown games, strategy games like Tower Defense, puzzle games and action games. All of these games come with full source code with the engine download. This helps you on the way to starting your own project based on one of these templates and makes it even easier to learn how to develop games with V-Play. Five of the demo games are even available in the app stores, allowing you to check out published game examples that also have monetization and analytics features built into them.

V-Play Sample Launcher

You can explore all the examples and demo games quickly using the desktop application, V-Play Sample Launcher. It allows you to run

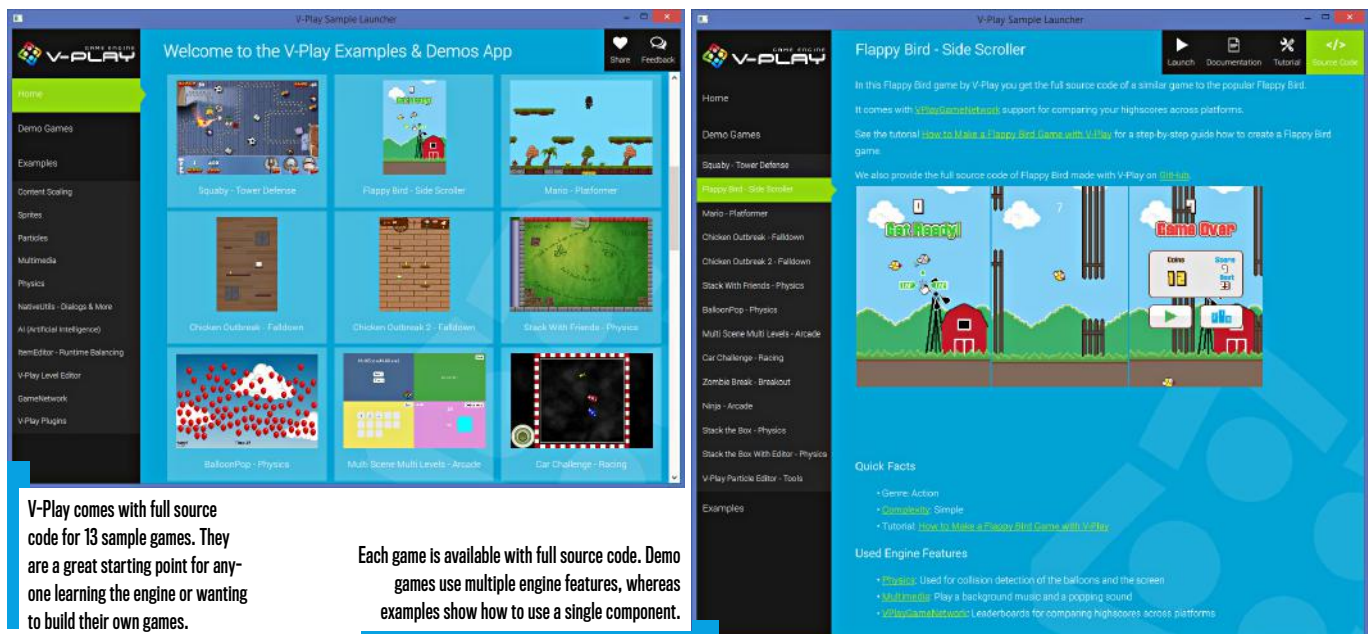
full demo games within a single application and then jump directly to its source code and the accompanying tutorials.

To make getting to grips with the engine as easy as possible, we and our developer community have written many tutorials for all different levels: from a ten-minute tutorial on how to make your own Flappy Bird or balloon popping game, to a 60-minute tutorial on how to create a Tower Defense game with user-generated content.

UI and Multi-Resolution Support

As QML was initially designed to be a language for rapid UI design and to quickly make animated UIs, this is one of the big benefits also available in V-Play. You can use components to anchor items relative to its parent objects and use `ListView`s, `TableView`s, `Rows`, `Columns` or `Grid` elements to position elements on the screen. This makes UI design much more efficient when compared to 3D-based engines forcing a 2D UI system on top of the 3D renderer.

As a game example putting this into practice, in [Listing 2](#) you can see how a `Text` is anchored on the right side of the screen. If you wanted to add an image to the left of the `Text`, you just need to add a `Row` element as their parent and all children will then be aligned next to



each other. The listing also shows the content scaling solution in V-Play. With content scaling, you can define your game for a single logical resolution and it's then scaled to full screen size. It automatically uses the textures that work best for the screen density, so your assets will look crisp on any screen and use the minimum amount of memory to ensure it works well with low-end devices. You'll also get the added benefit that your game will be equally fair across devices and platforms, something which becomes all the more important if you decide to add some kind of leaderboard or multiplayer functionality.

Another trick of content scaling is to make the background image bigger than the logical default size of 3:2 ratio. You then can use the same source code for all screen sizes and aspect ratios. The background image of the Tower Defense game Squaby (page 66) shows the highlighted logical »safe zone« in the center, which is visible on all devices. The surrounding background parts are only shown on devices with other aspect ratios. To test how your game looks on multiple devices, resolutions and aspect ratios, you can switch the resolution at runtime simply by pressing Ctrl + the number keys 1 to 9 while testing on your PC and you'll see the result immediately.

We also have extended Qt to support density-independence, which is useful if you want to make a responsive layout for phone, tablet and desktop. This makes loading a different layout easy with conditions that you set. For example, you can load a different layout if the screen width is bigger than five inches, while still guaranteeing that a button is 0.8 inches tall across screens. These capabilities also enable you to create cross-platform business apps or educational apps of all kinds, not only games.

Native Plugins

The key to success in Free2Play games is to analyze your players, increase engagement, and to

monetize your engaged players. For all of these, the V-Play plugins are at your disposal. They make it possible to use native iOS or Android functionality and industry-leading 3rd party services from the QML scripting environment with very few lines of code.

```
Scene {
    property int numBoxes: 0 // counter of existing boxes in the "game"

    PhysicsWorld { gravity.y: -9.81 } // make the box fall down

    Text {
        text: "Boxes: " + numBoxes // gets updated every time numBoxes changes
        anchors.right: parent.right
        anchors.rightMargin: 10 // 10 dp margin from the right side of screen
    }

    Timer {
        interval: 1000 // spawn a new box every second
        onTriggered: { // create a new box somewhere along the scene width on the top
            entityManager.createEntityFromComponentWithProperties(boxComponent,
                {x: utils.generateRandomValueBetween(0, parent.width)})
        }
    }

    Component {
        id: boxComponent
        EntityBase {
            entityType: "box"

            Component.onCompleted: numBoxes++ // constructor call
            Image { id: boxImage
                source: "../assets/box.png"
                anchors.fill: boxCollider
            }
            BoxCollider {
                anchors.fill: boxImage // make as big as the image
                fixture.onBeginContact: {
                    parent.removeEntity() // remove box at collision with floor
                    numBoxes-- // decrease the boxes counter
                }
            }
        }
    } // Box EntityBase
    } // Box Component

    EntityBase {
        // floor entity, make as wide as the screen and position it on bottom
        anchors {
            top: parent.bottom // top of the floor is the bottom of the scene
            left: parent.left
            right: parent.right
        }
        height: 30 // 30 logical pixels of a 480x320 logical scene
        BoxCollider { bodyType: Body.Static } // do not move this body from gravity
    }
} // Scene
```

Listing 2: Source code of a sample where boxes are spawned randomly across the top of the screen and then falling down with the Box2D physics system. The Text element displays the number of available boxes on the screen and is displayed on the top right of the screen.



To test how your game will look like on different device resolutions and aspect ratios, you can just press the keys Ctrl + 1-9 and see the result immediately.

Unlike the Unity Asset Store, all plugins are included in an Indie license, as we believe using them should be an essential part of every game. These are the plugins currently available:

- In-app purchase plugin to allow the adding of virtual currencies to your game across platforms in one go. You can even use it on desktop to test the purchase process or to test your in-game shop.
- Chartboost and AdMob plugins to monetize your game with different types of ads.
- Flurry and Google Analytics plugins to measure and improve your game engagement.
- Push notification plugins for remote and local push notifications to increase your player's engagement.
- HockeyApp plugin for cross-platform beta distribution of your games to a team of testers and automatic crash reporting.
- Social plugins like Facebook to increase game downloads.
- V-Play Game Network to increase player engagement with cross-platform leaderboards, achievements, and more. More on this in the next section.

New services are added every month based on popularity in the dev community. Alternatively, you can also add your own native SDK or third party service and make it available for QML.

Using the plugins in your game is then only a matter of minutes with the QML integration. In [Listing 3](#), you can see an application that shows banner ads unless the noAdPurchase was done. Also, each purchase is sent to the Flurry analytics service. The full implementation of three plugins takes less than 50 lines of code and works across platforms.

V-Play Game Network

V-Play Game Network is a cross-platform social gaming service that allows players to compete with game high scores, achievements, and more. We've built it based on the requirements from many developers to help them increase player retention. It is available exclusively for V-Play powered games and saves you a lot of time on social integration and multiplayer integration. Internally, V-Play Game Network is a cloud-based backend service with a client API for QML. By default you can use our hosted servers on S3 instances, or choose your own servers if you prefer to. With the networking components, you can then add these features to your game easily:

- **No Extra User Account or Registration:** Players do not need to create an extra account to compare their high scores or achievements across platforms, which leads to higher usage. A login with Facebook and Game Center is available, but not a requirement. This is a key benefit over other gaming services forcing a registration: in our games we have seen high drop-out rates up to 60 per cent of players who do not use other services because of pri-

vacy concerns or feeling insecure connecting a game with their Facebook account – with V-Play Game Network you can satisfy these players as well.

- **Full Offline Support:** High scores, achievements and Cloud Storage can be used even when your players have no Internet connection and get synced automatically as soon as connection is available again.
- **Fully customizable UI:** V-Play Game Network comes with a ready-made UI so you can integrate it in your game in ten minutes. However, with the strength of QML you can fully customize all views to your game easily and give it an extra-polished look.
- **Export Player Data:** You can export your player data and use it, for example, to send email newsletters to bring your players back to the game, or to promote your other games with personal emails.
- **Simple QML API:** Use properties and signals to get notified when important things happen, for example when the player connected with Facebook, reached a new high score or unlocked an achievement.
- **Sharing to Game Center:** With a single API call you can report your game progress to the V-Play Game Network and also to Apple Game Center. So your players can use both their familiar game service on iOS and also compare their progress with players on other platforms. Google Play Game Services will follow soon.
- **Easy Facebook Integration:** Let your players connect with Facebook in a single QML function call. We have integrated posting the player's score and achievements to your player's timeline, so the most important Facebook functionality to bring new downloads and make your players return to your games comes already built-in.
- **Cloud Storage:** With this component, storing your data in the cloud is as easy as it can be: you can use it to synchronize your player's data across multiple devices or to change the balancing of your game remotely from our Web Dashboard, without the need to release a new update to the app stores.
- **Cross-Platform Leaderboards:** In most games players can reach high scores representing how well a player has done. With V-Play Game Network you can use unlimited leaderboards to let players compare against each other in a competitive way and brag about it on Facebook, keeping them playing. You can either display scores across all platforms on one leaderboard or filter the leaderboards based on platform, time, or users.
- **Cross-platform Achievements:** Reward your players with achievements for reaching certain goals in your game and give them more incentive to keep playing your game. If connected with Facebook achievements get even synced between different devices of a single player.

More Resources

Use these links to dive deeper into game creation with V-Play

[v-play.net/editor](#) – Contains many resources for user-generated content with V-Play games like a step-by-step tutorial on how to add an in-game editor and a LevelStore to your game; or videos of games powering UGC.

[v-play.net/squaby-game](#) – Tower defense game published in the app stores as »Squaby«. Comes with full source code as part of the V-Play SDK and uses UGC.

[v-play.net/game-network](#) – Examples and documentation how to use V-Play Game Network in your game to add social cross-platform features.

[v-play.net/dl](#) – The V-Play Engine can be downloaded for free here, including 13 sample games, API examples and the V-Play Sample Launcher.

- **Multiplayer Challenges:** These are powerful tools to increase player retention as they motivate players to beat a friend's score, especially if you reward them for won challenges with, for example, in-game currency.
- **Level Sharing:** With V-Play Game Network, you can let your players easily share their user-generated levels. This means constant level updates for your players and saves you the time for additional level packs. Even more, you can charge for the levels made by your players and reward your best level creators with the LevelStore component, fully skinnable to your game. For more details on the user-generated content capabilities of V-Play, be sure to check out the first article of this series.

Integration of these features takes less than ten minutes. If the users grant Facebook permission, the high score and achievements will also be posted on their Facebook timeline when they have a higher score than a friend. As V-Play Game Network works across platforms, your players can compare their scores between iOS, Android, Windows Phone and even on Windows, Linux and Mac for Desktop.

Recap

This article has shown you why Qt was chosen as the core for V-Play: not only could we rely on a proven and well-established cross-platform framework, but also the scripting language QML and JavaScript are a perfect fit for component-based 2D games. The Qt technology choice allowed us to add components that accelerate 2D game development. We have covered the in-game level editor, runtime balancing and user-generated content capabilities in the first article. In this article, we introduced the V-Play Sample Launcher to browse demo games and examples,

```

1 import QtQuick 2.0
2 import VPlay 2.0
3 import VPlayPlugins.soomla 1.1
4 import VPlayPlugins.chartboost 1.1
5 import VPlayPlugins.flurry 1.0
6
7 GameWindow {
8
9     Store {
10         id: store
11         goods: [
12             LifetimeGood {
13                 id: noadsGood
14                 itemId: "noads_id"
15
16                 purchaseType: StorePurchase { id: noAdPurchase; productId: noadsGood.itemId; price: 0.89 }
17             }
18         ]
19         onStorePurchaseStarted: flurry.logEvent("Store.Purchase","Started", itemId)
20         onStorePurchased: flurry.logEvent("Store.Purchase","Purchased", itemId)
21         onStorePurchaseCancelled: flurry.logEvent("Store.Purchase","Cancelled", itemId)
22     }
23
24     Flurry {
25         id: flurry
26         apiKey: "<your-flurry-api-key>" // from Flurry Dashboard
27     }
28
29     Scene {
30         // put the buttons below each other, in a Column
31         Column {
32             spacing: 10 // 10 dp spacing between the 2 buttons
33             SimpleButton {
34                 text: "Purchase no-ads"
35                 onClicked: store.buyItem(store.noadsGood.itemId)
36             }
37             SimpleButton {
38                 text: "Show Interstitial if not purchased"
39                 onClicked: chartboost.showInterstitial()
40             }
41         ]
42
43         Chartboost {
44             id: chartboost
45             // do not display the interstitial if the no-ads good was purchased
46             shouldDisplayInterstitial: !noadsGood.purchased
47         }
48     } // Scene
49 } // GameWindow

```

Listing 3: Full application displaying an interstitial Chartboost ad, in-app purchase integration to remove the ad display and logging the purchase with Flurry Analytics in less than 50 lines of code.

native plugins to monetize and analyze your app, and the V-Play Game Network to increase player engagement with social elements.

In a recent global developer survey by Research2Guidance of more than 2,000 developers participating worldwide, V-Play grabbed first place in the categories easiest to learn, biggest time savings and best support: the core values we originally had in mind when we designed the engine and API. It's also for these reasons that we see many developers coming over from Unity, Corona or Cocos2d: they're now using V-Play for their game as it's better suited to their 2D game's requirements and they find it simpler and faster to use.

Christian Feldbacher

Leaderboards · Achievements · Multiplayer · Skinnable · User-Generated Content · Offline Availability · Cloud Sync

Highscores

All Time

Friends

- 1 Christoph Ackerflusser 12539
- 2 Vratislav Gréaopir 9851

All Players

- 1 Billy Smith 82312
- 2 Laura Lang 80495
- 3 Yash Shiva 80090
- 4 Arata Nikephoros 77950
- 5 Kilian Tomislav

One API to connect with:

```

import VPlay 2.0
import QtQuick 2.0
import VPlayPlugins.facebook 1.0

GameWindow {
    Scene {
        Column {
            SimpleButton {
                text: "Show leaderboard"
                onClicked: gameNetwork.showLeaderboard() // open the leaderboard view
            }
            SimpleButton {
                text: "Increase Highscore"
                property int score: 1
                onClicked: {
                    score++
                    gameNetwork.reportScore(score) // simulate a higher score
                }
            }
        } // Column
        // you can let your players connect with Facebook to see their friends list
        Facebook { id: facebook }

        VPlayGameNetwork {
            id: gameNetwork
            facebookItem: facebook // a score is auto-posted to Facebook when connected

            // increment the achievement counter until the game was opened 5 times
            Component.onCompleted: gameNetwork.incrementAchievement("5opens")

            achievements: [
                Achievement {
                    key: "5opens" // the achievement data is auto-uploaded to the backend
                    name: "Game 5times"
                    iconSource: "../assets/img/achievement_5opens.png"
                    target: 5
                    points: 10
                    description: "Open this game 5 times"
                }
            ]
        } // VPlayGameNetwork
    } // Scene
} // GameWindow

```

True cross-platform with a single API for:

iOS Mac OSX Desktop Windows Phone & WinRT

Achievements

Level Creator First level published	5 points 100% completed
Level Creator Master Publish 10 levels	10 points 10% completed
Level Creator Genius Publish 100 levels	50 points 1% completed
Quality Level Maker One of your levels got rated with more than 4 stars	5 points 100% completed
Novice Level Maker One of your levels got downloaded once	5 points 100% completed

The V-Play Game Network allows cross-platform leaderboards and achievements with 30 lines of code. In the code listing a score is uploaded to the game backend. The achievement gets incremented every game start until it reaches the five target points. If the player is connected with Facebook, the score and the earned achievement is also posted to his timeline from the backend service with a single client API call.

EIN TOOL ZUR UMSETZUNG VON IDEEN

Kickstarter entstand aus der Idee heraus, ein Konzert auf die Beine zu stellen. Inzwischen gibt es über 100 Projekte, die über die Plattform über eine Million Dollar gesammelt haben. CEO Yancey Strickler erklärt im Interview, wieso er kürzere Kampagnen für sinnvoller hält, welche Kampagnen am besten laufen und was er sich vom Kickstarter-Launch in Deutschland erhofft.



Yancey Strickler
ist Mitbegründer und CEO
von Kickstarter.

Yancey ist einer der Gründer von Kickstarter und fungiert derzeit als CEO. Davor war er für Community und Kommunikation zuständig. Bevor er zu Kickstarter kam, arbeitete Yancey als Musikjournalist und schrieb unter anderem für »The Village Voice«, »New York Pitchfork« und weitere Zeitschriften.

Making Games Wie kamt ihr auf die Idee zu Kickstarter?

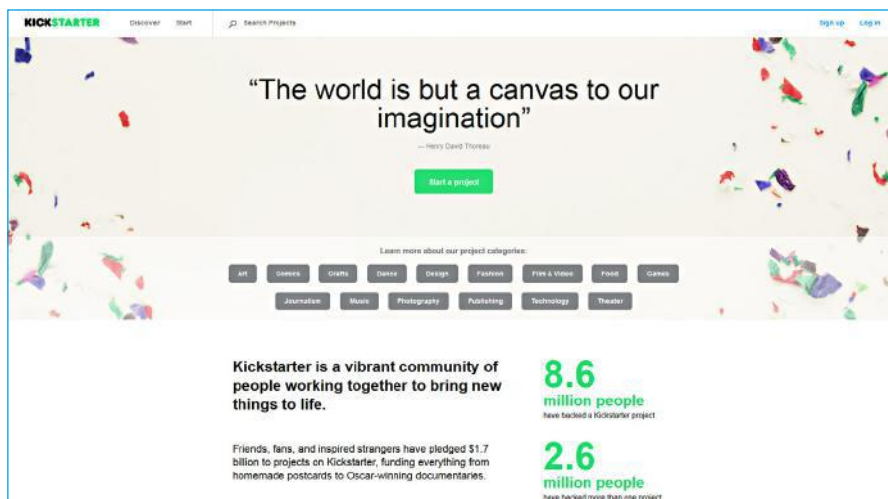
Yancey Strickler Die Website hat drei Mitbegründer. Die ursprüngliche Idee kam von meinem Partner Perry Chen schon 2001, also vor wirklich langer Zeit. Er lebte damals in New Orleans und wollte ein Konzert veranstalten, bei dem die österreichischen DJs Kruder & Dorfmeister auflegen sollten. Allerdings hatte er nicht das Geld, um das alles auf die Beine zu stellen. Also kam ihm die Idee, den Plan für das Konzert zuerst im Internet zu präsentieren. Wenn es am Ende ausverkauft wäre, würden die Leute die Karten auch wirklich erwerben und das Konzert würde stattfinden – und wenn nicht, dann würde eben nichts passieren. Es ging einfach darum, die Idee, die er allein nicht umsetzen konnte, anderen vorzustellen. Dieses Konzept hat ihn die nächsten vier bis fünf Jahre beschäftigt. Wir haben uns dann 2005 kennengelernt. Ich war zu diesem Zeitpunkt Musikjournalist, er war Künstler und Kellner. Ein dritter Freund

kam in weiterer Folge ebenfalls dazu, der Grafikdesigner Charles Adler. Ab 2005 haben wir dann versucht, Kickstarter zu realisieren. Allerdings hatte keiner von uns technisches oder businessbezogenes Hintergrundwissen, wir taten uns also ziemlich schwer. Es dauerte vier Jahre, bis die Seite im April 2009 live ging. Aber die Projekte, die man heute auf Kickstarter sieht, ähneln auf jeden Fall dem ursprünglichen Konzept. Es geht darum, Künstlern und Kreativen zu helfen, Dinge zu erschaffen. Man kann es außerdem als ein Tool sehen, das dabei hilft, Ideen umzusetzen, die ansonsten vermutlich nicht realisiert werden könnten.

Making Games Wie habt ihr die Plattform bekannt gemacht?

Yancey Strickler Ich kann mich gut in die Lage der Kickstarter-Nutzer versetzen. Vieles von dem, was sie für den Erfolg ihrer Kampagnen tun, entspricht dem, was wir damals gemacht haben, damit die Leute Kickstarter nutzen. Wenn jemand eine Kampagne startet, um etwas zu verwirklichen, was ihm am Herzen liegt und worauf er stolz ist, dann erzählt er natürlich als Erstes stolz davon und versucht, so viele Leute wie möglich ins Boot zu holen. Wir haben etwa einen Monat vor dem Launch der Website rund 50 unserer Freunde eingeladen, Projekte zu starten. Jeder von ihnen bekam die Möglichkeit, fünf seiner Freunde ebenfalls einzuladen. Das war im Grunde genommen alles, was wir gemacht haben. Weil das System funktioniert hat und die Leute davon begeistert waren, konnte die Plattform organisch wachsen. Das war von Anfang an der Weg, dem unsere Seite gefolgt ist. Ich hoffe, dass wir das auch in Deutschland sehen werden.

Making Games Kannst du Ratschläge geben, wie man eine erfolgreiche Kampagne auf die Beine stellt? Und gibt es deiner Erfahrung nach



Die Kickstarter-Website ging 2009 live: »Es geht darum, Künstlern und Kreativen zu helfen, Dinge zu erschaffen.«



Double Fines Kampagne zu
»Broken Age« spielte über
3 Millionen Dollar ein:
»Sie vereint so ziemlich alles,
was Kickstarter ausmacht.«

bestimmte Arten von Kampagnen, die besser funktionieren als andere?

Yancey Strickler Ich denke, jeder, der ernsthaft darüber nachgedacht hat, ein Projekt auf Kickstarter zu veröffentlichen, weiß über die offensichtlichsten Dinge Bescheid, wie zum Beispiel ein Video zum Projekt zu machen. Wenn es ein Spiel ist, sollte das Video Gameplay zeigen. Man kann einen Machbarkeitsnachweis einbinden, natürlich. Aber je klarer wird, wie das fertige Spiel funktionieren soll und welche Erfahrungen der Spieler dadurch haben soll, umso größer wird die Reichweite sein. Das heißt also, je weiter man im Projekt fortgeschritten ist, umso größer ist auch das potenzielle Publikum. Das sind grundlegende Punkte.

Was nicht so intuitiv ist und den Erwartungen der Leute widerspricht: Ich befürworte eine vergleichsweise kürzere Kampagnendauer. Bei uns ist es möglich, eine Kampagne bis zu 60 Tage laufen zu lassen, der Durchschnitt liegt bei 30 Tagen. Allerdings folgen alle Projekte einem bestimmten Muster: In den ersten 48 bis 72 Stunden gibt es viel Aktivität, danach wird es wirklich ruhig. Zu Beginn tweeten die Leute darüber oder posten etwas dazu auf Facebook. Aber wenn die Kampagne seit vier Tagen läuft ... das Internet ist eben das Internet und es fühlt sich an, als wären zwei Jahre vergangen. Alle Projekte erreichen also diesen gewissen Punkt niedriger Aktivität, der sich bis zu den letzten paar Tagen der Kampagne erstreckt. Diese Phase ist erfahrungsgemäß emotional sehr herausfordernd für die Verantwortlichen, weil die ersten paar Tage oft sehr aufregend sind: Man steht der Welt optimistisch gegenüber, Essen schmeckt besser und so weiter. Genau das Gegenteil ist allerdings der Fall, wenn es um die Sache dann stiller wird: Man beginnt, an sich selbst zu zweifeln, und wird unruhig. Das ist eine große Herausforderung. Ich denke, dass eine kürzere Kampagne diesen Unruhephasen vorbeugen kann. Wenn man die Dauer der Kampagne ein wenig komprimiert, macht es das Projekt insgesamt vielleicht etwas aufregender, weil es von »Oh, es ist ganz neu!« direkt in »Oh, es ist schon fast vorbei!« übergeht. Ich



warte eigentlich darauf, dass einige Initiatoren dahinterkommen, wie sie diese schnelllebige Mentalität für sich nutzen können. Meiner Meinung nach gibt es hier viel Potenzial.

Making Games Also statt eine Kampagne über 30 Tage zu strecken, sollte sie eher 15 Tage lang sein?

Yancey Strickler Möglicherweise. Ich weiß noch nicht ganz genau, wie es funktionieren würde, daher will ich niemanden dazu überreden, das sofort auszuprobieren. Aber in meiner Fantasie könnte sogar eine 24-Stunden-Kampagne das Internet für einen Tag beherrschen. Klar, wenn du ein großes Spiel machst, das auf iOS, Android und Steam erscheinen soll und du dafür 700.000 Dollar brauchst, ist eine so kurze Kampagne wohl nicht der richtige Weg. Aber wenn du an etwas Kleinerem, Einfacherem arbeitest, dann ist das auf jeden Fall eine Option. Wir werden sehen, ob das irgendwann passieren wird.

Making Games Welche drei Kampagnen – es müssen keine Spiele sein – sollte sich jeder angeschaut haben, der verstehen möchte, wie Kickstarter funktioniert?

Yancey Strickler Die beste Kampagne ist vermutlich die zum Double Fine Adventure »Broken Age«. Sie vereint so ziemlich alles, was Kickstarter ausmacht: einen visionären Künstler, der vom Establishment verkannt wurde (lacht). Tim Schafer und Double Fine machen großartige Spiele, die für Sony etc. aber uninteressant waren, weil sie nicht genug Geld eingebracht haben. Das lässt sich im Prinzip auf alle Projekte übertragen. Sie haben ihr Publikum direkt angesprochen und eingeladen, Teil des Projekts zu werden. Zum einen, was dessen Präsentation

»Die Projekte, die man heute auf Kickstarter sieht, ähneln dem ursprünglichen Konzept.«

»Alle Projekte folgen einem Muster: In den ersten 48 bis 72 Stunden gibt es viel Aktivität, dann wird es wirklich ruhig.«

angeht; aber auch eine Dokumentation über ein Spiel machen zu lassen, während du es entwickelst, ist großartig. Das ist die bestmögliche Art, so etwas anzugehen. Der andere Grund, warum ich »Broken Age« für ein gutes Beispiel halte, ist, dass sie ihre ursprünglichen Pläne weiterentwickelt und aus dem Spiel zwei Teile gemacht haben. Als der erste Teil veröffentlicht wurde, mussten sie dafür ziemlich viel einstecken, was ich etwas lächerlich fand. Die Leute fühlten sich betrogen. In der Realität allerdings sah jeder, der das Projekt unterstützt hatte, dass Double Fine sehr transparent in ihrer Vorgehensweise waren und erklärten, warum sie die Pläne geändert hatten. Und der Erfolg der Kickstarter-Kampagne, der ihre Erwartungen weit übertroffen hatte, führte dazu, dass sie höhere Ansprüche an ihr Projekt stellten. Also haben sie diese auch umgesetzt. Sie haben mit ihren Backern darüber gesprochen und alle haben sie in ihrem Vorhaben unterstützt. Es waren Leute außerhalb dieser Community, die gemeckert haben. Jeder liebt es, im Internet gegen etwas zu stänkern. Dieser ganze Prozess, für den sie meiner Ansicht nach zu Unrecht kritisiert wurden, fasst für mich die Kickstarter-Geschichte ganz gut zusammen, in der es auch einige Kontroversen gab.

Ich bleibe bei Spielen. Ein großes Spiel, das ebenfalls bei Kickstarter seinen Anfang genommen hat, war das Kartenspiel »Cards Against Humanity«. Das war 2010 oder 2011 ein Kickstarter-Projekt. Damals hat es um die 10.000 Dollar gesammelt, war also relativ klein. Daraus wurde ... na ja, das Spiel, von dem du hoffst, dass deine Eltern nie davon erfahren. Aber offensichtlich ist es trotzdem extrem beliebt.

Das Spiel, mit dem ich vermutlich die meiste Zeit verbracht habe und das auch über Kickstarter finanziert wurde, ist »Kentucky Route Zero«. Ein wirklich tolles Indie-Spiel, das viele Preise gewonnen und viel Aufmerksamkeit bekommen hat. Ich erinnere mich, wie ich damals das Video gesehen habe, als die Kampagne live ging. Es war optisch und vom Gamedesign her sehr ansprechend, es lief ein Countrysong ... es war einfach ungewöhnlich und hob sich von der Masse ab. Ich spiele viel, heutzutage meistens auf meinem Smartphone. Aber das fühlte sich einzigartig an und hatte eine eigene Vision, wie das Spiel aussehen sollte, was mich persönlich sehr angesprochen hat.

Making Games Welche Kampagnen laufen insgesamt am besten auf Kickstarter? Spiele? Tech? Kunst? Oder irgendetwas ganz anderes?

Yancey Strickler Die Definition von »Was am besten läuft« ist zunächst mal interessant. Wenn man sich Gaming- oder Tech-Blogs oder auch die normalen Medien ansieht, dann bedeutet es »Was sammelt am meisten Geld ein«. Wer schafft es, mehrere Millionen Dollar zu sammeln? Es gab insgesamt schon über 100 Projekte, die mehr als eine Million bekommen haben. Die meisten dieser Projekte waren entweder Tech oder Spiele. Und es passt auch zusammen, weil es das Publikum anspricht, das den ganzen Tag im Internet unterwegs ist. So funktioniert dieser Teil des Internets, Fan-kultur funktioniert auf diese Weise. Während diese beiden Kategorien das größere Potenzial für Blockbuster-Ergebnisse haben, haben sie zwei der niedrigsten Erfolgsraten in Bezug auf die Zahl der Projekte, die ihr Finanzierungsziel überhaupt erreichen.

Die beiden Kategorien mit den höchsten Erfolgsraten sind Tanz und Theater. Tanz liegt mit weitem Abstand an der Spitze. 70 bis 75 Prozent der Tanzprojekte erreichen ihr Ziel, bei Tech und Spielen sind es 20 Prozent. Das typische Tanzprojekt strebt etwa 3.000 Dollar Finanzierung an und hat rund 80 Unterstützer. Es ist zum Beispiel jemand, der ein Avantgarde-Ballett in einer Berliner Fabrikhalle organisieren will. Kleine, abgeschlossene Projekte wie eine Vorführung oder ein Event, nur ein Moment, den man mit einer Gruppe zusammen schaffen möchte. Diese Projekte sind am ehesten erfolgreich. Sie bekommen aber natürlich auch nicht viel Aufmerksamkeit, weil ein Großteil des Publikums nichts damit anfangen kann.

Wenn ich also darüber nachdenke, was am besten läuft, sehe ich das immer im Verhältnis dazu, was jemand möchte. Es ist wichtig, dass nicht jeder Millionen-Dollar-Erfolge haben möchte. Ich wäre ziemlich unglücklich, wenn das Kickstarter ausmachen würde oder Kickstarter auf dem Weg dorthin wäre. Das wäre falsch. Uns ist wichtig, dass die Leute ihre Wünsche verwirklichen können, statt die Erwartungen anderer zu erfüllen, was in allen Lebensbereichen passiert. Viele der kleinen Projekte sind die erfolgreichsten. Ich selbst unterstütze auch etliche dieser Projekte. Ich habe inzwischen glaube ich um die 1.400 Projekte unterstützt, vor allem Filme und Veröffentlichungen. Ich liebe es, Zeitschriften durch Kickstarter zu bekommen. Ich habe sie zu Hause stapelweise rumliegen. Jeder, der über irgendetwas eine Zeitschrift machen will, wird vermutlich von mir unterstützt. Ich habe in der High School selbst Zeitschriften gemacht. Ich mag diese extrem speziellen Sichtweisen. 20 Seiten über ein Thema, über das ich niemals einen Gedanken verschwenden würde, aber jemand anderes hat sich sehr detailliert damit beschäftigt. Das schätze ich sehr, das gefällt mir.



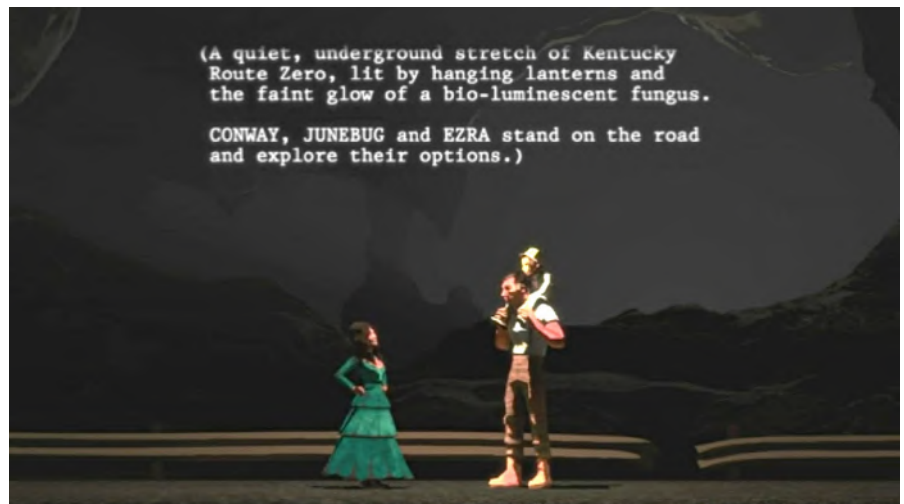
Das Kartenspiel »Cards Against Humanity« nahm ebenfalls auf Kickstarter seinen Anfang: »Es hat um die 10.000 Dollar gesammelt, war also relativ klein. Daraus wurde das Spiel, von dem du hoffst, dass deine Eltern nie davon erfahren.«

Ich denke, was man daraus lernen kann, ist: Ja, es ist das Potenzial da, mit einem Projekt international Aufmerksamkeit zu erregen. Aber tatsächlich funktioniert es am besten, wenn man leidenschaftlich hinter seiner Idee steht. Dafür haben wir Kickstarter gemacht.

Making Games Du hast das Double Fine Adventure vorhin als Beispiel genannt. Nach dieser Kampagne haben Spiele auf Kickstarter für rund zwei Jahre einen großen Boom erlebt. Seitdem ist jedoch die Zahl der erfolgreich finanzierten Projekte gesunken, wie auch die Gesamtsumme. Woran liegt das deiner Meinung nach?

Yancey Strickler Die Gesamtsumme ist 2014 gesunken, aber die Zahl der erfolgreichen Kampagnen ist gestiegen. Nach dem Double Fine Adventure gab es innerhalb weniger Wochen fünf oder sechs Spiele, die mehrere Millionen eingesammelt haben. Es war dieser Moment, in dem viele überschwänglich und enthusiastisch waren. Es war eine neue Weise, wie Spiele gemacht werden. Die Zahl dieser Blockbuster-Spiele hat abgenommen. Ich weiß nicht, wie viele Spiele im vergangenen Jahr mehr als eine Million bekommen haben. Es sind immer noch einige, die meisten Kampagnen im Vergleich zu anderen Kategorien. Doch generell reicht es nicht mehr an diese erste Explosion heran. Aber trotzdem, die Zahl der erfolgreich finanzierten Spiele, vor allem Brettspiele, steigt stetig an. Und das ist okay. Mir ist es lieber, wenn 100 Spieleprojekte erfolgreich sind, von denen jedes 10.000 Dollar bekommt, anstatt fünf Spiele, die 20 Millionen sammeln. Das passt irgendwie besser zu unserer Vision.

Aber die größere Frage ist meiner Meinung nach: Gab es eine Art Blase, die geplatzt ist? Ich denke, es gibt weniger von diesem reinen, ungehemmten Enthusiasmus und Optimismus gegenüber allem – was vermutlich auch ganz gesund ist. Die Leute sehen sich die Projekte an und fragen sich: »Wie sehr wünsche ich mir, dass das Projekt umgesetzt wird? Wie engagiert bin ich?« Wenn jemand nur ein neues Spiel haben möchte und 20 Dollar übrig hat, kannst du auch einfach auf Steam gehen und dir was herunterladen. Darum geht es nicht. Es geht darum, Teil der Entstehung von etwas zu sein und einen Bezug dazu zu haben. Ich finde es in Ordnung, die Maßstäbe daran, was man unterstützen möchte, etwas höher anzulegen. Viele der Leute, die wegen Double Fine und »Broken Age« verärgert waren, gehören vermutlich zu denen, die zwei Dollar für ein iOS-Spiel ausgeben und dann die nächsten zwei Tage spielen möchten. Das können sie ja auch machen. Aber uns geht es darum, einen anderen Blick auf die Entstehung von Kunst und kreativen Produkten zu bekommen. Ich fühle mich damit wohl, dass es nicht jeden Tag einen neuen Million-Dollar-Blockbuster gibt. Jetzt ist es eben einer alle paar Tage (lacht). Vor kurzem hat der »Castlevania«-Schöpfer Koji Igarashi eine Kampagne gestartet, von der ich vorher keine Ahnung hatte. Nach



Das Video zur Kampagne von »Kentucky Route Zero« hinterließ einen bleibenden Eindruck: »Es war optisch und vom Gamedesign her sehr ansprechend, es lief ein Countrysong ... es war einfach ungewöhnlich und hob sich von der Masse ab.«

einem Tag hatten sie schon um die 1,7 Millionen Dollar. Die Woche davor gab es ein britisches Spiel, das innerhalb eines Tages über eine Million bekommen hat. Diese Geschichten passieren nach wie vor. Was für uns zählt ist, dass wir ein effektives Tool bleiben, mit dem Initiatoren mit ihrem Publikum in Kontakt treten können.

Making Games Unglücklicherweise gab es in der Vergangenheit einige Betrugsfälle. Welche Pläne habt ihr, um Kickstarter in dieser Hinsicht sicherer zu machen?

Yancey Strickler Projekte vollständig umzusetzen ist wichtig, es ist ja so etwas wie unser Zeugnis. Die meiste Arbeit machen wir schon im Vorfeld. Wir prüfen jedes Projekt, bevor es überhaupt live geht. Dabei schauen wir uns täglich etwa 400 Projekte an, damit sind zwölf Leute beschäftigt. Wir führen aber keine tiefgehenden Nachforschungen durch. Stattdessen prüfen wir anhand von Richtlinien, die wir aufgestellt haben, um unserer Meinung nach unpassende Projekte auszusortieren, wie zum Beispiel Spendenaktionen oder eben Dinge, die später eventuell zu Problemen werden könnten. Es gibt also einen Genehmigungsprozess, bevor eine Kampagne live geht.

Es kann natürlich passieren, dass Projekte schiefgehen. Im vergangenen Jahr haben wir unsere Nutzungsbedingungen angepasst, um dazu Stellung zu nehmen und zu sagen: »Natürlich ist es wichtig, dass die Initiatoren das Versprochene auch leisten, aber das wird nicht immer passieren.« Wenn es nicht möglich ist, müssen sie folgende Schritte gehen: Sie müssen ein Post Mortem veröffentlichen, in dem sie genau darlegen, was schiefgelaufen ist. Sie müssen erklären, wie das erhaltene Geld genutzt wurde. Wenn Geld übrig ist, müssen sie anbieten, es den Backern zurückzugeben. Außerdem noch ein paar andere Dinge, aber das ist das Wesentliche. Wenn sie das geleistet haben, haben sie von unserer Seite aus alle Pflichten erfüllt, weil sie transparent dargelegt

»Die beiden Kategorien mit den höchsten Erfolgsraten sind Tanz und Theater.«

»Wir prüfen jedes Projekt, bevor es überhaupt live geht.«

haben, was passiert ist. Wenn sie das jedoch nicht gemacht haben, können Backer theoretisch rechtliche Schritte einleiten.

Making Games Es hat lange gedauert, bis Kickstarter in Deutschland verfügbar war. Warum eigentlich? Gab es rechtliche Fragen, die ihr aus dem Weg schaffen musstet? Musstet ihr Dinge anpassen, um den Launch möglich zu machen?

Yancey Strickler Das ist eine gute Frage. Der größte Punkt war, dass wir bis Anfang 2015 Amazon als Zahlungsdienstleister genutzt haben. Das hat auch gut funktioniert, allerdings aber eingeschränkt, wo wir unsere Plattform verfügbar machen konnten. Das Produkt, das wir nutzten, war außerhalb der USA nicht verfügbar. Wir haben etwa ein Jahr damit verbracht, unsere ganze Zahlungsinfrastruktur zu einem Unternehmen namens Stripe zu verlagern. Das macht es uns möglich, die Plattform überall dort zugänglich zu machen, wo wir es möchten. Dieser Übergang war im Januar beendet. Anschließend war Deutschland der Ort, an dem wir alles in die Wege geleitet haben. Wir hatten schon lange darauf gewartet. Es gab also diese Zahlungskomponente, die zwar nicht sehr sexy oder interessant ist, aber eben dafür sorgt, dass die Plattform überhaupt funktioniert. Wir mussten außerdem die Seite übersetzen, was wir vorher noch nicht gemacht hatten – das war also auch eine Herausforderung.

Es gibt einige Gründe, warum wir zuerst nach Deutschland gegangen sind. Zum einen existiert hier bereits eine beachtliche Backer-Community; die Leute haben bereits mehrere Millionen in Projekte investiert. Es gibt also eine Vertrautheit und ein gewisses Interesse daran. Wir haben außerdem viele Anfragen von deutschen Initiatoren bekommen, Kampagnen zu starten. Davon abgesehen ist Deutschland der größte Markt in ganz Europa, was es zum idealen Ort macht, um eine Präsenz in Europa aufzubauen. Und am interessantesten finde ich, dass die deutsche Kreativkultur große Übereinstimmungen mit einigen unserer stärksten Kategorien aufweist. Es gibt viele starke Ingenieure, eine starke Startup-Kultur,

aber eben auch eine tolle Kreativkultur. Musik, Kunst ... es kommt viel transgressive, exzentrische Kunst aus Deutschland. Das ist unser Ding – das sind die Projekte, die uns interessieren.

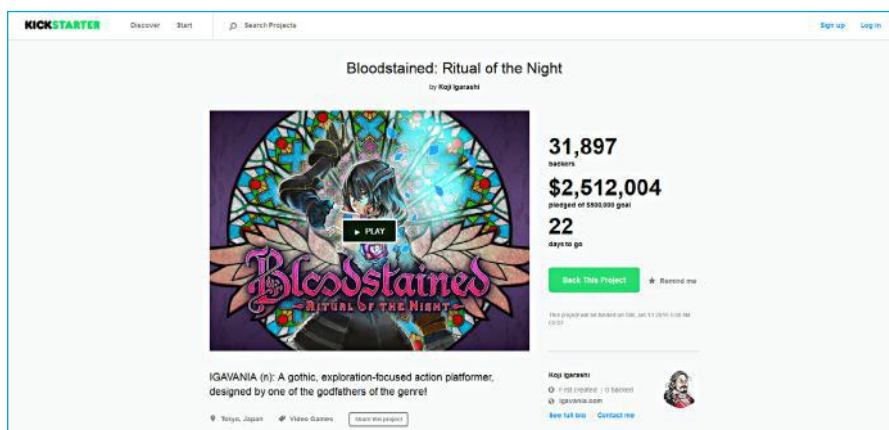
Wir mussten drei Punkte für den Launch in Deutschland anpassen, die wir für wichtig erachtet haben. Zum einen können die Initiatoren in ihren Videos jetzt Untertitel einfügen; das Video kann also auf Deutsch sein, aber mit englischen Untertiteln trotzdem das amerikanische Publikum ansprechen, denn rund 70 Prozent unserer Backer kommen aus den USA. Zum anderen werden die Projekte aus Deutschland hier zwar in Euro angezeigt, in den USA allerdings in Dollar. Also auch wieder eine Bemühung, diese Projekte amerikanischen Backern schmackhaft zu machen. Außerdem erlauben wir jetzt ebenso das Lastschriftverfahren als Zahlungsmöglichkeit. Diese drei Punkte mussten zusammenkommen, bevor wir die Plattform hier launchen konnten.

Making Games Es war theoretisch ja schon vorher möglich, Kampagnen von Deutschland aus zu starten. Welche Vorteile hat der Launch? Und gibt es Unterschiede in den Kampagnen in den verschiedenen Ländern?

Yancey Strickler Es war möglich, von Deutschland aus sein Projekt zu starten, wenn die Initiatoren ein Bankkonto in einem anderen Land hatten. Das war also eine Grauzone. Zudem mussten die Leute größere Anstrengungen unternehmen als meiner Meinung nach notwendig, um unser System zu nutzen. Jetzt kannst du einfach loslegen, wenn du ein deutsches Bankkonto hast.

Was die lokalen Unterschiede angeht: Wir denken ausgiebig über die Vielseitigkeit der Plattform nach, was die Projektkategorien betrifft. Ich habe vorhin ja Tanz erwähnt – das Wohl dieser Kategorie liegt uns sehr am Herzen. Wir wollen immer sicherstellen, dass es zu jedem Zeitpunkt in jeder Kategorie gute Projekte gibt. Wir wollen nicht von einer bestimmten Rubrik dominiert werden, uns interessiert ein breitgefächertes Spektrum. Das haben wir auch in allen Ländern gesehen, in denen Kickstarter bisher verfügbar war. Ich glaube, im ersten Jahr gab es in Großbritannien kein Tanzprojekt, vielleicht hadert die britische Tanzszene momentan. Wir möchten eine Vielfalt an Projekten sehen. Die USA, Kanada und Australien sind sehr ähnlich, Skandinavien unterscheidet sich leicht und ist Design-lastiger. Mal schauen, wie es sich in Deutschland entwickelt. Ich könnte mir vorstellen, dass es viele Hardware-Projekte geben wird. Zum Start gab es auch viele Games-Projekte. In sechs Monaten wird klar sein, wie die Community aussieht, die Kickstarter als Heimat für ihre Projekte ansieht. Ich würde mir wünschen, dass die Community in Deutschland eine individuelle Ausrichtung hat und sich nicht homogen in den bisherigen Tenor einfügt. Wir werden sehen, was die deutschen Entwickler zu sagen haben.

Interview: Patricia Geiger

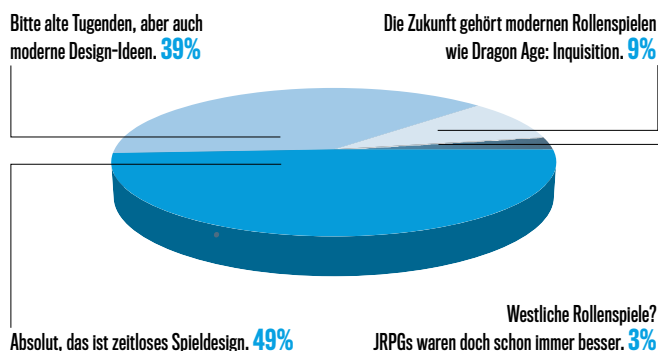


Große Blockbuster-Kampagnen sind etwas seltener geworden, aber immer noch vorhanden: »Der »Castlevania«-Schöpfer Koji Igarashi hat Kampagne gestartet, die nach einem Tag schon um die 1,7 Millionen Dollar hatte. Diese Geschichten passieren nach wie vor.«

DATAFLOW

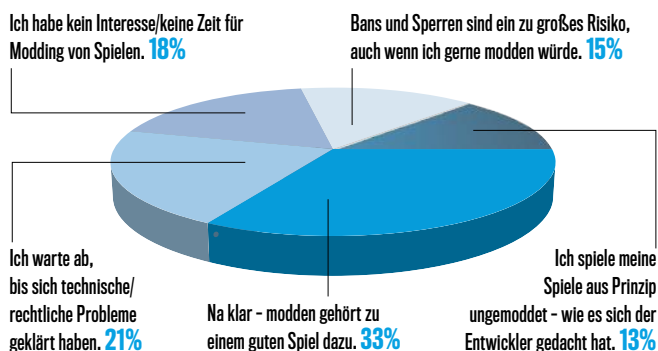
GameStar und GamePro befragen regelmäßig Heftkäufer und Website-Besucher zu aktuellen Games-Themen. Eine kleine Auswahl finden Sie auf dieser Seite.

■ Interessieren Sie sich für klassische Rollenspiele wie Pillars of Eternity?



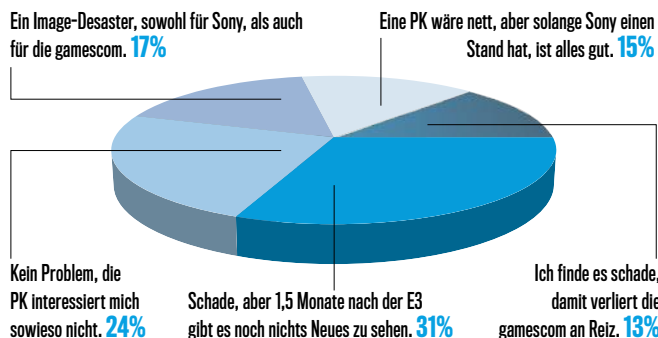
Quelle: Umfrage auf GameStar.de, 14.911 Teilnehmer

■ Modden Sie ihre Spiele trotz Problemen und Bans wie bei GTA 5 und Dark Souls 2?



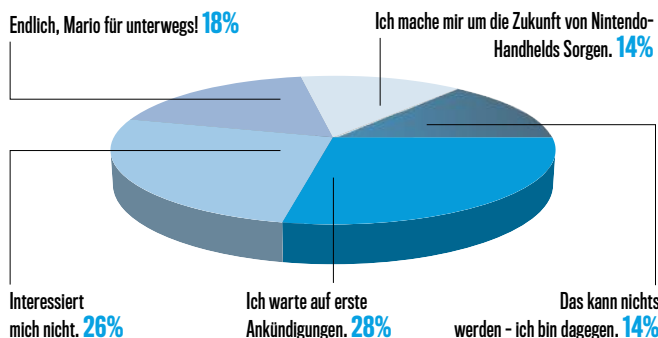
Quelle: Umfrage auf GameStar.de, 6.513 Teilnehmer

■ Stört es Sie, dass Sony keine gamescom-Pressekonferenz haben wird?



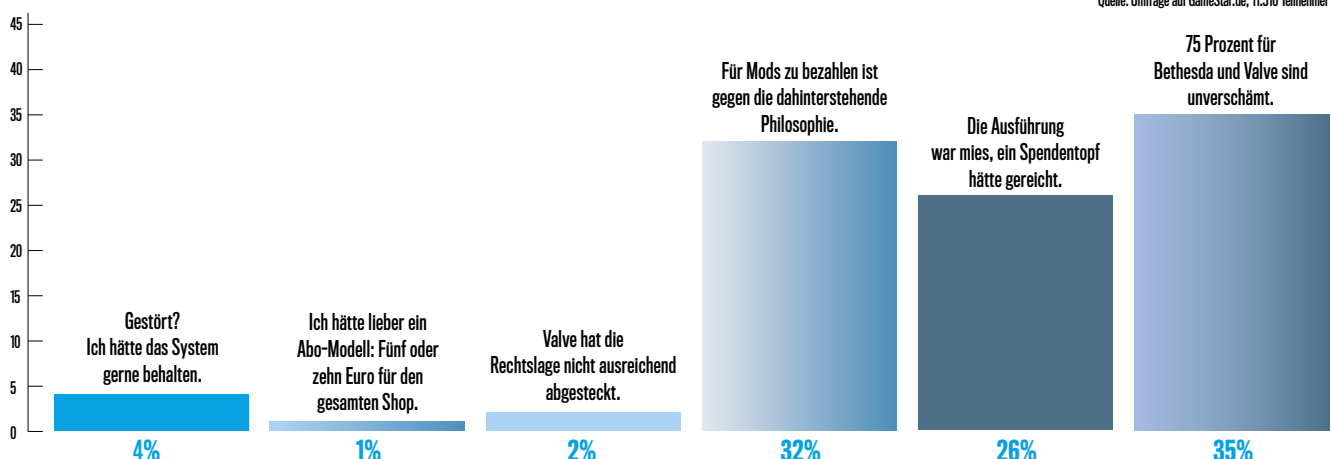
Quelle: Umfrage auf GamePro.de, 1.843 Teilnehmer

■ Was halten Sie von den geplanten Mobile-Spielen von Nintendo?



Quelle: Umfrage auf GamePro.de, 2.927 Teilnehmer

■ Was hat Sie am Bezahlssystem für Skyrim-Mods gestört?



Quelle: Umfrage auf GameStar.de, 11.510 Teilnehmer

www.makinggames.biz

Online-Register mit Suchfunktion. Zu jedem Key Player finden Sie zusätzlich ein detailliertes Firmenportrait.

Helfen Sie uns!

Sie haben einen Fehler entdeckt oder vermissen Ihre Firma? Dann schreiben Sie uns:
info@makinggames.de, Betreff: Firmenregister

DIE BRANCHE IM ÜBERBLICK FIRMENREGISTER

Vergessen Sie die Auskunft! Hier finden Sie sämtliche uns bekannten Publisher, Entwickler, Dienstleister, Ausbilder und Verbände der deutschsprachigen Spiele-Industrie - von 3d-io bis Z-Software.

DEVELOPMENT					
Firma	Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail	
3d-io	Rheingaustraße 94	65203 Wiesbaden	0611 / 5829602	info@3d-io.com	
5d lab	Georges-Köhler-Allee 79	79110 Freiburg	0761 / 2038043	info@5dlab.com	
Adrastea	Bielefelder Str. 5	90425 Nürnberg	0700 / 42300000	info@adrastea.com	
Aesir Interactive	Grafingerstraße 6	81671 München		hq@aesir-interactive.com	
ANIMA Entertainment	Konsul-Smidt-Straße 84	28217 Bremen	0421 / 5784220	info@anima-entertainment.de	
AntMe!	Friedenstraße 33	85221 Dachau		info@antme.net	
B-Alive	Biberacher Straße 38	88444 Ummendorf	07351 / 441307	admin@b-alive.de	
Beardshaker Games	Wartenau 16	22089 Hamburg	040 / 28471900	contact@beardshaker.com	
BIGITEC	Theaterstraße 22	53111 Bonn	0228 / 42976585	info@bigitec.com	
BIGPOINT	Drehbahn 47-48	20354 Hamburg	040 / 8814130	info@bigpoint.net	
b-interaktive	Lohbachstraße 12	58239 Schwerte	02304 / 3382391	info@binteraktive.com	
Bit Barons c/o Wailua Games	Schellingstraße 35	80799 München	089 / 41617744	info@bitbarons.com	
Black Forest Games	Innovationszentrum, Hauptstraße 1a	77652 Offenburg	0781 / 31057910	info@bfgames.biz	
Blazing Badger	Niddastraße 64	60329 Frankfurt am Main		info@blazingbadger.com	
Blue Byte	Adlerstraße 74	40211 Düsseldorf	0211 / 338000	info@bluebyte.de	
Blue Byte Mainz	Römerpassage 1	55116 Mainz	06131 / 554470	info@related-designs.de	
Bongfish	Schönaugasse 8a/2	A-8010 Graz	+43 / 676 / 7822203	info@bongfish.com	
Bright Future	Gustav-Heinemann-Ufer 56	50968 Köln	0221 / 78982171	info@brightfuture.de	
Brightside Games	Marie-Elisabeth-Lüders-Straße 1	10625 Berlin	0177 / 3354447		
Broken Rules Interactive Media	Johann Strauss Gasse 20/8	A-1040 Wien		hello@brokenrul.es	
Bronx Studios Entertainment	Olof-Palme-Straße 13	60439 Frankfurt	069 / 870050660	info@bronxstudios.com	
BSS web consulting	Savignystraße 74	45147 Essen	0201 / 5200044	info@bss-consulting.de	
Buntware Games	Colmarerstrasse 9	79106 Freiburg	0761 / 3891598	contact@buntware.com	
Bumblebee	Gerresheimer Straße 7	51674 Wiehl	02261 / 4795001	tassilo@bumblebee-games.com	
ByteRockers	Schivelbeiner Straße 1	10439 Berlin	030 / 232573390	info@byterockers.de	
Bytro Labs	Rödingsmarkt 29	20459 Hamburg		info@bytro.com	
Caipirinha Games	Kiefernweg 1a	24161 Altenholz	0431 / 324540	info@caipirinhagames.de	
Candygun Games	Hamburger Strasse 132	22083 Hamburg	040 / 20947878		
cerasus.media	Rosenthaler Straße 51	10178 Berlin	030 / 802089990	info@cerasus.de	
Chasing Carrots	Hornbergstraße 94	70188 Stuttgart	0711 / 64519353	office@chasing-carrots.com	
Chimera Entertainment	Karlstraße 68	80335 München	089 / 210205790	info@chimera-entertainment.com	
CipSoft	Prüfener Straße 20	93049 Regensburg	0941 / 6308280	kontakt@cipsoft.com	
Cliffhanger Productions	Seidengasse 32/NB/7	A-1070 Wien		office@cliffhangerproductions.com	
ClockStone STUDIO	Claudiastraße 14	6020 Innsbruck	+43 / 512 / 890377	info@clockstone.com	
Cooee	Europaallee 7-9	67657 Kaiserslautern	0900 / 1002273		
Crafty Studios	Lichtenwörthgasse 112a	A-7033 Pötsching		info@craftystudios.com	
Crazy Bunch	Billhorne Röhrendamm 8	20539 Hamburg	0173 / 4361509	team@crazybunch.biz	
createale	Mörkestraße 28/2	71636 Ludwigsburg	07141 / 2399563	contact@createale.de	
CreaTeam Software	Sünderup 6	24943 Flensburg	0461 / 64808		
Crenetic	Kreuzstraße 3	45468 Mülheim an der Ruhr	0208 / 4442142	info@crenetic.de	
Crytek	Grüneburgweg 16-18	60322 Frankfurt am Main	069 / 21977660	info@crytek.de	
Cyber Arena Entertainment Group	Sebastian Kneippgasse 30	2380 Perchtoldsdorf		info@c-arena.com	
Dachshund Games	Turmstraße 48	10551 Berlin		team@dachshund-games.com	
Daedalic Entertainment	Papenreue 51	22453 Hamburg	040 / 43261270	info@daedalic.de	
Decane	Hallerstraße 5e	20146 Hamburg	040 / 42910947	info@decane.net	
Deck13 Interactive	Gutleutstraße 82	60329 Frankfurt am Main	069 / 71671660	info@deck13.com	
Deep Silver Fishlabs	Großer Burstah 50-52	20457 Hamburg	040 / 8880000	business@fishlabs.net	
Destrax Entertainment	Merkurstraße 10	33739 Bielefeld		info@destrax.de	
Digitalmindsoft	Wichernstraße 10	89073 Ulm		info@digitalmindsoft.eu	
Doublesmith	Ranzonigasse 3/3	A-3100 St. Pölten	+43 / 699 / 19447880	mail@doublesmith.com	
dreINET	Bergedorfer Straße 96	12623 Berlin		philip@dreinet.org	
eelusion	Paul-Lincke-Ufer 39	10999 Berlin		contact@eelusion.com	
Egosoft	Heidestraße 4	52146 Würselen	02405 / 4239970	egoinfo@egosoft.com	
Enter-Brain-Ment	Littenstraße 106/107	10179 Berlin		interesse@enter-brain-ment.com	
Envision Entertainment	Binger Straße 38	55218 Ingelheim		info@envision-entertainment.de	
eos interactive	Carl-Zeller Straße 16	75015 Berlin	07252 / 957757	info@eosinteractive.de	
exDream	Hildesheimer Straße 5	30169 Hannover	0511 / 37077675	info@exdream.com	
exozet	Platz der Luftbrücke 4-6	12101 Berlin	030 / 2465600	games@exozet.com	
extra toxic	Hanauer Landstraße 196a	60314 Frankfurt am Main	069 / 48440040	info@extratoxic.com	
eyefactive	Feldstraße 128	22880 Wedel	04103 / 903800	info@eyefactive.com	
FAKT Software	Dufourstraße 28	04107 Leipzig	0341 / 14990564	info@fakt-software.de	
Fameside Entertainment	Storkower Straße 158	10407 Berlin	030 / 40577053	info@fameside.com	

DEVELOPMENT					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Fancy Bytes		Niederstraße 5	26931 Huntorf	02151 / 3259912	biz@fancy-bytes.de
FDG Entertainment		Agnes-Pockels-Bogen 1	80992 München	089 / 18941370	info@fdgertainment.com
flaregames		Kaiserstraße 146-148	76133 Karlsruhe	0721 / 1831540	info@flaregames.com
Frame6		Mindener Straße 30	40227 Düsseldorf	0211 / 23938236	contact@frame6.de
Freudenreich & Tollknaepper		Paracelsusstraße 57	42283 Wuppertal	0202 / 2721048	mail@german-railroads.com
Funatics Software		Essener Straße 66	46047 Oberhausen	0208 / 3066670	info@funatics.de
funworld		Photo Play Straße 1	A-4860 Lenzing	+43 / 7662 / 67850	info@photoplay.com
Fusionsphere Systems		Rosenu 8	86551 Aichach	0700 / 38746600	info@fusionsphere.com
GameArt Studio		Bismarckstraße 107	10625 Berlin		kontakt@GameArtStudio.de
GameDuell		Taubenstraße 24-25	10117 Berlin	030 / 288768210	impressum@gameduell.de
Gameforge		Albert-Nestler-Straße 8	76131 Karlsruhe	0721 / 3548080	info@gameforge.de
gameus		Kufsternerstraße 40	83064 Raubling	08035 / 6689	info@gameus.de
Gaming Minds Studios		Georgstraße 6	33330 Gütersloh		contact@gamingmindsstudios.com
Gbanga		Bräuerstraße 31	8004 Zürich	+41 / 43 / 5366701	matthias@gbanga.com
GIANTS Software GmbH		Rütistrasse 18	8952 Zürich-Schlieren	+41 / 44 / 5082176	mail_us@giants-software.com
Goal Games		Feurigstraße 54	10827 Berlin		info@goal-games.de
goodbeans		Winsstraße 62	10405 Berlin	030 / 802087400	jobs@goodbeans.com
Goodgame Studios		Theodorstraße 42-90, Haus 9	22761 Hamburg	040 / 2198800	pr@goodgamestudios.com
Greentube		Mariahilfer Straße 47/1/102	A-1060 Wien	+43 / 1 / 4945056	office@greentube.com
HandyGames		i_Park Klingholz 13	97232 Giebelstadt	09334 / 97570	info@handy-games.com
iLogos Europe		Pezoldamm 98a	22715 Hamburg	0160 / 7002927	alexander@ilogos.biz
Independent Arts Software		Münsterstraße 5 / HAMTEC Haus 2b	59065 Hamm	02381 / 488818	info@independent-arts-software.de
InnoGames		Friesenstraße 13	20097 Hamburg	040 / 78893350	info@innogames.de
Instance Four		Friederikastraße 65	44789 Bochum	0234 / 89038971	info@instancefour.com
Island Games		Vahrenwalder Straße 7	30165 Hannover	0511 / 35319898	kontakt@islandgames.de
it Matters Games		Am Studio 2A	12489 Berlin		info@itmattersgames.com
Jo-Mei		Monumentenstraße 35, Aufg. C	10829 Berlin	030 / 81494080	info@jo-mei.com
K1010 Media		Erkelenzdam 59-61	10999 Berlin	030 / 57709936	info@k1010.de
keen games		Dreieichstraße 59	60594 Frankfurt am Main	069 / 66058838	info@keengames.com
King Art		Martinstraße 47-49	28195 Bremen	0421 / 322760	contact@kingart.de
KORION Simulation		Mömpelgardstraße 16	71640 Ludwigsburg	07141 / 3898231	info@korion.de
kr3m.media		Kaiserstraße 158	76133 Karlsruhe	0721 / 18395918	
Kritzelkratz 3000		Dieselstraße 2-6	97082 Würzburg	0931 / 3536091	info@kritzelkratz.de
kunst-stoff		Kottbusser Damm 73	10967 Berlin	030 / 23457178	info@kunst-stoff.de
Limbic Entertainment		Heinrich-Hertz-Straße 21	63225 Langen	06103 / 5019780	info@limbic-entertainment.de
MADE		Alter Schlachthof 39	76131 Karlsruhe	0721 / 78809460	
MaDe Games GbR Neetix - Wycislik		Hochstraße 34	41334 Nettetal		info@madegames.com
McPeppergames		Nikolaus-Molitor-Straße 11a	97702 Münnerstadt	09733 / 761006	info@mcpeppergames.com
Media Seasons		Liviastraße 6	4105 Leipzig	0341 / 2156752	contact@mediasaisons.com
Mex IT		Mallinckrodtstraße 320	44147 Dortmund	0231 / 58680722	mex@mex-solutions.com
Mimimi Productions		Berg-am-Laim-Straße 64	81673 München	089 / 809115070	info@mimimi-productions.de
Mind Elevator Games		Prenzlauer Promenade 190	13189 Berlin	030 / 91744394	info@mindelevatorgames.com
mobivention		Gottfried-Hagen-Straße 24	51105 Köln	0221 / 6778110	info@mobivention.com
Moonbyte Games		Wormser Landstraße 17	67346 Speyer	06232 / 8774482	info@moonbytegames.com
NeoBird		Südwestpark 37-41	90449 Nürnberg	0911 / 3092097	info@neobird.de
netmin games		Philipp-Reis-Straße 6	55129 Mainz	06131 / 507986	schreiber@netmin.de
northworks		Stahlwiese 23	22761 Hamburg	040 / 78107075	info@northworks.de
Noumena Studios		Lützowstraße 33	10785 Berlin	030 / 53148825	info@noumenastudios.com
NovaTrix / Animedo		Schlüsselwiesen 19	70186 Stuttgart	0711 / 3102770	info@animedo.de
Nurogames GmbH		Schaafenstraße 25	50676 Köln	0221 / 3988040	info@nurogames.com
online4ever Limited		Manitiusstraße 6	1067 Dresden	0351 / 2063875	info@online4ever.biz
OFM - OnlineFussballManager		Eupener Straße 60	50933 Köln	0221 / 88892920	office@onlinefussballmanager.de
Outline Development		Hauptstraße 122	57074 Siegen	0271 / 3131331	webmaster@outline-development.de
Pappuga		Fürther Straße 212	90429 Nürnberg	0911 / 32378310	info@pappuga.com
Phobeta		Wiesentalstraße 5	90419 Nürnberg	0911 / 9389292	wendt@phobeta.de
Piranha Bytes		Ruhrallee 63	45138 Essen	0201 / 806720	info.nospam@piranha-bytes.com
pixeltamer.net		Clausewitzstraße 6	10629 Berlin	030 / 34347690	welcome@pixeltamer.net
Pixon		Jaffestraße 10	21109 Hamburg	040 / 33399430	welcome@pixion.de
Playnik		Im Derdel 17	48161 Münster	02534 / 9733530	info@playnik.de
playzo		Heinrich-Herz- Straße 6	64295 Darmstadt	06151 / 6673700	info@playzo.biz
plazz entertainment		Augustmauer 1	99084 Erfurt	0361 / 2169460	info@plazz-entertainment.com
Pop Rocket Games		Gasstraße 10	22761 Hamburg	040 / 822449040	
Procontis		Am Bauhof 18	64807 Dieburg	06071 / 430111	sales@procontis.de
Promotion Software		Karlstraße 3	72072 Tübingen	07071 / 91670	mail@promotion-software.de
Quadrige Games		August-Bebel-Straße 27	14482 Potsdam	0331 / 23189660	info@quadrigagames.com
Rat King Entertainment		Robert-Franz-Ring 3	06108 Halle	0345 / 6141741	mail@ratking.de
Ravensburger Digital		Kaflerstraße 8	81241 München	089 / 12228180	tellme@ravensburger-digital.com
Reality Twist		Berg-Am-Laim-Straße 64	81673 München	089 / 380129500	mail@reality-twist.com
RealForge Studios		Hofer Straße 15	81737 München	089 / 55069134	info@realforgestudios.com
RedMoon Studios		Dieselstrasse 16	86368 Gersthofen	0821 / 45598000	info@redmoonstudios.de
REDOX Game Labs		Jakob-Haringer Straße 5	A-5020 Salzburg	+43 / 662 / 273273	info@redox-labs.com
remote control productions		Karlstraße 68	80335 München	089 / 21020570	info@r-control.de
RockABYTE		Schaafenstraße 25	50676 Köln	0221 / 80147950	info@rockabyte.com
SAFKAS Development		Schulze-Delitzsch-Straße 39	04315 Leipzig		mail@safkas.de
Sandbox		Bornholmer Strasse 71	10439 Berlin	030 / 40045181	info@sandbox-interactive.com
Scorpius Forge		Hörvelsingerg Weg 29-31	89081 Ulm	0731 / 14410099	info@scorpius-forge.de
Serious Games Solutions		August-Bebel-Straße 27	14482 Potsdam	0331 / 23189680	mail@serious-games-solutions.com
Sharkbom Studios		Durlacher Allee 53	76131 Karlsruhe	0160 / 92502633	contact@sharkbombs.de
Silent Dreams		Kreuzstraße 3	45468 Mülheim an der Ruhr		contact@silentdreams.de
SilentFuture		Triebelsheide 37b	42111 Wuppertal	0202 / 3935366	info@silentfuture.de
Simlity		Moritz-von-Rohr-Straße 1a	07745 Jena	03641/3279684	info@simlity.com
Skunk Brothers		Dieselstraße 12	61191 Rosbach		hello@skunkbrothers.de
SlipShift		Friedrichstraße 122	10117 Berlin	030 / 27581581	info@slipshift.net
Snowstep Development		Heinrich-Holtzschneider-Weg 76	40489 Düsseldorf	0201 / 17848176	business@snowstep.com

DEVELOPMENT					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Socialspiel		Schmalzhofgasse 26 / Top 5-6	A-1060 Wien		office@socialspiel.com
SOLID WHITE design digital media		Mönchhaldenstraße 27A	70191 Stuttgart	0711 / 12375149	contact@solidwhite.de
Solimedia Productions		Bahnhofstr. 4a	99084 Erfurt	0361 / 658550	hatmut.niemeyer@solimedia.de
Sparrow Games		Wegenerstraße 1	13088 Berlin	030 / 50915356	mail@sparrowgames.de
SpinBottle Games		Weidkamp 3	45355 Essen	0201 / 1789614	info@spinbottlegames.com
Spirit		Friedrichstraße 15	70174 Stuttgart	0711 / 96685510	weyrich@godspirit.com
Splitscreen Studios		Neuer Kamp 30	20357 Hamburg		info@splitscreenstudios.com
Springwald Software		Alter Eistreff 36	44789 Bochum	0700 / 777464925	info@springwald.de
Sproing Interactive Media	key	Fernkorngasse 10	A-1100 Wien	+43 / 1 / 60430280	office@sproing.com
stillalive studios		Rosengasse 1	A-6020 Innsbruck	+43 / 650 / 3468942	info@stillalive-studios.com
Studio Fizbin		Hoferstraße 20	71636 Ludwigsburg	07141 / 9923866	hello@studio-fizbin.de
Sunlight Games		Im Klapperhof 7-23	50670 Köln	0221 / 16823466	kontakt@sunlight-games.com
symlCrowd		In der Schleh 27	5224 Eschweiler	0152 / 55878044	apps@symlcrowd.de
Team Vienna Games		Kaiserstrasse 84/1/6	A-1070 Wien		office@teamviennagames.com
TG Nord		Große Düwelstraße 28	30171 Hannover	0511 / 9409208	mail@tg-nord.com
the Good Evil		Gilbachstraße 22	50672 Köln	0221 / 16894248	hello@thegoodevil.com
Travian Games		Wilhelm-Wagenfeld-Straße 22	80807 München	089 / 3249150	mail@traviangames.com
unikat media		Eckdrift 10	19061 Schwerin	0385 / 64105688	info@unikatmedia.de
upjers	key	Hafenstraße 13	96052 Bamberg	0951 / 51098100	mail@upjers.com
VisionaryX		Friedrichstraße 12	71101 Schönaich	07031 / 2850 / 414	contact@visionaryx.com
V-Play		Kolonitzgasse 9/11-14	A-1030 Wien		team@v-play.net
Wolpertinger Games		Karlstraße 68	80335 München	089 / 210205700	contact@wolpertingergames.com
wooga		Saarbrücker Straße 38	10405 Berlin	030 / 32505294	info@wooga.com
Xendex		Mariahilfer Strasse 176/6	A-1150 Wien	+43 / 1 / 89080090	office@xendex.com
XYRALITY	key	Friedensallee 290	22763 Hamburg	0 / 4035730010	info@xyrality.com
YAGER Development		Pfuelstraße 5	10997 Berlin	030 / 69597660	contact@yager.de
zeitland media & games		Mathildenstraße 10/1	71638 Ludwigsburg	07141 / 5050565	contact@zeitland.com
Zeroscale		Schwedter Straße 9a	10119 Berlin	030 / 405057929	info@zeroscale.com
Zone 2 Connect		Fürstenwall 70	40219 Düsseldorf	0211 / 5422400	info@zone2.de
Z-Software		Lindemannstrasse 81	44137 Dortmund	0231 / 33015031	contact@z-software.net
FREELANCER DEVELOPMENT					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Caspar Siebel		Theresienhöhe 1	80339 München	017661615291	mail@casparsiebel.com
Christian Kuchmeister (3D Artist / Game Design)		Lauensteinstraße 9	22307 Hamburg	0163 / 4841841	ck@ckuechmeister.de
Eric Jannot (Game Design, Game Writing)		Sonnenallee 70	12045 Berlin	0178 / 5223697	
Filippo Beck Peccoz	key	Franz-Joseph-Straße 19	80801 München	089 / 80033204	mail@fbpsound.com
Jochen Hamma (Producing)	key	Wehinger Weg 9	78583 Boettingen	07429 / 910793	jochen.hammer@fantastic-realms.com
Jörg Beilschmidt (Game Design)		Berliner Ring 20	21406 Melbeck	0172 / 4311243	j.beilschmidt@creatown.de
Kim Lange (Full Service Agency)		Landwehr 192	46045 Oberhausen	0178 / 2375418	lange@eins2design.de
Marco Sowa (Producing, Development)		Postfach 420566	50899 Köln	0221 / 16916844	mail@marcosowa.de
Martin Linnartz (3D Artist, Game Developer)		Achenbachstraße 19	40237 Düsseldorf	0211 / 17838926	kontakt@cubicity.de
Martin Nerurkar (Game Design)		Durlacher Allee 53	76131 Karlsruhe		mnerurkar@sharkbombs.com
Nico Nowarra (Storytelling)		Friedbergstraße 9	14057 Berlin	030 / 31801899	nno@menintext.com
Niels Bauer Games (Game Design)		Sickingenstraße 37	79117 Freiburg	0761 / 66947	contact@nielsbauergames.com
Pinky Pills (2D-Grafik, Game Design, Concept)		Swindonstraße 162	38226 Salzgitter	05341 / 2886036	pinkypills@arcor.de
Thomas Obermaier (Programmierung)		Kapellenstraße 1	86869 Gutenberg	0170 / 3815144	contact@polyforming.com
Wolfgang Walk (Full Scale Producing)		Karl-Leopold-Straße 6	76229 Karlsruhe	0721 / 48399963	wolfgang.walk@t-online.de
PUBLISHING/DISTRIBUTION					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
505 Games		Brunnfeld 2-6	93133 Burglengenfeld	09471 / 3088792	
Activision Blizzard Deutschland		Fraunhoferstraße 7	85737 Ismaning	089 / 9998170	info@activision.de
ad2games	key	Rosenstraße 17	10178 Berlin	030 / 240888221	info@ad2games.com
Aeria Games Europe		Schlesische Straße 27, Aufgang C	10997 Berlin		info@aeriagames.com
Aerosoft		Lindberghring 12	33142 Bielefeld	02955 / 760310	info@aerosoft.de
Application Systems Heidelberg		Pleikartsförsterhof 4/1	69124 Heidelberg	06221 / 300002	kontakt@application-systems.de
Applift	key	Rosentraße 17	10178 Berlin	030 / 240888208	info@applift.com
astragon Software	key	Limitenstrasse 64-78	41236 Mönchengladbach	02166 / 146450	info@astragon.de
BHV		Novesiastraße 60	41564 Kaarst	02131 / 76501	info@bhv.de
Bigpoint	key	Drehbahn 47-48	20354 Hamburg	040 / 8814130	info@bigpoint.net
bitComposer Entertainment		Mergenthalerallee 79-81	65760 Eschborn	06196 / 7793810	info@bit-composer.com
Capcom Entertainment Germany GmbH		Borselstraße 20	22765 Hamburg	040 / 6965620	info@capcomeuro.de
Covus Crobo	key	Schwedter Straße 263	10119 Berlin		info@crobo.com
Crimson Cow		Tarpen 40, Haus 5	22419 Hamburg	040 / 59355237	info@crimsoncow.de
Crytek	key	Grüneburgweg 16-18	60322 Frankfurt am Main	069 / 21977660	cryengine@crytek.com
Daedalic Entertainment	key	Papenreye 53-Workport Unit 2	22453 Hamburg	040 / 43261270	info@daedalic.de
dreamfab		Domplatz 3	93047 Regensburg	0941 / 569591520	
dtp entertainment		Goldbekplatz 3-5	22303 Hamburg	040 / 6699100	info@dtp-entertainment.com
dtp young entertainment		Goldbekplatz 3-5	22303 Hamburg	040 / 6699100	info@dtp-young.com
Electronic Arts		Im Zollhafen 15-17	50678 Köln	0221 / 975820	
European Games Group		Wiener Platz 7	81667 München		contact@gamesgroup.eu
EuroVideo Medien	key	Oskar-Messter-Straße 15	85737 Ismaning	089 / 9624440	info@eurovideo.de
flashpoint	key	Valvo Park - Haus 5c / Tarpen 40	22419 Hamburg	040 / 710060	info@flashpoint.de
GameDuell		Taubenstraße 24-25	10117 Berlin	030 / 288768211	impressum@gameduell.de
Gameforge	key	Albert-Nestler-Straße 8	76131 Karlsruhe	0721 / 3548080	info@gameforge.de
GameGenetics	key	Alte Jakobstraße 85-86	10179 Berlin	030 / 922512769	info@gamegenetics.de
GameLoft		Karl-Liebknecht-Straße 5	10118 Berlin	030 / 20188429	gregory.wintgens@gameloft.com
GameTwist		Mariahilfer Straße 47/1/102	10600 Wien	+43 / 1494 / 5056	3rdparty@gametwist.com
GamesInFlames		Karlstraße 68	80335 München	089 / 210205715	info@gamesinflames.com
gamigo	key	Behringstraße 16b	22765 Hamburg	040 / 4118850	info@gamigo.com
Halycon Media		An der Autobahn 24	23858 Reinfeld	04533 / 61090	info@halycon.de
Headup Games	key	Nordstraße 102	52353 Düren	02421 / 4868700	info@headupgames.com

PUBLISHING/DISTRIBUTION					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
HitFox Group		Rosenstraße 17	10178 Berlin	030 / 240888200	info@hitfoxgroup.com
Infernum Productions	key player	Uhlandstraße 175	10719 Berlin	030 / 91201090	contact@infernum.com
InnoGames		Friesenstraße 13	20097 Hamburg	040 / 78893350	info@innogames.de
Intenium		Neuer Pferdemarkt 1	20359 Hamburg	040 / 25494296	info@intenium.de
Kalypso Media Group	key player	Wilhelm-Leuschner-Straße 11-13	67547 Worms	06241 / 5061900	info@kalypsomedia.com
Koch Media	key player	Lochhamer Straße 9	82152 Planegg/München	089 / 242450	office@kochmedia.de
Konami		Berner Straße 103-105	60437 Frankfurt am Main	069 / 9855730	support@konami.de
McGame.com		Fürstenwall 69	40217 Düsseldorf	0211 / 63550100	support@mcgame.com
Microsoft		Konrad-Zuse-Straße 1	85716 Unterschleißheim	089 / 31760	kunden@microsoft.com
Millform		Ringstraße 110	CH-4106 Therwil	+41 / 61 / 7220712	werner.sala@millform.ch
Morphicon Limited	key player	Lindwurmstraße 88	80337 München	089 / 45235460	info@morphicon.de
Namco Bandai Partners		Carl-Benz-Straße 21	60386 Frankfurt am Main	069 / 24449100	de.info@namcobandaipartners.com
NBG		Brunnfeld 2-6	93133 Burglengenfeld	09471 / 70170	info@nbg-online.de
Nintendo		Nintendo Center	63760 Grossostheim	06026 / 9500	info@nintendo.de
Nordic Games		Landstraßer Hauptstraße 1/18	A-1030 Wien	+43 / 1 / 2365487-0	office@nordicgames.at
Novitas		Frankfurter Straße 2	38122 Braunschweig	02574 / 9270	kontakt@greenpepper.de
Phenomenia		An der Sandkull 81a	47445 Moers	02327 / 9970	info@phenomenia.com
Playa Games	key player	Alstertor 9	20095 Hamburg	040 / 76793220	info@playa-games.com
Players Rock Entertainment		Goseriede 4	30159 Hannover	0511 / 3539940	contact@playersrock.com
ProSiebenSat.1 Games		Medienallee 6	85774 Unterföhring	089 / 950710	info@sevengames.de
Ravensburger Digital	key player	Kaflerstraße 8	81241 München	089 / 12228180	tellme@ravensburger-digital.com
Reality Twist		Berg-Am-Laim-Straße 64	81673 München	089 / 380129500	mail@reality-twist.com
redspotgames		Postfach 310464	80104 München	089 / 4036370	info@redspotgames.com
Rondomedia	key player	Limtenstraße 64-78	41236 Mönchengladbach	02166 / 618660	info@rondomedia.de
RTL Games		Achnerstraße 1036	50858 Köln	0221 / 4560	entwickler@rtl-interactive.de
Rumble Media		Kriegsstraße 5	76137 Karlsruhe	0721 / 2018020	info@fettspielen.de
S.A.D.		Rötelbachstraße 91	89079 Ulm	07305 / 96290	kontakt@s-a-d.de
Sega		Rahlstraße 24	81669 München	089 / 22848000	info@sega.de
SERU		Industriestraße 8	24589 Nortorf	04392 / 84900	info@seru.de
Sony Computer Entertainment		Frankfurter Straße 233	63263 Neu-Isenburg	06102 / 4330	scee_presscentre@scee.net
Square Enix		Domstraße 17	20095 Hamburg	040 / 30633400	
Take-Two Interactive		Agnesstraße 14	80798 München	089 / 278220	info@take2.de
Thumbr		Pfuehlstraße 5	10997 Berlin	030 / 300137100	info@thumbr.com
Tipp24 Entertainment		Straßenbahnring 11	20251 Hamburg		info@t24e.de
Tivola Publishing		Overseestraße 10-12	22769 Hamburg	040 / 7070063	mail@tivola.de
TopWare Interactive		Rittnerstraße 36	76227 Karlsruhe	0721 / 9151010	info@topware.de
Traffic Captain		Wendenstraße 21b	20097 Hamburg	040 / 23706800	info@trafficcaptain.com
Travian Games		Wilhelm-Wagenfeld-Straße 22	80807 München	089 / 3249150	mail@traviangames.com
Ubisoft	key player	Adlerstraße 74	40211 Düsseldorf	0211 / 338000	info@ubisoft.de
United Soft Media Verlag		Thomas-Wimmer-Ring 11	80539 München	089 / 29088175	info@usm.de
upjers	key player	Hafenstraße 13	96052 Bamberg	0951 / 510908100	mail@upjers.com
ZeniMax Germany		Schillerstraße 15/17	60313 Frankfurt am Main		kontakt.de@zenimax.com
PUBLIC SECTOR					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
BIU	key player	Rungestraße 18	10179 Berlin	030 / 24087790	info@biu-online.de
FilmFernsehFonds Bayern	key player	Sonnenstraße 21	80331 München	089 / 5446020	filmfoerderung@ff-bayern.de
Film- und Medienstiftung NRW	key player	Kaistrasse 14	40221 Düsseldorf	0211 / 9305000	info@filmstiftung.de
G.A.M.E.	key player	Französische Straße 48	10117 Berlin	030 / 346550980	info@game-bundesverband.de
gamearea FRM		Holzhausenstraße 22	60322 Frankfurt am Main	069 / 94419441	info@gamearea-frm.de
gamecity:Hamburg	key player	Habichtstraße 41	22305 Hamburg	040 / 2270190	info@gamecity-hamburg.de
GAMEplaces		Hanauer Landstraße 126-128	60314 Frankfurt am Main	069 / 21236214	manuela.schiffner@gameplaces.de
games.net berlinbrandenburg	key player	Ackerstraße 3A	10115 Berlin	030 / 24628570	games.net@medianet-bb.de
GDI.Ruhr		Kreuzstraße 1	45468 Mülheim an der Ruhr	0208 / 69801520	info@games-ruhr.com
Hessen-IT		Abraham-Lincoln-Straße 38 - 42	65189 Wiesbaden	0611 / 7748481	info@hessen-it.de
mediaboard berlin-brandenburg	key player	August-Bebel-Straße 26-53	14482 Potsdam-Babelsberg	0331 / 743870	info@mediaboard.de
Mitteldeutsche Medienförderung	key player	Hainstraße 17-19	4109 Leipzig	0341 / 269870	info@mdm-online.de
nordmedia	key player	Expo-Plaza 1	30539 Hannover	0511 / 1234560	info@nordmedia.de
Northstar Developers		Bödekerstr 84	30161 Hannover	0160 / 97880272	d.brueinig@collision-studios.com
OVUS		Aichhornaße 14	A-1120 Wien	+43 / 1810 / 328912	office@ovus.at
Stiftung Digitale Spielkultur		Torstraße 6	10119 Berlin	030 / 29049290	kontakt@stiftung-digitale-spielkultur.de
USK	key player	Torstraße 6	10119 Berlin	030 / 24088660	kontakt@usk.de
WERK1 München	key player	Grafinger Straße 6	81671 München	089 / 9952990	info@werk1muenchen.de
EDUCATION					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Acagamics e.V.		Universitätsplatz 2, G29	39106 Magdeburg		info@acagamics.de
Akademie für Medien		Balinger Straße 39A	70567 Stuttgart	0711 / 4209444	h.ehlers@amk-net.de
BTk - Hochschule für Gestaltung		Bernburger Straße 24-25	10963 Berlin	030 / 338539510	berlin@btk-fh.de
Cologne Game Lab / Fachhochschule Köln		Ubierring 40	50678 Köln	0221 / 82753095	info@colognegamelab.de
Designhochschule Schwerin		Bergstraße 38	19055 Schwerin	0385 / 5559775	info@designschule.de
EC Europa Campus		Lyonerstraße 34	60528 Frankfurt am Main	0621 / 8425660	
FH OÖ Studienbetriebs GmbH		Softwarepark 11	A-4232 Hagenberg	+43 (0) 5080420	info@fh-hagenberg.at
FH Trier		Schneidershof	54293 Trier	0651 / 8103345	sek@informatik.fh-trier.de
FH Technikum Wien		Mariahilfer Straße 37-39	1060 Wien	+43 / 1 / 588390	info@technikum-wien.at
FH Salzburg		Urstein Süd 1	A-5412 Puch/Hallein	+43 / 50 / 22110	press@fh-salzburg.ac.at
Games Academy Berlin	key player	Rungestraße 20	10179 Berlin	030 / 29779120	info@games-academy.de
Games Academy Frankfurt	key player	Hanauer Landstraße 146	60314 Frankfurt	069 / 42696460	info@games-academy.de
Gesellschaft für Personalentwicklung und Bildung		Beuthstraße 7-8	10117 Berlin	030 / 9339480	gpb-berlin@gpb-berlin.de
Hochschule Harz		Friedrichstraße 57-59	38855 Wernigerode	03943 / 659100	dackermann@hs-harz.de
Hochschule Mittweida (FH)		Technikumplatz 17	09648 Mittweida		kontakt@hs-mittweida.de
HTW Berlin	key player	Wilhelminenhofstraße 75a	12459 Berlin	030 / 50190	
Macromedia Akademie		Eisenheimerstraße 63	80687 München	089 / 96160800	ausbildung@macromedia.de

EDUCATION					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Macromedia Akademie		Naststraße 11	70376 Stuttgart	0711 / 2807380	info.stgt@macromedia.de
Macromedia Akademie		Brüderstraße 17	50667 Köln	0221 / 3108223	akademie.koeln@macromedia.de
Macromedia Hochschule		Gollierstraße 4	80339 München	089 / 5441510	info.muc@macromedia.de
media Akademie - Hochschule Stuttgart		Tübinger Straße 12-16	70178 Stuttgart	0711 / 925430	
Mediadesign Hochschule	key play	Lindenstraße 20-25	10969 Berlin	030 / 3992660	info-ber@mediadesign-fh.de
Mediadesign Hochschule		Werdener Straße 4	40227 Düsseldorf	0211 / 1793930	info-dus@mediadesign-fh.de
Mediadesign Hochschule		Claudius-Keller-Straße 7	81669 München	089 / 4506050	info-muc@mediadesign-fh.de
S4G School for Games		Gubener Straße 47	10243 Berlin	030 / 96595244	info@school4games.net
SAE Institute Berlin	key play	Soltauer Straße 18-22	13509 Berlin	030 / 43094470	berlin@sae.edu
SAE Institute Bochum	key play	Metzstraße 23	44793 Bochum	0234 / 93451310	bochum@sae.edu
SAE Institute Frankfurt	key play	Homburger Landstraße 182	60435 Frankfurt am Main	069 / 543262	frankfurt@sae.edu
SAE Institute Hamburg	key play	Feldstraße 66	20359 Hamburg	040 / 23688080	hamburg@sae.edu
SAE Institute Köln	key play	Medienzentr. Ost, Carlswerkstr. 11c	51063 Köln	0221 / 9541220	koeln@sae.edu
SAE Institute Leipzig	key play	Dittrichring 10	40109 Leipzig	0341 / 3085160	leipzig@sae.edu
SAE Institute München	key play	Bayerwaldstraße 43	81737 München	089 / 89068771	muenchen.sae.edu
SAE Institute Stuttgart	key play	Stuttgarter Straße 23	70469 Stuttgart	0711 / 81473690	stuttgart@sae.edu
SRH Fachschulen		Bonhoefferstraße 1	69123 Heidelberg	06221 / 884225	it.heidelberg@fachschulen.srh.de
SRH Hochschule Heidelberg		Ludwig-Guttmann-Straße 6	69123 Heidelberg	06221 / 882790	Daniel.Goerlich@Hochschule-Heidelberg.de
Zürcher Universität der Künste	key play	Ausstellungsstraße 60	CH-8005 Zürich	+41 / 43 / 4464646	hs.admin@zhdk.ch
BUSINESS SERVICE					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
4-REAL INTERMEDIA (Lokalisation)		Ludwigstraße 32	63067 Offenbach	069 / 8090880	info@4-real.com
adjust		Saarbrücker Strasse 36	10405 Berlin	030 / 91460083	
Anakan (Lokalisation)		Pfuelstraße 5	10997 Berlin	030 / 531420450	ana@anakan.de
Aruba Events		Kreuzstraße 1-3	45468 Mülheim an der Ruhr	0208 / 6982608	infos@aruba-events.de
arvato games task force		Carl-Bertelsmann-Straße 161F	33332 Gütersloh	05241 / 803074	kontakt@gamtaskforce.de
Brehm & v. Moers		Anna-Louisa-Karsch-Straße 2	10178 Berlin	030 / 2603050	berlin@bvm-law.de
Die Hobrechts (Game Thinking)		Hobrechtstraße 65	12047 Berlin	030 / 62901232	info@diehobrechts.de
Dr. Behrmann & Härtel Rechtsanwälte		Kantstraße 98	10627 Berlin	030 / 319984910	info@behrmannhaertel.de
Buschbaum Media & PR		Emanuel-Leutze Straße 21	40547 Düsseldorf	0211 / 5988140	presse@buschbaum-media.com
Clans.de (Network)		Pinnsberg 47	20359 Hamburg	040 / 78800990	info@clans.de
CULTURETRANSLATE		Lyoner Straße 34	60528 Frankfurt am Main	069 / 65998000	info@culturetranslate.de
DELASOCIAL		Christoph-Probst-Weg 31	20251 Hamburg	040 / 87979890	contact@delasocial.com
Digital River		Vogelsanger Straße 78	50823 Köln	0221 / 310 88614	sales@digitalriver.com
Effective Media		Josef-Haumann-Straße 10	44866 Bochum	02327 / 22310	info@effective-media.de
Frankfurter Buchmesse		Reineckstraße 3	60313 Frankfurt am Main	069 / 21020	info@book-fair.com
Freaks 4U Gaming		An der Spreeschanze 10	13599 Berlin	030 / 4171913	info@freaks4u.de
Friedmann Kommunikation		Schanzenstraße 36	51063 Köln	0221 / 5348710	info@fr-k.de
Gärtner PR		Destouchesstraße 68	80796 München	089 / 3076854	info@gaertner-pr.de
Game Developers Conference Europe	key play	Kaiser-Wilhelm-Straße 30	12247 Berlin	030 / 34622644	carsten.kohlenbeck@ubm.com
gamescom	key play	Messeplatz 1	50679 Köln	0221 / 8213894	info@koelnmesse.de
Games Foundation		Neuer Kamp 30	20357 Hamburg		info@gamesfoundation.com
Games Quality		Zum Wasserwerk 15	15537 Erkner	03362 / 885688	info@games-quality.com
Gamify Now!		Karlstraße 68	80335 München	089 / 210205715	info@gamify-now.de
GAMINSIDE		Kaiserstraße 2-4	1070 Wien	+43 / 0676 / 5379776	info@gaminside.com
GAN Game Ad Net		Stresemannstraße 342	22761 Hamburg	040 / 67586750	info@gan-online.com
Ganz & Stock (Personnel Consulting)		Westhafentower, Westhafenplatz	60327 Frankfurt am Main	069 / 710456380	a.stock@ganz-stock.de
GRAEF Rechtsanwälte		Jungfrauenthal 8	20149 Hamburg	040 / 80600090	www.graef.eu
HoneyTracks (Game Analytics)		Schelling Straße 35	80799 München	089 / 28723995	info@honeytracks.com
IDG Entertainment Media	key play	Lyonel-Feininger-Straße 26	80807 München	089 / 360860	dbhulapatra@idg.de
IEM Consulting (Business Development, Finanzierung)		Hanauer Landstraße 192	60314 Frankfurt am Main	069 / 15346479	info@iem-consulting.com
IHDE & Partner (Rechtsberatung)		Schoenhauser Allee 10-11	10119 Berlin	030 / 44318660	berlin@onlinelaw.de
INPROMO		Gasstraße 6a	22761 Hamburg	040 / 23881771	kontakt@inpromo.de
iVentureCapital		Wendenstraße 21 B	20099 Hamburg	040 / 8081250	info@iventurecapital.com
LaterPay (Zahlungsdienstleister)		Auenstraße 100	80469 München	089 / 416137319	info@laterpay.net
Lightning PR		Gundelstraße 13	60435 Frankfurt		koetting@lightning-pr.de
Marchsreiter Communications		Guldeinstraße 41a	80339 München	089 / 51919942	dm@marchsreiter.com
Medienachse (Full-Service, Businessplanung)		Trappentreustraße 10	80339 München		info@medienachse.de
M.I.T. - Media Info Transfer		Albert-Einstein-Ring 8	22761 Hamburg	040 / 30066890	info@mediainfotransfer.de
MOTION AREA		Sommerstraße 3	65197 Wiesbaden	0611 / 56597577	info@motionarea.de
Osborne Clarke (Anwaltskanzlei)		Innere Kanalstraße 15	50823 Köln	0221 / 51084000	info@osborneclarke.de
Partnertrans (Lokalisation)		Leipziger Straße 28	40822 Mettmann	02104 / 172660	info@partnertrans.com
paysafecard.com Wertmarken		Am Europlatz 2	1120 Wien	+43 / 1 / 72083800	sales@paysafecard.com
PIXABLE STUDIOS		Buchenstraße 16b	01097 Dresden	0351 / 56341360	info@pixable.de
Pixelworkshop / IG Computergrafik		Hirschengasse 1/26	1060 Wien	+43 / 1 / 5880118698	info@pixelworkshop.at
Quinke Networks (PR, Marketing)		Bei den Mühnen 70	20457 Hamburg	040 / 43093949	mail@quinke.com
remote control productions (Producing)		Sendlinger-Tor-Platz 6	80336 München	089 / 210205700	info@r-control.de
Rode + Mathé (Rechtsberatung)		Henrietteweg 4	20259 Hamburg	040 / 43270432	info@rode-mathe.de
ROESSLER PR (Marketing, PR, Consulting)		Walter-Leisze-Straße 2	60320 Frankfurt am Main	069 / 514461	communicate@roesslerpr.de
S&H Entertainment Localization (Lokalisation)		Weidenstraße 10a	82110 Germering	089 / 80076290	stefan@shentloc.com
SCC (Crossmedia)		Orleansstraße 5a	81669 München	089 / 33094660	info@scc-feld.de
Schanz International (Consulting)		Bgm.-Oberhettinger-Straße 2a	69146 Deidesheim	06326 / 6010	info@schanzgames.com
Schulte Riesenkampff (Rechtsberatung)		An der Hauptwache 7	60313 Frankfurt am Main	069 / 900266	schulte@schulte-lawyers.de
Selected Minds (Personalberatung)		Walther-Rathenau-Straße 16	64521 Groß-Gerau	06152 / 1871830	info@selected-minds.de
Serious Games Conference		Expo Plaza 1	30539 Hannover	0511 / 1234560	sgc@nordmedia.de
Sputnic Consulting		Ehrenbergerstr. 11	98693 Ilmenau	03677 7996990	media@sputnic-consulting.com
swordfish PR (PR, Marketing)		Habsburgerplatz 2	80801 München	089 / 96160840	info@swordfish-pr.de
Translation-Taxi		Beusselstraße 28	10553 Berlin	030 / 28371428	info@translation-taxi.com
Turtle Entertainment (eSport)		Siegburger Straße 189	50679 Köln	0221 / 8804490	info@turtle-entertainment.de
Two Pi Team		Gerrseheimer Straße 9	40721 Hilden	02103 / 9411914	info@two-pi-team.de
Visibility Communications (PR, Text)	key play	Wichterstraße 16/17	10439 Berlin	030 / 58859941	hi@visi.bi
waza!		Sonnenallee 70	12045 Berlin	0178 / 5223697	jannot@wazaservices.de

FREELANCER BUSINESS SERVICE					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Ahmet Isciturk		Kernstraße 37	90429 Nürnberg	0911 / 27472885	texte@me.com
Anja Weilmann		Kramers Kamp 1a	48291 Telgte	02504 / 7399601	kontakt@spielworte.de
Anne-Petra Lellwitz (Marktforschung)		Fichtestrasse 65	63303 Dreieich	0160 / 2014364	info@apl-mafo.de
Arwed-Ralf Grenzbach (Consulting)		Friedrich-Ebert Anlage 18	60325 Frankfurt am Main	0171 / 6486720	arg@conflutainment.com
Christiane Gehrke		Friedrich-Wilhelm-Straße 15	12103 Berlin	030 / 72010080	cg@tom-putzki-consulting.com
Elisabetta Corapi				0176 / 99763437	info@corapi-translations.com
Falko Löffler (Autor)		Lindenstraße 8a	36355 Grebenhain-Ilbeshausen	06643 / 918577	mail@falkoeffler.de
Kerstin Fricke		Wissmannstraße 9	12049 Berlin	030 / 20059246	info@kf-uebersetzungen.de
Marc Huppke (Projektmanagement)		Frankfurter Straße 5	64521 Gross-Gerau	06152 / 9614432	marc@make-projects.com
Marc Oberhäuser (Consulting)		Berzbuirer Strasse 98	52355 Düren	02421 / 51547	marc.oberhauser@freelancer-games.com
Stefan Köhler (Lektor / Autor)		Am Salgenteich 34	38259 Salzgitter	0176 / 22512273	stefanclenmskoehler@web.de
TECHNICAL SERVICE					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Akamai Technologies		Betastraße 10B	85774 Unterföhring	089 / 94006308	contact-germany@akamai.com
Augenpulver (Grafik)		Dotzheimerstraße 164, 4.0G	65197 Wiesbaden	0611 / 94574340	info@augenpulver-design.de
Augmented Minds		Fasangartenstraße 134	81549 München	089 / 69386904	info@augmented-minds.com
BiteTheBytes (Middleware)		St. Gallerer Straße 9	36039 Fulda	0661 / 2006899	info@bitethebytes.com
B.TON Medien (Audio, Musik)		Luise-Ullrich-Straße 4	82031 Grünwald	089 / 189425490	mix@bton.de
Centroid (Motion Capturing)		Schlesische Straße 27	10997 Berlin	030 / 69807474	sven.bergmann@centroidanimation.com
Chromatrix (Mobile Gaming)		Kalkofenstraße 2	72411 Bodelshausen	07471 / 740828	info@chromatrix.com
Codecult (3D-Engine)		Kemnastraße 21a	44866 Bochum	02327 / 35474	info@codecult.com
Contentainer (Leveldesign)		Eichelkopfstraße 17	63584 Gründau	06058 / 9178150	info@contentainer.de
Creature Factory		Graefestraße 33	10967 Berlin	030 / 26301330	info@creature-factory.com
Crytek (3D-Engine)		Grüneburgweg 16-18	60322 Frankfurt am Main	069 / 21977660	cryengine@crytek.com
Doublesmith (Art/Animation)		Ranzonigasse 3/3	A-3100 St. Pölten	+43 / 699 / 19447880	mail@doublesmith.com
Dutyfarm		Oranienstraße 6	10997 Berlin	030 / 30368430	info@dutyfarm.com
Dynamedion (Sounddesign)		Barbarossaring 8	55118 Mainz	06131 / 5847895	info@dynamedion.com
eurosimtec		Merowingerplatz 1	40225 Düsseldorf	0211 / 3018560	info@eurosimtec.de
Exit Games (Network Engine)		Hongkongstraße 7	20457 Hamburg	040 / 4135960	business@exitgames.com
Eye Rock Media (Design, Animation)		Essenerstraße 3	46047 Oberhausen	0208 / 82858923	contact@eyerock.de
fadeout (Audio Design)		Austraße 6	CH-4153 Reinach	+41 / 7 / 92793244	info@fadeout.ch
fatfoogoo		Mariahilferstraße 50	A-1070 Wien	+43 / 1 / 23622970	office@fatfoogoo.com
Flow Studios		Gustav-Meyer-Allee 25	13355 Berlin	030 / 80929203	info@flow-studios.eu
Games Quality (Quality Assurance)		Zum Wasserwerk 15	15537 Erkner	03362 / 885688	info@games-quality.com
Global Producer (Production)		Hauptstraße 172	51143 Köln	02203 / 97722400	getintouch@global-producer.com
Glare Studios (Grafik)		Mansteinstraße 18	20253 Hamburg	02307 / 4388404	contact@glarestudios.de
Goal Games		Feurigstraße 54	10827 Berlin	030 / 78957102	info@goal-games.de
GreenMamba-Studios (Animation)		Aloysstraße 7	48249 Dülmen	02594 / 7928217	info@greenmamba-studios.de
Hans HiScore (Audio)		Friedrich-Ebert-Anlage 11-13	60327 Frankfurt	0151 / 11616133	info@hans-hiscore.de
Havok		Arbachtalstraße 6	72800 Eningen	0712 / 1986993	info@havok.com
Intulo (3D, Animation)		Lindholz 89	31139 Hildesheim	05121 / 2944022	info@intulo.de
Keuthen		Robert-Koch-Straße 41	55129 Mainz	06131 / 880890	info@keuthen.net
Konsole Labs		Gritznerstraße 42	12163 Berlin	030 / 23634838	info@konsole-labs.com
Lingoona (Lokalisations-Middleware)		Hildastraße 38	68723 Plankstadt	06202 / 4095030	info@lingoona.com
Living Liquid Software		Langer Weg 15	A-6020 Innsbruck	+43 / 0 / 512263535	info@livingliquid.com
metricminds (Grafik)		Rüsselsheimer Straße 22	60326 Frankfurt am Main	069 / 7593380	info@metricminds.com
MobileBits (Mobile Engine)		Lerchenstraße 28	22767 Hamburg	040 / 33429566	info@mobilebits.de
morro images (Grafik)		August-Bebel-Straße 27	14482 Potsdam-Babelsberg	0331 / 97996610	info@morroimages.com
Nevigo (Tools & Middleware)		Lyrenstraße 13	44866 Bochum	02327 / 8369840	info@nevigo.com
Periscope Studio (Audio)		Am Diebsteich 55	22761 Hamburg	040 / 31811767	info@periscopestudio.de
Phenomatics (XNA-Entwicklung, Consulting)		Hafenstrasse 47-51	A-4020 Linz/Donau	+43 / 732 / 90155230	office@phenomatics.com
Polyce (Grafik)		Littenstraße 106/107	10179 Berlin	0178 / 3592560	info@polyce.de
Polygonfabrik (Grafik)		Friesenstraße 21	28203 Bremen		kontakt@polygonfabrik.de
Project-C (HYDRA)		Dorfstraße 50	29336 Nienhagen	05144 / 6988871	kontakt@project-c.eu
Rabcat Computer Graphics (Grafik)		Fernkornegasse 10	A-1100 Wien	+43 / 1 / 5237425	mail@rabcat.com
Razer (Hardware)		Winterhuder Weg 82	22085 Hamburg	040 / 41929300	
Rocketbox Studios (Grafik)		Leonhardtstraße 10	30175 Hannover	0511 / 8984384	info@rocketbox.de
Spinor (3D-Engine)		Agnes-Pockels-Bogen 1	80992 München	089 / 54043980	spinor@spinor.com
The Light Works (Grafik)		Otto-Hahn-Straße 7	50997 Köln	02236 / 967322	info@thelightworks.com
ToBringAlive (Animation, Design)		Ostmarkstraße 49	48145 Münster	0251 / 392312	angerbauer@tobringalive.com
Tektroop (Support)		Dechenstraße 8	40699 Erkrath		info@tektroop.ch
TON & SPOT audiodesign (Sound)		Immanuelkirchstraße 15	10405 Berlin	030 / 92129727	post@ton-und-spot.de
Toygardens Media (Grafik, Video)		Walderseestraße 54	30163 Hannover	0511 / 33659800	akunze@toygardens.com
VCC Perfect Pictures (Grafik, Video)		Doormannsweg 43	20259 Hamburg	040 / 431690	hamburg@vcc.de
Versant (Data Management)		Wiesenkamp 22b	22359 Hamburg	040 / 609900	info@versant.com
Virgin Lands (Grafik, Cinematics)		Gneisenaustraße 10/11	97074 Würzburg	0931 / 8049000	info@virgin-lands.com
Z-Ground Illustration (Grafik)		Oeltzenstraße 17	30169 Hannover	0511 / 1696959	jz@z-ground.com
zuuka! (Audio, Lokalisation)		Christian-Pless-Straße 11-13	63069 Offenbach am Main	069 / 2475700	mail@zuuka.de
FREELANCER TECHNICAL SERVICE					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Alexander N. Ostermann (Animator)		Breite Strasse 45	14199 Berlin	0176 / 10365759	nico@nico-ostermann.com
Andreas Adler (Audio)		Wernigeroder Weg 18	30419 Hannover	0511 / 3889163	andreas@adler-audiopictures.de
Calined (Audio)		Lange Straße 22	58089 Hagen	01577 / 9508456	business@calined.com
Daniel Pharos (Audio)		Golddistelanger 15	80937 München	0179 / 5210100	daniel@knightsofsoundtrack.com
Johan Weigel (Musik / Ton / Sound)		Christburger Straße 28	10405 Berlin	0176 / 4146494	johan@sonic-gallery.com
Sound42 / Lukas Hasitschka		Kulmgasse 38/8	1170 Wien	+43 / 6641059157	lukas@sound42.com
Max Schulz					info@xoco1.com
Olaf Bartsch (Sounddesign)		Sültstraße 62	10409 Berlin	0170 / 7743432	info@olafbartsch.com
Sabrina Heuer (Audio Production)		Bachemer Straße 191	50935 Köln	0163 / 8604004	sabrina@froschtatze.de

Ein Tag bei ...

OFM



Fernab von Hektik und Stress versteckt sich das Büro der OFM GmbH in einem lauschigen Hinterhof.



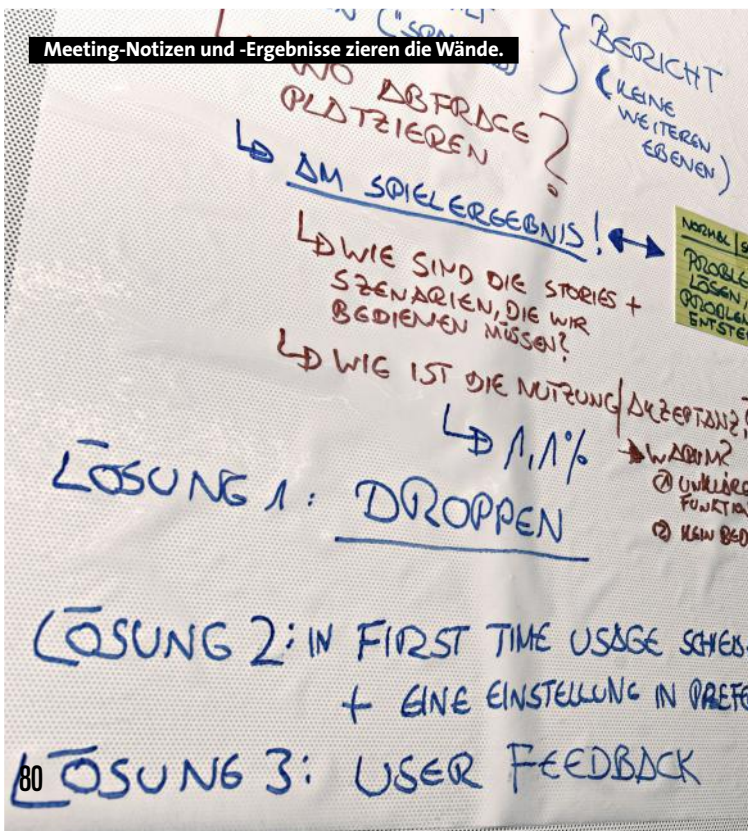
Das Büro ist in einer alten Druckerfabrik untergebracht und versprüht echten Industriecharme.



Warp-Antrieb à la OFM. Manchmal braucht man einfach einen zusätzlichen Energieschub.



Auch die entspannten Momente werden hier mit Sorgfalt vorbereitet. Im Sommer wird oftmals gemütlich im Innenhof gegrillt und gefeiert.



Lead Game Designer Guido Schmidt zaubert gerade mit Leidenschaft das nächste Produkt.



Halbhohe Mauern unterteilen das Großraumbüro in freundliche Arbeitsbereiche.

Bereits seit 2003 läuft das Browser-Spiel OnlineFußballManager erfolgreich und wurde mit der Zeit für die unterschiedlichsten Länder lokalisiert. Hier zeigt das Team aus Köln die Räumlichkeiten, in denen die neuen Inhalte für den OFM und ein weiteres bisher unveröffentlichtes Produkt entstehen.



Der Chef persönlich rückt die noch freien Stühle zurecht. Für das neue Mobile-Projekt sucht die OFM GmbH noch Mitarbeiter, zum Beispiel Unity-3D-Entwickler und einen 3D Artist.



Jedem Mitarbeiter stehen ein Laptop und zwei große Monitore zur Verfügung.



Gutes Wetter bringt gute Ideen! Der Innenhof lädt zu kreativer Entfaltung ein.



Bei den Daily Stand-up Meetings wird ein Fußball als Rede-Token benutzt.



Gute Laune und ein lockeres Teamgefühl sind bei der OFM GmbH wichtig. Nur wer gerne zur Arbeit kommt, bringt auch volle Leistung.



Wer eine digitale Auszeit braucht, kann sich im Aufenthaltsraum außerdem bei einer Runde Kickern entspannen.



Fußball, Fußball, Fußball! Es wird auch gerne mal zusammen gekickt. Dann allerdings eher in der Fußballhalle.



Der Konferenzbereich bietet mit Beamer, Leinwand und genügend Sitzen ausreichend Platz, um alle Mitarbeiter eines Teams an einen Tisch zu bekommen.



Vier verschiedene Arten von Führungsstil: El Presidente, Heisenberg, Stromberg und unser Chef Maik.



#läuftbeidir.

MAKING GAMES VORSCHAU

05/2015
ERSCHEINT AM 03. AUGUST 2015
MAKING GAMES

Making of World of Warships

Nach dem Überraschungserfolg »World of Tanks« wagt sich Wargaming an die nächste große Kriegsmaschine: »World of Warships«. Während sich der Multiplayer-Titel noch in der Beta-Phase befindet, gewähren die Entwickler Einblicke, wie sie Cross-Plattform-Partien zwischen PC und Konsolen ermöglichen, wie sie historische Genauigkeit und Spielspaß unter einen Hut bringen, wie sie beeindruckende 3D-Schlachten auf Mobilgeräte umsetzen und wie viel Arbeit in einer aktiven Community steckt.

Weitere Themen

Entwickler-Tools Gameworks

Nvidia gibt Einblicke, wie die hauseigenen Gameworks-Werkzeuge funktionieren und welche Möglichkeiten sie für Entwickler bieten.

Musikkomposition in Videospielen

Komponist Russell Brower von Blizzard erklärt, wie Musik und Leitmotive für Spiele wie »World of Warcraft« oder »Diablo 3« entstehen.

Deus Ex: Mankind Divided

Eidos Montreal gewährt Einblicke in seine hauseigene Dawn-Engine und zeigt, wie man unter anderem eine realistische Haarsimulation umsetzt.

IMPRESSUM

Chefredakteur
Heiko Klinge
(verantwortlich,
Anschrift der Red.)

Managing Editor
Sebastian Weber

Redaktion
Yassin Chakhchoukh
Patricia Gelger
Markus Schwerdtel

Layout & Design

Sigrun Rüb (ltd.)
Manfred Aumaier
Alexander Wagner
Eva Zechmeister
Anita Blockinger
(Freelancer)

Lektorat (Freelancer)

Marion Schneider

Freie Mitarbeiter in dieser Ausgabe (Redaktion)

Marco Apel, André Bernhardt, Thomas Bideaux, Benjamin Bischoff, Nurdogan Erdem, Sam Faulkner, Christian Feldbacher, Sergey Galyonkin, Hans Hamm, Gunnar Lott, Jakob Munthe, Stoyan Nikolov, Arne Öhsen, Chris Polus, Guido Schmidt, Daniel Wilkins, Oleg Yavorsky, Salomon Zwecker

Wir danken den Interview-Partnern:

Dominik Abé, Angelika Achter, Yancey Strickler

Assistenz: Isa Stamp, Anita Thiel, Annie Weissenberger

KONTAKT

Anzeigenanfragen:
Dustin Reichl, 089 / 360 86-671
dreichl@idg.de

Artikel- & Themenvorschläge:
editor@makinggames.de

Recruiting- & Projektanfragen:
projects@makinggames.de

Adressänderungen Abo:
kundenservice@makinggames.de

Änderungen für Firmenregister:
info@makinggames.de
Betreff: Firmenregister

Digitale Anzeigenannahme Print:
Manfred Aumaier, 089 / 360 86-602
anzeigendisoprint@gamestar.de

Digitale Anzeigenannahme Online:
banner@idg.de

Anzeigenpreise:
www.idg-eas.de

HAFTUNG & CO.

© Copyright IDG Entertainment Media GmbH

Haftung: Für die Richtigkeit von Veröffentlichungen können Redaktion und Verlag trotz Prüfung nicht haften. Die Veröffentlichungen in Making Games erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Auch werden Warennamen ohne Gewährleistung einer freien Anwendung benutzt.

KUNDENSERVICE

Abonnement und Einzelbestellung:
Making Games Kundenservice
ZENIT Pressevertrieb GmbH
Postfach 81 05 80, 70522 Stuttgart
Tel.: 0711 / 7252-275
Fax: 0711 / 7252-377
Österreich: Tel.: 01 / 219 55 60
Schweiz: Tel.: 071 / 314 06-15
E-Mail: kundenservice@makinggames.de
Web: www.makinggames.biz/shop

Bezugspreise:
Einzelpreis: 6,90 Euro
Jahresabo: 35,40 Euro
(beinhaltet 6 Ausgaben)
Auslandspreise u. Mengenrabatt auf Anfrage

Zahlungsmöglichkeit:
Postbank Stuttgart, BLZ 600 100 70
Konto-Nr.: 311 704

So erreichen Sie die Redaktion:
IDG Entertainment Media GmbH
Redaktion Making Games
Lyonel-Feininger-Straße 26
80807 München
Tel.: 089 / 360 86-660, Fax: 089 / 360 86-652

VERTRIEB HANDELSAUFLAGE

MZV GmbH & Co. KG
Ohmstraße 1, 85716 Unterschleißheim
Telefon: 089 / 319 06-0, Fax: -113
E-Mail: info@mzv.de
Web: www.mzv.de

VERLAG

IDG Entertainment Media GmbH
Lyonel-Feininger-Straße 26
80807 München
Tel.: 089 / 360 86-0, Fax: 089 / 360 86-501
www.idgmedia.de

Gründer: Patrick J. McGovern (1937 - 2014)
Geschäftsführer: York von Heimbürg
Verlagsleitung: Frank Maier
Director Sales: Ralf Sattelberger (-730)
Leiter CRM/Marketing: Matthias Weber (-154)

Veröffentlichung gemäß §8, Absatz 3 des Gesetzes über die Presse vom 08.10.1949: Alleiniger Gesellschafter der IDG Entertainment Media GmbH ist die IDG Communications Media AG, München, die 100% Tochter der International Data Group Inc., Boston, USA, ist.

Vorstand:
York von Heimbürg, Keith Arnot, David Hill

Produktionsleitung: Jutta Eckbrecht

Druck und Beilagen:
SDV Direct World GmbH
01159 Dresden



ANNA

Junior VFX Artist



FLORIAN

Principal VFX Artist

More than 130 people from
20 different countries
working together in Berlin,
right on the banks
of the Spree river.

Creating awesome games
since 1999.

Join Our Team!

**WWW.YAGER.DE
/CAREER**



GAMEFORCE

START YOUR CAREER,
**BE PART OF
THE TEAM!**

[CORPORATE.GAMEFORCE.COM/JOBS](https://corporate.gameforce.com/jobs)