

making games

DESIGN | BUSINESS | ART | TECHNOLOGY



CONQUERING NEXT GEN



OPEN WORLD MIT UNITY
WIE MAN RIESIGE SPIELWELTEN
AUF MOBILGERÄTEN ERSCHAFFT

SERVER BACKENDS
WELCHE ARCHITEKTUR WELCHE VOR-
UND NACHTEILE MIT SICH BRINGT

MOBILE BUSINESS
WIE KOSTENLOSE SPIELE-APPS AUF
AMAZON DEN UMSATZ BEFLÜGELN

CELEBRATING

GDC

30TH EDITION

GAME DEVELOPERS CONFERENCE®

Moscone Center · San Francisco, CA

March 14–18, 2016 · Expo: March 16–18, 2016

GDCONF.COM

#GDC16

Save up to \$300 before March 9, 2016



Readers, use code **GDC16MG** to save an extra 10% on All Access & Main Conference Passes to GDC 2016!



NEXT-GEN-KONSOLEN

EINE BEQUEME SACHE FÜR KUNDEN, ABER AUCH ENTWICKLER?

Konsolen hatten gegenüber dem PC schon immer einen ganz entscheidenden Vorteil als Spieleplattform für die Kunden: den Bequemlichkeitsfaktor. Die Geräte waren in der Regel günstiger als ein vergleichbarer Computer, für den man on top einen Bildschirm, Maus und Tastatur sowie Software braucht, während die kleinen Kisten fürs Wohnzimmer einfach an den sowieso vorhandenen Fernseher angeschlossen werden (die sogenannten Early Adopter bei der Preisfrage mal ausgeschlossen). Einmal gekauft, sind die Konsolen dann für die nächsten fünf oder mehr Jahre gesetzt – kein Aufrüsten und ähnliche Späße, die ambitionierte PCler alle paar Monate über ihren Geldbeutel ergehen lassen. Zugegeben, dieser Faktor hat auch mich als ursprünglich eingefleischten PC-Spieler über die Jahre ins Lager von PlayStation 4 und Xbox One getrieben, und dieses umfasst, zumindest laut EAs CFO Blake Jorgensen, rund 55 Millionen Spieler. 36 Millionen von ihnen halten zwar Sony die Stange, aber Microsoft kann immerhin auf 19 Millionen Nutzer blicken – auch wenn diese Angabe von EA und nicht von Microsoft selbst stammt. Das bedeutet zwar, dass der PC rein von den Zahlen her noch immer größer ist (Steam hat 125 Millionen Nutzer), doch Konsolen sind – das wird keiner bezweifeln – ein immens wichtiger Markt.

So einfach wie nie

Die gute Nachricht dabei: Seit der Einführung der aktuellen Konsolengeneration wurden auch für die Entwickler die Einstiegshürden immens gesenkt. Während sich PlayStation und Xbox früher enorm voneinander unterschieden, sind die Geräte heute vergleichbar. Während es für kleinere Entwickler und Publisher früher quasi unmöglich war, ihre Titel in eins der Ökosystem zu veröffentlichen, bieten die Hersteller heute Indie-Programme. Während früher utopischer Aufwand betrieben werden musste, um Spiele auf mehreren Plattformen zu veröffentlichen, gibt es heute Engines, die einem diese Arbeit mehr oder weniger abnehmen. Wie ich dazu komme, das einfach zu behaupten? Nun, dafür lege ich Ihnen die Erfahrungsberichte unserer Experten ans Herz, wo KING Art zum Beispiel erklärt, wie man dank Unity Engine für verschiedene Geräte gleichzeitig arbeiten kann (S. 30), wo Realmforge zeigt, wie man ein klassisches PC-Spiel auf die Konsolen portiert (S. 24) oder wo das Indie-Team Juicy Beast darauf zurückblickt, welche Fehler man bei einem Konsolen-Release machen kann (S. 38). Nach der Lektüre werden Sie mir sicherlich zustimmen, dass die Veröffentlichung von Konsolenspielen noch nie so einfach war wie heute.

Viel Spaß mit der aktuellen Ausgabe!

das Team von Making Games

Sebastian Weber
Ist Managing Editor
vom Making Games Magazin.



»Die Veröffentlichung von Konsolenspielen war noch nie so einfach wie heute.«

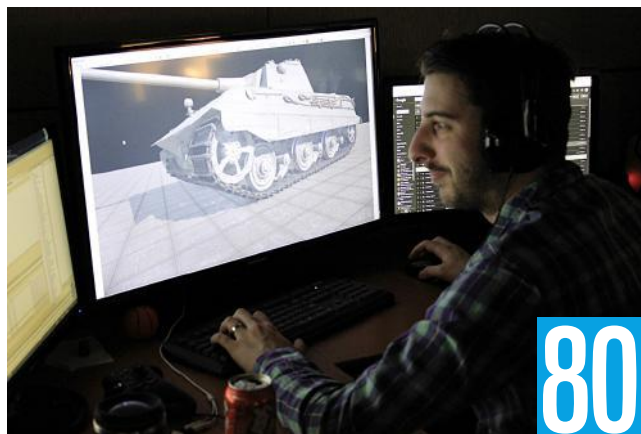


INHALT

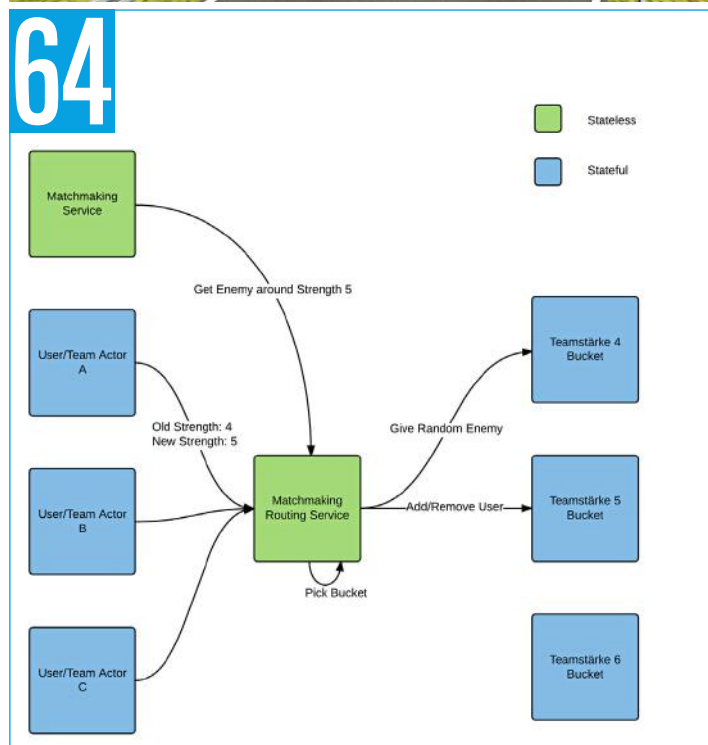
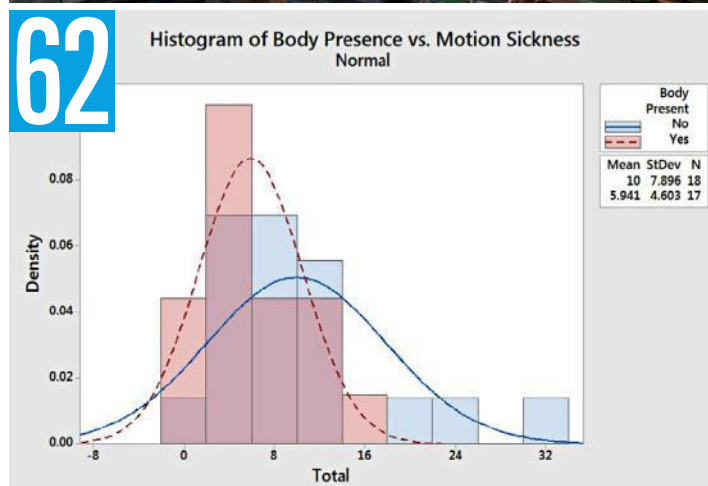
02/2016

- 03** Editorial
- 80** Ein Tag bei ...
Wargaming Chicago-Baltimore
- 82** Vorschau / Impressum

- 06** Branche / Köpfe
Bemerkenswertes aus der Branche
- 07** Branche / Köpfe
BIU Academy
- 08** Fokus G.A.M.E.
von Timm Walter
- 09** Eventkalender
Wichtige Branchenveranstaltungen
- 10** Passiert auf makinggames.biz
Interessantes aus dem World Wide Web
- 12** Making Games Tools
Die Meinung der Profis zu aktueller Soft- und Hardware
- 13** Marktforschung DataFlow
GameStar-/GamePro-Leserdaten
von Patricia Geiger



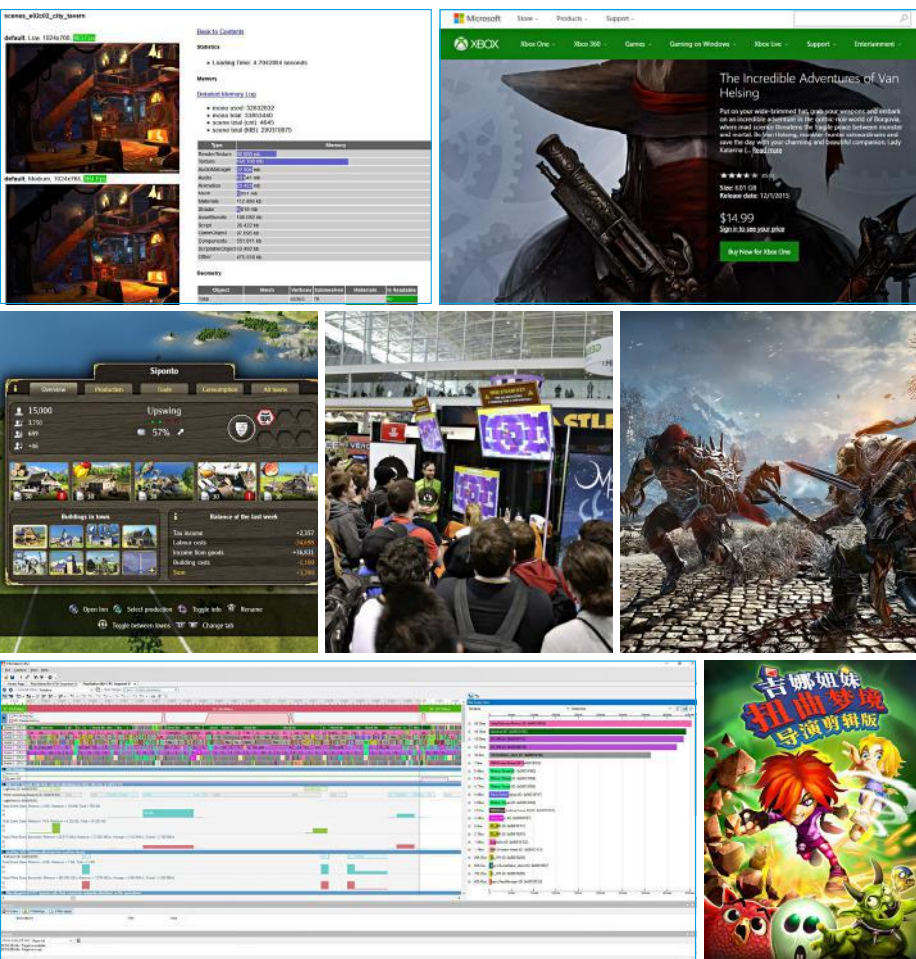
80



TITELTHEMA

Conquering Next-Gen

- 14 Case Study**
How Next-Gen Consoles are Changing Developers' Opportunities
von Jan Klose
- 18 Case Study**
Porting Van Helsing to Xbox One
von Balázs Farkas
- 22 Best Practice**
Erfahrungen bei der Portierung von PC-Spielen auf die PlayStation 4
von Daniel Dumont
- 24 Case Study**
Technische Herausforderungen einer PlayStation-Portierung
von Christian Schlütter
- 28 Post Mortem**
Als Indie auf die großen Konsolen
von Mischa Strecker
- 30 Best Practice**
Multiplattformentwicklung mit Unity
von Arne von Öhsen und Jan Theysen
- 38 Post Mortem**
Releasing an Indie Game on 3 Consoles at Once and Failing Financially
von Yowan Langlais



- 46 Best Practice**
Actually Free on Amazon Underground – Beyond Freemium and In-App-Purchases
von Carlos Lievano
- 52 Best Practice**
Localising Comedy
von Mark Estdale
- 54 Game Design Case Study**
7 Wege, um den Spieler bei der Stange zu halten
von Alexander Karl
- 57 Case Study**
Open World auf Mobile mit Unity
von Christoph Ender
- 62 Scientific Case Study**
Body Parts and Motion Sickness in VR – Mitigating Motion Sickness with Legs
von Dr. Matthew White
- 64 Best Practice**
Frei skalierbare Backend-Architektur mit Hilfe von Actor Model und Service Fabric
von Andreas Pohl und Kevin Daniel Setiono
- 74 Firmenregister**
Die Branche im Überblick



GOODGAME-MITARBEITER SPRECHEN SICH GEGEN BETRIEBSRAT AUS

Seit Ende November 2015 war der Hamburger Free2Play-Gigant Goodgame Studios ein fester Bestandteil der Berichterstattung, nicht nur von Medien mit Spielebezug sondern auch von Publikationen wie Spiegel Online oder verschiedenen Tageszeitungen. Grund: Das Unternehmen hatte 28 Mitarbeiter entlassen, von denen 15 die Gründung eines Betriebsrats vorbereitet haben sollen. Es stand die Vermutung im Raum, dass die Geschäftsführung dies mit den Entlassungen verhindern wollte, die Gewerkschaft Ver.di schaltete sich ein. Ende Januar kam es schließlich zu einer Abstimmung während einer Betriebsversammlung, bei der sich 62,8 Prozent der anwesenden 1.035 Mitarbeiter gegen die Gründung eines Betriebsrats aussprachen. Stattdessen werde eine interne Mitarbeitervertretung ins Leben gerufen. Goodgame-CEO Kai Wawrzinek freute sich über die Entscheidung seiner Mitarbeiter, auf eine individuell auf das Unternehmen abgestimmte Lösung zu setzen, Ver.di indes akzeptierte das Wahlergebnis.

Offen bleibt allerdings weiterhin die Frage, ob das Management von Goodgame Studios die zuvor entlassenen Mitarbeiter tatsächlich daran hindern wollte, einen

Betriebsrat zu gründen oder ob die Aussage stimmt, dass betriebliche Gründe ausschlaggebend gewesen seien – darüber werden Gerichte entscheiden müssen. Während Ver.di aber zum Beispiel das Ergebnis der Abstimmung akzeptierte, gingen die Diskussionen in der Branche weiter und viele vermuteten, dass die Mitarbeiter eingeschüchtert worden sein könnten. Doch könnten tatsächlich rund 650 Mitarbeiter, die mit »Nein« votierten, alle Angst vor der Befürwortung eines Betriebsrats haben, gerade wenn sich eine Gewerkschaft schon in den Waden des Managements verbissen hat? Würde Goodgame tatsächlich über der Hälfte der Belegschaft mit Entlassung drohen? Schwer anzunehmen und zumindest in öffentlichen Äußerungen der Mitarbeitern der Firma klingt es so, als ob sie mit dem Ausgang des Votums tatsächlich zufrieden wären. Ob das nur gute Miene zum bösen Spiel oder die Wahrheit ist und ob das Abstimmungsergebnis nun die wirkliche Meinung der Mitarbeiter widerspiegelt oder nicht, das lässt sich nur schwer herausfinden. Aber es ändert nichts daran, dass die Goodgame-Belegschaft sich gegen einen Betriebsrat ausgesprochen hat und es sei ihr zu wünschen, dass es die richtige und eine freie Entscheidung gewesen ist.

Sebastian Weber



Kai Wawrzinek,
CEO der Goodgame Studios

WWW.makinggames.biz
■ aktuelle News und Gerüchte

GAMESTOP STEIGT INS PUBLISHING-GESCHÄFT EIN



Ende Januar hat Entwickler Insomniac Games (»Sunset Overdrive«) sein neues Spiel »Song of the Deep« angekündigt, das im zweiten Quartal 2016 für PC, PlayStation 4 und Xbox One erscheinen soll. Die große Überraschung der Ankündigung war jedoch, dass die Handelskette GameStop erstmals als Publisher agieren wird. Während die digitalen Versionen des Spiels von Insomniac Games selbst in die einzelnen Stores verteilt werden, wird die Disc-Fassung exklusiv nur bei GameStop erhältlich sein. Außerdem unterstützt der Händler die Entwickler in Marketing-Fragen und plant, verschie-

dene Merchandise-Artikel zu Song of the Deep in seinen Geschäften anzubieten. Abseits dessen hat GameStop aber auch Ted Price, dem Gründer von Insomniac Games, keinen Einfluss auf die Entwicklung und auch die Rechte am Spiel bleiben bei dem Studio. GameStop wiederum erklärte den Deal damit, dass das Unternehmen schon immer nach exklusiven Inhalten sucht, um seine Kundschaft in die Läden zu locken – üblicherweise Dinge wie DLCs oder besondere Sammlerversionen von AAA-Titeln – und diese Art der Publishing-Tätigkeit eine interessante Option sei.

NEUE TERMINE FÜR »BIU ACADEMY«-WEBINARE STEHEN FEST

Die Webinar-Reihe »BIU Academy«, die der Bundesverband Interaktive Unterhaltungssoftware in Kooperation mit Making Games im Januar gestartet hat, widmet sich als nächstes dem Thema Projekt Management.

BIU Academy« in Kooperation mit Making Games ist eine Webinar-Reihe für Entwickler, Gründer und Interessierte; eine Weiterbildungsmaßnahme ohne Reisekosten zu allen Themen rund um Spieleentwicklung und -vermarktung, zum Beispiel Business Modelle, Design, Management, Legal, Marketing und viele mehr.

- **Länge:** 60 Minuten
- **Monatlich; Dienstags** um 17:00 Uhr
- **Preis:** 19,90 EUR
(für Mitglieder des BIU-Netzwerks kostenfrei)
- **Aktuelle Informationen unter:**
www.biu-online.de
Facebook: BIUeV
Twitter: BIU_eV

Die dritte Webinar-Veranstaltung widmet sich dem Thema »Vor- und Nachteile agiler Projektmanagement-Methoden in der Video-spielentwicklung«. Das Webinar findet statt am **Dienstag, 08.03.2016 um 17:00 Uhr.**

In seinem Webinar gibt Referent Ralf C. Adam einen Einblick in verschiedene agile Projektmanagement-Methoden wie SCRUM, Lean oder XP-Programming und ihre Anwendung bei der Entwicklung von Video- und Computerspielen. Dabei beschäftigt er sich insbesondere mit potenziellen Fallstricken, Missverständnissen und Fehlinterpretationen, die ihm häufig im Zuge der Benutzung agiler Methoden begegnen. Außerdem zeigt Adam verschiedene Best-Practice-Beispiele wie den sinnvollen Einsatz von Kanban-Boards oder die Wichtigkeit einer sauberen Pre-Production auf.

Ralf C. Adam ist seit 1993 in der Spielebranche tätig und arbeitete unter anderem für Unternehmen wie Sunflowers (Project Lead), Travian (Head of Game Development), Phenomic (Project Lead), Infogrames (Executive Producer) und JoWooD (Executive Producer). Aktuell hat er die Position des Executive Producers für flaregames inne und führt seit 2004 seine eigene Consulting-Firma namens TigerTeam Productions.



Ronald Kaulbach
Studio Brand Director
bei Blue Byte.



Ralf Adam
Executive Producer
bei flaregames.



Sabiha Ghellal
Professorin
an der Hochschule der Medien
Stuttgart.



Lutz Anderie
Managing Director
Germany & Austria bei
Smartbox Group



Konstantin Ewald
Rechtsanwalt und Partner
bei Osborne Clarke.

Termine 2016

Termin	Thema	Referent
16. Februar	Brand Management	Ronald Kaulbach (BlueByte)
08. März	Methoden des Projekt Management	Ralf Adam (flaregames)
05. April	Mobile Game Design Workshop	Sabiha Ghellal (HDM Stuttgart)
Mai	Virtual Reality Development Insights	N.N. (Crytek)
Juni	Unternehmensfinanzierung	Lutz Anderie (Anderie Management)
Juli	Urheberrecht für Spieleentwickler	Konstantin Ewald (Osborne Clarke)

Änderungen vorbehalten

FOKUS GAME

DAS »DEUTSCHE SPIEL« – GAR NICHT MAL SO ANDERS?

Timm Walter über den sich wandelnden Charakter deutscher Spieleproduktionen und wieso diese heute auch international erfolgreich sein können.



Timm Walter
ist Leiter der Geschäftsstelle
des GAME Bundesverband der
deutschen Games-Branche e.V.

Timm Walter ist Politikwissenschaftler und kümmert sich als Leiter der Geschäftsstelle um das operative Geschäft des GAME. Vor seiner Zeit im Verband konnte er weitreichende Erfahrungen in der Politik und der Medienbranche sowie im Vereinswesen sammeln.

Das Jahr 2016 ist noch jung, und wir alle befinden uns in den Vorbereitungen für die nächsten Monate. Neben den Milestones einer Entwicklung stehen auch wieder einige wichtige Events vor der eigenen Haustür an. Den Anfang macht der Deutsche Computerspielpreis (DCP) am 07. April in München, gefolgt von der Quo Vadis und International Games Week in Berlin zwei Wochen später. Mitte August dominiert wieder die gamescom in Köln den Kalender und abgeschlossen wird das Jahr mit dem Deutschen Entwicklerpreis (DEP) in Köln im Dezember. Eines haben all diese Veranstaltungen gemeinsam: den Fokus Games. Aber nur beim DCP und DEP liegt dieser auf dem deutschen Spiel. Doch was genau ist das deutsche Spiel und kommt es in der Community auch so gut an wie internationale AAA-Produktionen?

Was charakterisiert deutsche Spiele?

Deutschland, das Land der Simulationen, Aufbaustrategie und klassischen Adventures – so zumindest die Theorie. Auch praktisch kommen »X-Plane«, ein »Anno« oder die »Deponia«-Reihe aus hiesigen Landen und sind vergleichsweise rentabel. Der deutsche Spieler kann sich also nach wie vor für »unsere« altbewährten Genres begeistern. Warum auch nicht? Wird hier doch seit Jahren hochwertige Qualität abgeliefert,

die mit gewohnten Werten überzeugt. Nicht umsonst wurde ein Adventure wie »The Inner World« (2014) zum Besten Deutschen Spiel beim DCP gewählt. Doch Deutschland kann heute mehr als das Bekannte und Deutschland kann mittlerweile vor allem international. »Lords of the Fallen« ist hier ein Beispiel, aber auch ein »Crysis«, »Ryse«, die »Risen«-Reihe, »Anno« oder »Angry Birds Epic« zeigen, dass Deutschland mit globalen Großproduktionen mithalten kann.

Das mag vor allem daran liegen, dass der Geschmack der Community gar nicht mehr so stark zwischen typisch deutsch und international unterscheidet. Bestes Spiel beim DCP 2015? »Lords of the Fallen«. Gewinner des Publikumspreises 2015? »Dark Souls 2«. Auch wenn die Jungs und Mädels bei Deck13 nie vor hatten, eine direkte »Dark Souls«-Konkurrenz zu entwickeln, so sind die Parallelen offensichtlich. Und noch etwas fällt auf: Deutsche Produktionen werden immer anscheinlicher und auch die Budgets wachsen, wenn auch nicht in dem Maße wie in Übersee. Trotzdem brauchen wir den internationalen Vergleich nicht zu scheuen. Nicht nur sind die Spielmechaniken eines »Crysis« innovativer als die von »Call of Duty«, nein, es sieht auch noch um Welten besser aus. Ein »Ryse« war Launch-Titel der Xbox One und sorgt auch heute noch für offene Münder, und die Grafikstile von »Lords of the Fallen«, »Anno« oder den Daedalic-Titeln sind unverwechselbar. Klassische Alleinstellungsmerkmale also.

Anpassung ohne Identitätsverlust

Doch wo führt uns diese Feststellung hin? Die Antwort ist ganz einfach: zurück an den Anfang. Ja, es gibt es noch, das typisch deutsche Spiel. Ja, es ist in den meisten Fällen auch sofort zu erkennen. Aber nein, es ist eben nicht mehr komplett anders im internationalen Vergleich. Deutsche Games-Entwicklung hat sich angepasst, ist internationaler geworden, ohne dabei die eigene Identität zu verlieren. Und sie hat nebenbei auch noch den globalen Markt für sich entdeckt. Eine Entwicklung, die mehr als beachtenswert ist. Gerade dann, wenn einem bewusst wird, welche Flut an Produkten auf dem Weltmarkt mit uns konkurriert. Bravo deutsche Games-Branche. Weiter so! **Timm Walter**

International erfolgreiche Produktionen wie »Anno«, »Lords of the Fallen« oder »Crysis 3« (im Bild) zeigen, dass deutsche Spiele sich nicht mehr zu verstecken brauchen.





GERMAN DEV DAYS ANGEKÜNDIGT

Nachdem Stefan Marcinek zum Jahreswechsel nach einem Management Buyout seine Tätigkeiten als Geschäftsführer der Kalypso Media GmbH niedergelegt hatte, meldete er sich Ende Januar mit der Ankündigung der »German DevDays« zurück. Die kostenlose Entwicklerkonferenz findet am 24. und 25. Mai 2016 in Frankfurt statt und soll sich ausschließlich auf die deutschsprachige Games-Branche fokussieren. Geplant sind rund 30 Vorträge, mehrere Podiumsdiskussionen sowie Streitesprache, für die mehr als ein Dutzend Entwickler zugesagt haben – unter anderem KING Art Games, HandyGames, Mimimi Productions und Studio Fitzbin. Abgerundet wird das Programm mit einer Ausstellungsfläche, wo Entwickler ihre Projekte vorstellen können sowie einem Barbecue am Abend. Eingeladen sind Branchenvertreter, Spieleentwickler, Studenten und branchenfremde Besucher, die sich über die Website der Veranstaltung registrieren können und dann später handverlesen ihr Ticket zugeteilt bekommen.

Vier Fragen an ...

Stefan Marcinek
Ist Veranstalter
der German DevDays.



Making Games Wie entstand die Idee für die German DevDays?

Stefan Marcinek Ich selber bin ja schon lange, seit den 90ern Teil der Games-Industrie. Erst als freier Redakteur, dann als Chefredakteur und später als PR-Manager. Nachdem ich Ende letzten Jahres meine Hälfte an der Kalypso Media GmbH, die ich im Sommer 2006 mitgegründet habe, verkauft habe, war es für mich nun Zeit etwas zurückzugeben und solch eine Veranstaltung auf eigene Kosten zu organisieren. Wichtig war mir dabei, dass diese erste Veranstaltung völlig kostenlos für alle ist.

Making Games Welche Zielsetzung hat die Konferenz und wieso richtest du sie explizit auf deutsche Entwickler aus?

Stefan Marcinek Es gibt zwar einige kleinere Veranstaltungen dieser Art, aber keine, die gleich über zwei Tage geht und sich dabei komplett an die deutsche Entwicklerszene und Community richtet. Die großen Veranstaltungen in diesem Bereich sind auch in Deutschland mit englischsprachigen Vorträgen international ausgerichtet. Eine rein deutsche Konferenz fehlt einfach. Und den Ansatz, dass auch normale Besucher, die nicht aus der Spieleindustrie kommen, gibt es so nicht.

Making Games Du peilst eine bunte Mischung an Besuchern an – was versprichst du dir von

dieser Zusammensetzung und was entscheidet am Ende darüber, wer teilnehmen darf?

Stefan Marcinek Wie schon erwähnt, gibt es eigentlich keine Veranstaltung, bei denen Gamer bzw. Besucher auf Entwickler treffen – vielleicht die gamescom noch ausgenommen, aber da muss man ja erstmal wissen, wer Entwickler ist und wer nicht. Ich finde solch einen Ansatz sehr interessant, da hier die zwei Seiten quasi aufeinandertreffen und sich austauschen können und Besucher, die keine Entwickler sind, sich informieren können, wie Spiele entwickelt werden und somit einen Einblick in unsere Branche erhalten.

Making Games Planst du, die German DevDay langfristig als jährliches Event aufzuziehen oder ist es für dich erstmal ein »Testbalon«?

Stefan Marcinek Die GDD – German DevDays soll es von nun an jedes Jahr geben. Allerdings bleibt natürlich abzuwarten, ob sie jedes Jahr auch kostenlos sind. Angedacht ist schon, dass sie jedes Jahr auch wachsen und neue Sachen anbieten. Daher gehe ich eher davon aus, dass ein kleiner Ticketpreis verlangt werden müsste, der die Veranstaltungskosten trägt und die ganze Sache plus/minus Null ausgeht. Das Konzept ist dabei nicht gewinnorientiert.

FEBRUAR 2016

Mobile World Congress

Ort Barcelona, Fira Gran Via
Termin 22. bis 25. Februar 2016
Thema Messe/Konferenz
Kosten 749 bis 4999 Euro
Homepage www.mobileworldcongress.com

GamifyConference 2016

Ort München, Haus der Bayerischen Wirtschaft
Termin 23. Februar 2016
Thema Konferenz
Kosten Kostenlos
Homepage www.gamify-con.de

MÄRZ 2016

Game Connection America

Ort San Francisco, AT&T Park
Termin 14. bis 16. März 2016
Thema Messe
Kosten 290 bis 2900 Euro
Homepage www.game-connection.com

GDC San Francisco

Ort San Francisco, Moscone Center
Termin 14. bis 18. März 2016
Thema Konferenz
Kosten 157 bis 1685 Euro
Homepage www.gdconf.com

Eventkalender

www.makinggames.biz

Online-Eventkalender mit Suchfunktion und Veranstaltungsinfos

CeBit

Ort Hannover, Messegelände
Termin 14. bis 18. März 2016
Thema Messe
Kosten 55 bis 1999 Euro
Homepage www.cebit.de

APRIL 2016

International Games Week Berlin

Ort Berlin, verschiedene Locations
Termin 18. bis 24. April 2016
Thema Konferenz
Kosten 20 bis 709 Euro
Homepage www.internationalgamesweekberlin.com

Quo Vadis

Ort Berlin, Café Moskau und Kino International
Termin 18. bis 20. April 2016
Thema Konferenz
Kosten 148 bis 1903 Euro
Homepage qvconf.com

Pax East

Ort Boston, Boston Convention and Exhibition Center
Termin 22. bis 24. April 2016
Thema Messe
Kosten 47 bis 95 Euro
Homepage east.paxsite.com/registration

Making Games Talents Berlin

Ort Berlin, Mediadesign Hochschule
Termin 23. April 2016
Thema Konferenz
Kosten 29 Euro
Homepage www.makinggames.biz/talents

MAI 2016

Media Convention Berlin

Ort Berlin, Station-Berlin
Termin 02. bis 03. Mai 2016
Thema Konferenz
Kosten 99 bis 640 Euro
Homepage www.mediaconventionberlin.com

Passiert auf makinggames.biz



Tencent übernimmt Riot Games

Mitte Dezember hat der chinesische Internet-Gigant Tencent alle verbleibenden Anteile an Riot Games gekauft, der Kaufpreis ist unbekannt. Zuvor waren bereits 93 Prozent der Aktien in Besitz von Tencent. Riot Games und Tencent sind seit 2008 verbandelt, da Tencent seinerzeit dabei half, »League of Legends« in China auf den Markt zu bringen. Die ersten Anteile erwarb die asiatische Firma im Jahr 2009 und investierte 2011 erneut 400 Millionen Dollar, um den Großteil der Aktien zu übernehmen. Riot kündigte nach der Meldung an, mit dem eingenommenen Betrag

seine Mitarbeiter für ihre Leistungen monetär zu belohnen. Tencent betreibt die Messaging-Apps WeChat und QQ und macht die Hälfte seines Umsatzes mit Online-Spielen, außerdem besitzt das Unternehmen Anteile an anderen Spielefirmen wie Activision Blizzard oder Epic Games.



John Smedley gründet Pixelmage Games



Im Februar 2015 verkaufte Sony sein Studio Sony Online Entertainment an eine Investorengruppe und das Studio wurde in Daybreak Game Company umbenannt, die Online-Spiele wie »Everquest« und »PlanetSide« liefen weiter. Im August schließlich verließ John Smedley, bis dahin Präsident des Studios, die Firma, da es nach einer DDoS-Attacke auf die betriebenen Titel nicht nur beleidigende Angriffe auf das Unternehmen, sondern auch auf Smedleys Familie gegeben hatte. Anfang des Jahres meldete sich Smedley mit einer neuen Unternehmung namens Pixelmage Games zurück, die er mit einigen Veteranen aus Sony- und Daybreak-Zeiten gegründet hat. Zeitgleich ging eine Kickstarter-Kampagne für das erste, »Hero's Song« genannte Spiel online, in dem das Team 800.000 Dollar für das Pixel Art, rouge-like MMO einsammeln wollte. Ende Januar wurde der Crowdfunding-Versuch gestoppt, doch nach Angaben von Smedley würden die Arbeiten am Spiel mit Hilfe von Privatinvestments in Höhe von rund einer Million Dollar weitergeführt.

Microsoft und Crytek verstärken Ausbildungsbemühungen

Sowohl Microsoft als auch Crytek wollen 2016 mit speziellen Programmen Ausbildungseinrichtungen unterstützen, das kündigten die Unternehmen Ende Januar an. Microsoft enthüllte eine eigens für Schulen ausgelegte »Minecraft«-Version. Diese wird als Minecraft: Education Edition im Sommer erscheinen und basiert auf MinecraftEdu der Firma TeacherGaming, die der Softwarekonzern aus Redmond kürzlich übernommen hat. Die Schulversion des erfolgreichen Spiels soll Lehrern die Möglichkeit bieten, die Inhalte gezielt auf den Unterrichtsstoff auszurichten. Crytek konzentriert sich derweil auf Hochschulen und startet eine Initiative namens »VR First«. Für diese sollen sich Universitäten und ähnliche Einrichtungen einschreiben können, die auf dem neuen Forschungsgebiet VR tätig sein wollen. Crytek stellt dann wiederum die CryEngine zur Verfügung und gewährt gar Zugriff auf den Quellcode



der Engine – etwas, das Lizenznehmern normalerweise verwehrt bleibt. Außerdem verspricht Crytek, zusammen mit Partnern die notwendige VR-Hardware bereitzustellen. Das erste »VR First«-Labor hat an der Bahçeşehir Universität in Istanbul eröffnet.

Hideo Kojima geht Partnerschaft mit Sony ein



Das Drama um »Metal Gear Solid«-Erfinder Hideo Kojima scheint beendet zu sein. Nachdem sich Kojima im vergangenen Jahr von seinem Publisher Konami getrennt hatte, was zu vielen Gerüchten, der Schließung seines Studios und einem Eklat bei den Game Awards im Dezember führte, hat der japanische Designer nun mit Sony einen neuen Partner gefunden. Mitte Dezember verkündete er zusammen mit Andrew House, dem Präsident und Global CEO von Sony

Computer Entertainment, dass unter dem altbekannten Namen Kojima Productions ein neues Studio eröffnet worden sei. Dort arbeitet das Team an einem Spiel, das exklusiv für die PlayStation 4 erscheinen wird. Um was es sich dabei genau handelt und wann der Titel erscheinen soll, das blieb offen. Allerdings sehen Sony und Kojima das Projekt als Start einer neuen Spieleserie. Kojima betonte aber, dass Kojima Productions als unabhängiges Studio agieren werde.



Tim Schafer finanziert Psychonauts 2 erfolgreich über Fig

Tim Schafer hat es ein weiteres Mal geschafft: 2012 brachten Schafer und sein Team von Double Fine Productions den Kickstarter-Stein in der Gaming-Kategorie ins Rollen, als die Kampagne für das Adventure »Broken Age« rund 3,3 Millionen statt der geforderten 400.000 Dollar einbrachte. Für »Psychonauts 2« setzte Schafer Ende 2015 wieder auf Crowdfunding und konnte rund 3,8 Millionen Dollar für die Entwicklung einsammeln, hatte diesmal aber mit 3,3 Millionen Dollar im Vorfeld ein deutlich höheres Ziel gesteckt. Für Psychonauts 2 wählte Double Fine zudem nicht Kickstarter sondern die neue Plattform Fig, über die nicht nur Geld »gespendet« sondern investiert

werden kann, mit der Aussicht am späteren Erfolg des Spiels beteiligt zu werden. Das Konzept, normale Backer und Investoren auf einem Portal zu vereinen, scheint aufzugehen: Der gesammelte Betrag für Psychonauts 2 setzt sich zu etwa gleichen Teilen aus Crowdfunding- und Investorenbeiträgen zusammen.



Oculus Rift ab April erhältlich

Anfang Januar war es endlich soweit und Oculus gab bekannt, wann die Consumer-Version des heiß erwarteten VR-Headsets Rift erhältlich sein wird. In einem Blog-Eintrag verkündete die Firma rund um Gründer Palmer Luckey den März 2016 als initialen Versandtermin für vorbestellte Geräte, während das Rift ab April im Handel verfügbar sein soll. Die Freude über die Nachricht hielt sich allerdings in Grenzen, denn als Preis nannte Oculus 599 Dollar (für deutsche Kunden: 699 Euro plus Versand), wofür Käufer neben der VR-Brille einen Xbox One Controller, die nötigen Kabel, eine Fernbedienung, einen Bewegungssensor und zwei Spiele (»Evel: Valkyrie« und »Lucky's Tale«) bekommen. Die hohen Kosten verärgerte die VR-Enthusiasten vor allem deshalb, da Luckey zuvor stets von einem ungefähren Preis zwischen 350 und 500 Dollar gesprochen hatte – einen Fehler, den der Rift-Erfinder später auf Reddit einräumte. Gleichzeitig stellte er klar, dass trotz der nun höheren Kosten Oculus keinen Gewinn mit dem Rift mache, Mutterkonzern Facebook bestätigte Ende Januar ebenso, dass die Verkäufe des Headsets keinen Einfluss auf die Finanzen des Unternehmens hätten.



Ubisoft arbeitet mit Hollywood-Studio an VR-Inhalten



Ende Januar hat Publisher Ubisoft angekündigt, eine Partnerschaft mit der Hollywood-Produktionsfirma SpectreVision geschlossen zu haben. SpectreVision wurde im Jahr 2010 unter anderem von Schauspieler Elijah Wood (»Der Herr der Ringe«) gegründet und machte sich seither mit Indie-Horrorfilmen wie »Toad Road«, »A Girl Walks Home« oder »The Boy« einen Namen bei Horror-Fans. Ziel der Zusammenarbeit sollen VR-Erlebnisse sein, unter anderem im Horrorgenre – eine Expertise von SpectreVision, die Ubisoft besonders hervorhob. An welchen Titeln Ubisofts Fun-House-Studio (»Child of Light«, »Grow Home« und »Valiant Hearts«) in Montreal zusammen mit den Filmemachern arbeitet, blieb bisher offen. Ubisoft scheint aber nach der Ankündigung von »Eagle Flight« für PlayStation VR der erste große Publisher zu sein, der an das Potenzial von Virtual Reality glaubt.



Links

Bethesda gibt Einblicke in iteratives Level Design.
www.makinggames.biz/bethesda

Star Wars Undercover Boss: Star Killer Base
www.makinggames.biz/starwars

Scrum in sieben Minuten verstehen.
www.makinggames.biz/scrum

Was den Spaß in Videospielen ausmacht.
www.makinggames.biz/findingfun



BRANCHEN-GEZWITSCHER @MAKINGGAMES_DE/INTERESTING-PEOPLE



@DANIELVAVRA

So I decided to update bios on my msi 970 mobo and now it tells me that it won't boot cause my cpu that is there for a year is not safe. WTF?
Daniel Vavra, Gründer der Warhorse Studios



@GRUMPYGAMER

The most exciting thing about games these days is they can be about something.
Ron Gilbert, Erfinder von Monkey Island



@IBJADE

My flight from SFO to YUL was struck by lightning. Huge ball of electric fire! Happy discovery that planes are designed to withstand this
Jade Raymond, Miterfinderin von Assassin's Creed



@LOCUST9

Oldest daughter just discovered Pokemoncenter.com so there goes my rent.
David Goldfarb, Game Director von Payday 2



@THEREALCLIFFYB

The Internet is like Los Angeles. If you can do it on Your Own Terms, it's great, otherwise, it's Pure Hell.
Cliff Bleszinski, CEO von Boss Key Productions



@ID_AA_CARMACK

Greetings follower 1<<19 over the holiday! More followers in last five months than previous five years an indicator of imminent VR takeoff?
John Carmack, Oculus VR CTO



@GEORGE3DR

Today will be 3 things: some iOS development, playing The Witness and pizza. I may or may not even put on pants today.
George Broussard, Erfinder von Duke Nukem



@MIKEBITHELL

The problem with VR is that there's not any major competitors who will drive down price massively this year. Oh. No. Wait.
Mike Bithell, Entwickler von Thomas Was Alone



@BENJAMINCOUSINS

I'd just like to thank the people behind Assassins Creed for providing such a detailed account of the history of dickheads through the ages.
Ben Cousins, Mitgründer und CEO von The Outsiders

JOIN THE CONVERSATION

TOOLS

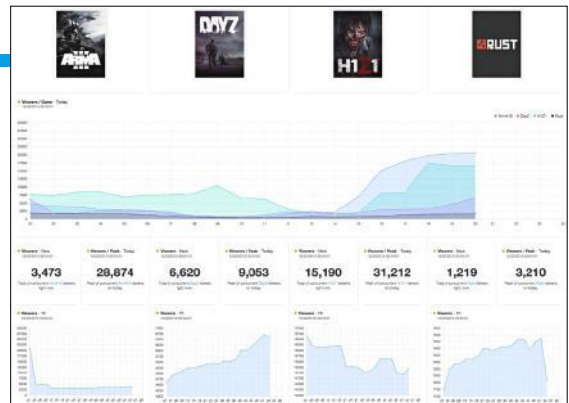
Keine guten Werkzeuge – keine guten Spiele! Experten aus der Branche sagen ihre Meinung zu aktuellen Programm-Updates sowie neuen Tools und verraten ihre persönlichen Software-Geheimtipps.

LOOTILITY WWW.LOOTILITY.COM

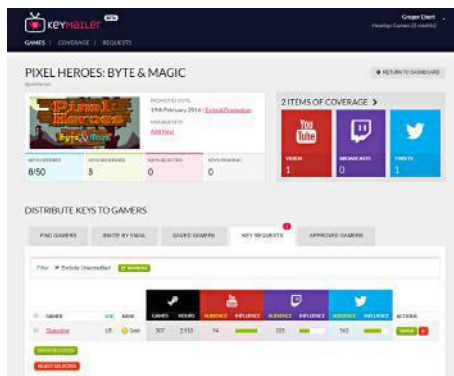


Gunnar Lott
ist Inhaber
der PR- und Textagentur
Visibility Communications.

Lootility ist ein Analyse-Tool für Twitch. Es wurde vom Berliner Team »Fish in the Woods« als Teil einer anderen Anwendung entwickelt und ist derzeit noch im Beta-Betrieb. Lootility zeigt durchsuchbare Live-Daten aus Twitch, von der Zahl der Channels oder Zuschauer pro Spiel, mit Vergleichsmöglichkeiten, bis zu Einblicken in die Channels selber. Wer wissen will, wieviel Prozent der Zuschauer von League of Legends »C9Sneaky« gerade bindet oder ob bei World of Tanks mehr zugeschaut wird als Darkest Dungeon, der ist hier richtig. Lootility ist derzeit noch in der Beta, Invite-Codes verschickt das Berliner Team aber großzügig. Historische Daten sind in der aktuellen Fassung noch nicht einsehbar.



Lootility ist noch im Beta-Betrieb, taugt aber schon für Analysen des Traffics von Twitch.



Wer soll einen Review-Key bekommen? Keymailer zeigt genau die User-Zahlen der verschiedenen Plattformen.

KEYMAILER WWW.KEYMAIL.CO

Keymailer ist ein hilfreiches System zur Verbreitung von Review-Keys zwischen Social Media Influencern und Lizenzgebern. Twitcher und YouTuber melden sich an, werden geprüft und können anschließend entweder direkt kontaktiert werden oder den Key eines Spiels anfragen. Das Einpflegen der Keys auf Publisher-/Entwickler-Seite ist extrem simpel, ausgeweitet werden soll dieses System demnächst sogar noch auf Xbox One und PS4-Codes. Mit jeder Anfrage sind die jeweiligen Userzahlen von Twitch, YouTube und Twitter angegeben; so ist eine schnelle Entscheidung möglich. Wird ein Titel von einem registrierten User erwähnt oder ein Video erstellt, erscheint der Link automatisch in einer Coverage-Übersicht. Keymailer befindet sich aktuell in der kostenlosen Beta-Phase, später werden gezielte Auswertungstools oder Werbemaßnahmen im Tool selbst kostenpflichtig.



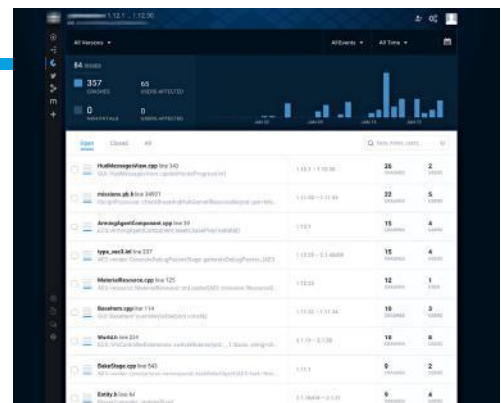
Gregor Ebert
ist PR-Manager
bei Headup Games.

CRASHLYTICS WWW.CRASHLYTICS.COM



Max Röhrbein-Kling
ist Engine Programmer
bei Deep Silver FISHLABS.

Crashlytics, inzwischen ein Teil von Twitters Fabric Framework, dient der Auswertung und Analyse bei Abstürzen von iOS- oder Android-Apps. Dabei sammelt Crashlytics die Daten der Abstürze als sogenannte Crashlogs und bereitet diese in einer für die Entwickler erreichbaren Webseite auf. So kann man beispielsweise die Häufigkeit einzelner Probleme untersuchen oder auch detaillierte Informationen über den Stacktrace einer Ursache erhalten. Die Entwickler können zusätzlich zu den Crashlogs auch weitere Lognachrichten hinterlegen, um die Analyse zu erleichtern. Die gesammelten Crashlogs werden nach Zeit des Auftretens und Version der App gruppiert. Dadurch lassen sich Rückschlüsse darauf ziehen, wann sich Fehler eingeschlichen haben. Crashlytics ist kostenlos, sehr leicht zu integrieren (in Unity3D zum Beispiel als Plug-In verfügbar) und hilft ungemein dabei, ein stabiles Produkt zu entwickeln.

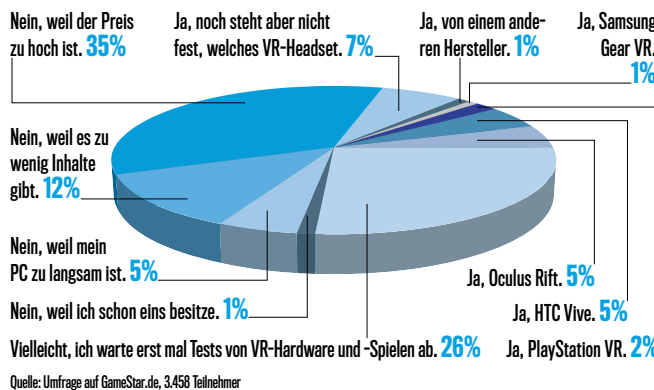


Mit Crashlytics behält ein Entwickler stets den Überblick über aktuelle Probleme seiner App.

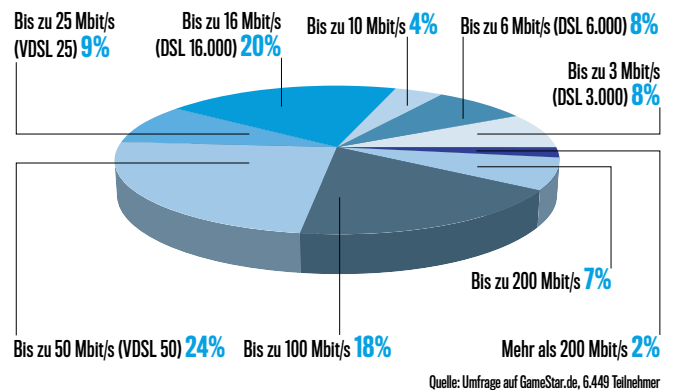
DATAFLOW

GameStar und GamePro befragen regelmäßig Heftkäufer und Website-Besucher zu aktuellen Games-Themen. Eine kleine Auswahl finden Sie auf dieser Seite.

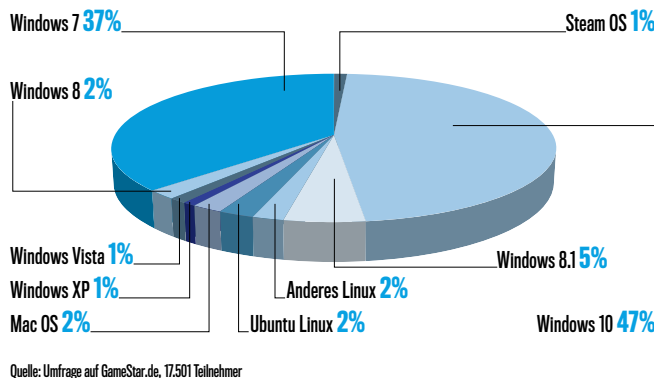
Planen Sie 2016 den Kauf eines VR-Headsets?



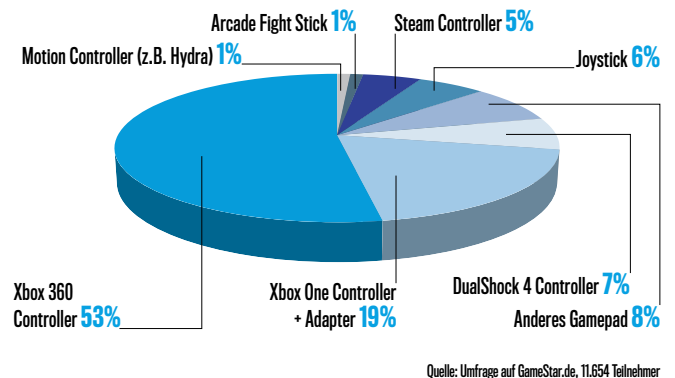
Wie schnell ist Ihr Internet-Anschluss?



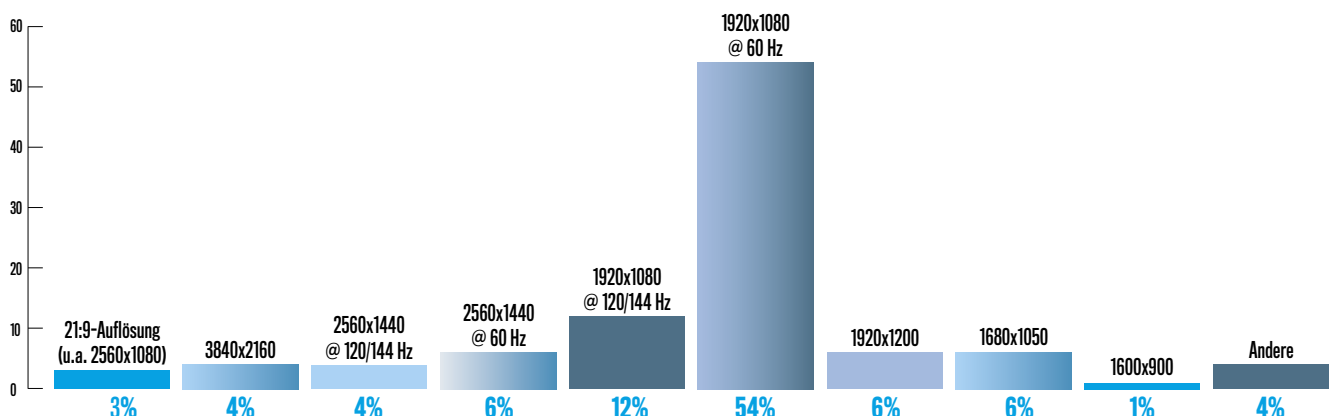
Welches Betriebssystem nutzen Sie hauptsächlich?



Welche Controller nutzen Sie am PC?



In welcher Auflösung spielen Sie hauptsächlich?





HOW NEXT GEN CONSOLES ARE CHANGING DEVELOPERS' OPPORTUNITIES

Who would have thought: The current cycle of consoles is more developer friendly than any of them was before. Jan Klose from Deck13 explains how the situation for independent developers has changed.



Jan Klose
is Creative Director
at Deck13 Interactive.

Jan was one of the men in charge of the development of »Lords of the Fallen«. In 2001 he founded Deck13 Interactive together with Dr. Florian Stadlbauer and worked on about 20 games that were released since then.

Development for consoles never was an easy terrain for independent developers. Especially in Germany which was, and in parts still is, a strong PC country by tradition. While there were always notable exceptions with developers like Keen Games and Shin'en, the few blockbusters produced in Germany that sold significant numbers were PC titles like »Anno«, »Settlers«, and »Gothic«.

With the advent of the »next gen« consoles it wasn't very clear if the big platform holders had any intentions at all to support independent developers on a strong level. Especially when Microsoft started advertising their new platform more or less as a TV set top box.

So to the majority of developers it came as a bit of a surprise when all of a sudden the preconditions changed, and only a few would probably have believed that two years after their debut, the Xbox One and the PlayStation 4 – and even the Wii U – are turning into attractive platforms for independent developers, even enabling them to go without a publisher as a middleman at all. How did that happen, and how can even startup game studios benefit from the trend before this amazing window of opportunity might close again?

The Deck13 Story

Back in 2009, we had developed our own game engine at Deck13. We were using OGRE3D for rendering. Our game »Venetica« was in fact inspired by the »Gothic« series. Not having a multiplatform engine ready, we had to rely on a third party to do a port to PlayStation 3 and Xbox 360. There weren't a lot of German experts around to offer such a port, so we ended up with an experienced French team. The decision to work together with a third party didn't leave us much freedom in terms of the game's look and feel on the consoles. This led to some unfortunate decisions that had a heavy impact on the quality of the final product.

We didn't want this to happen again, so we worked on our very own engine, called »Fledge«, and this time we had a multiplatform approach in mind right from the start.

We needed the support of our publisher to get access to Dev Kits for Xbox and PlayStation. Being new to the console space, and with Dev Kits still costing a fortune, there wasn't any other way.

The games we were creating, called »Blood Knights« and »Tiger & Chicken«, were supposed to be released in the digital stores of PlayStation and Xbox. But even though this meant that we wouldn't need a publisher for

retail, in the end we still needed one because (besides funding) of a strange rule by platform holders that required the studio to have shipped physical titles before they were allowed to publish digitally. This rule led to the fact that basically only traditional publishers were offered »slots« in the stores because ... well, before doing something digital, you had to have a retail product in your track record. And yes, in your own track record, not in the one of your publisher. This effectively ruled out the possibility of digital self-publishing. (Unless you found a publisher who offered »selling« you their slot in the digital marketplace – yes, this did happen.)

After about a year, Fledge was »console ready«. We worked on our next major title, the Action-RPG »Lords of the Fallen«. And all of a sudden, the publisher asked us if we could please »make it next gen« and switch to the new (still unreleased) consoles. Well ... yeah, sure. That felt like the only good answer, and so we started working towards a »next gen« game release, ditching PS3 and Xbox 360, even though we didn't even know the specs of the new hardware. We were just relying on rumors from around the Internet. (The Internet might be a terrible place to find out »the truth« in general, but in terms of the actual console specs it proved quite accurate!)

One of the most interesting rumors was that porting to the next generation was supposed to be incredibly straight-forward. Wait ... porting to a console should suddenly be a simple thing? Well, when the dev kits finally arrived, we found out that there was at least some truth to that. Compared to working with PlayStation 3 or Xbox 360, it was a lot more straightforward, indeed. It felt even more similar to developing for the PC, and the tools provided by the platform holders were much improved compared to »last-gen«.

A year after the release of Lords of the Fallen and two years after the consoles' debut, we're now working on our next big title, the Action-RPG »The Surge«. Was it worth taking the risk and switching to the new tech? Absolutely! With the experience we gathered, we are now able to utilize the consoles in other interesting ways too, and that's mainly because of an entirely new philosophy of the platform holders.

The Platform Holders' new Approach

After the apparent failure of the Xbox Live Indie Games (XBLIG) store concept, Microsoft stated that it will shut down the XBLIG and XNA Creators Club programs, with the XBLIG marketplace shutting down in September 2017. However, instead of ending their partnerships with these small development studios, Microsoft now focuses on their new ID@Xbox program for the Xbox One.

The ID@Xbox program currently has over 1,000 developers creating Indie games for the Xbox One, with many developers receiving

free Dev Kits, getting access to Xbox One and Xbox Live components such as Kinect and SmartGlass, and even a subscription to the Unity engine.

Wait, what was that? No more barriers for Indies on the Xbox? No more barriers to enter the marketplace? Self-publishing accepted? And even a free engine on top? Now how did that happen? And what are the other platform holders doing?

Well, Sony is doing something quite similar, even though it might be a bit more expensive for Indie developers to get Dev Kits. But again, no more barriers like in the old days. Feel free to release your game.

This is an amazing trend, and Nintendo doesn't want to be far behind, offering a comparable concept for their Wii U. So you're ready to go, just get in touch with the account managers and they will help you to get all set. Of course, it might take a while due to the huge number of Indie developers currently subscribing for these services.



Venetica was ported to Xbox 360 by an external partner as Deck13 was not capable of doing it by themselves, leaving the studio with almost no influence of the outcome.



Tiger & Chicken was created with Deck13's proprietary engine Fledge, but back then the engine was not yet fully optimized for consoles.

You can do it. (And everyone else, too.)

So here we go, two years after the new consoles were launched the platform holders are embracing independent developers. It's a wonderful move that makes a much more thriving games economy possible. However, it also causes a much stronger competition. When the barriers to enter the market are that low, a lot of developers will enter it and fight over the gamer's money. Take a look at what's happening with Steam: According to SteamSpy, 2015 saw over 3,000 new games on the platform, compared to about 1,900 in 2014. We can assume that the Xbox and PlayStation stores will see a strongly increasing number of titles in 2016 as well.

Goodbye Mr. Publisher?

So can we developers now sell directly, without a publisher, and finally become rich and famous without being forced to share the spoils? Well, it isn't that easy. Consider why you needed a publisher in the past: Retail, marketing, PR, project management, Quality Assurance tests (especially to meet the TCRs or XRs, the tough set of rules that platform holders require of a game to be published at all), age rating and localization tasks.

But first of all, most developers turned towards a publisher for (relative) financial security. The publisher offers to fund the development in advance, a thing that almost no bank in the world would do. And in most of the publishing contracts, the advances are defined non-refundable even when the game doesn't sell at all. In that case, the publisher loses the investment but the developer is allowed to keep what he already got. The price for this is being paid off with 15 to 25 percent of the publisher's income (after deducting most of his expenses!). This won't change much in the new world. For a big project, you will need the financial power of a publisher and you'll need to give away large chunks of the game's IP and royalties.

For smaller projects, however, it is now entirely different. You don't need a lot of funding. Great engines are available almost for free. You don't need to feed a big team. Great indie games can be created by a handful of people. Smart teams are able to manage themselves.

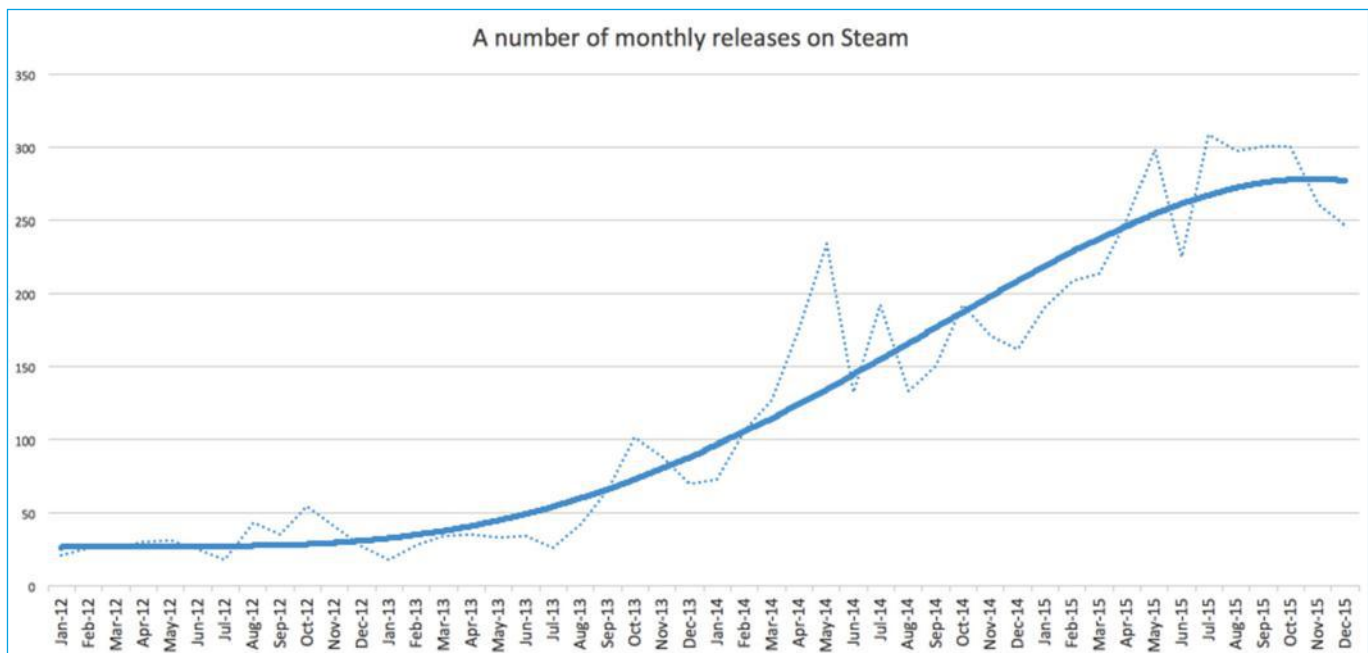
The downside is that, as said above, almost every small studio is able to release their games by themselves, creating an awful lot of competition. Having created a great game doesn't guarantee that you will sell more than a dozen of copies. There are just so many great little games out there that when you're a game developer, you aren't someone special anymore. What you need now is the attention of the players. You need to get the word out. You need to build a community. You need to scream and shout to be heard in the tempest that is social media. When advertising your game, you might feel like you are a salesman knocking on the door of every single player. Those of us who have done a Kickstarter project will know the feeling: Just having »a cool Kickstarter project going« may give you, well, zero backers. You might need to talk, beg, call up friends and family, collect all the favors that people might owe you, just to have your campaign succeeding, even if you had set just a modest funding goal. And that was just the start. If you finance, create, produce and market a game all on your own, this can sometimes feel like a very hard and lonely business – unless you were lucky and hit gold, with just the right product, at just the right time, in just the right place.

You don't need to be alone

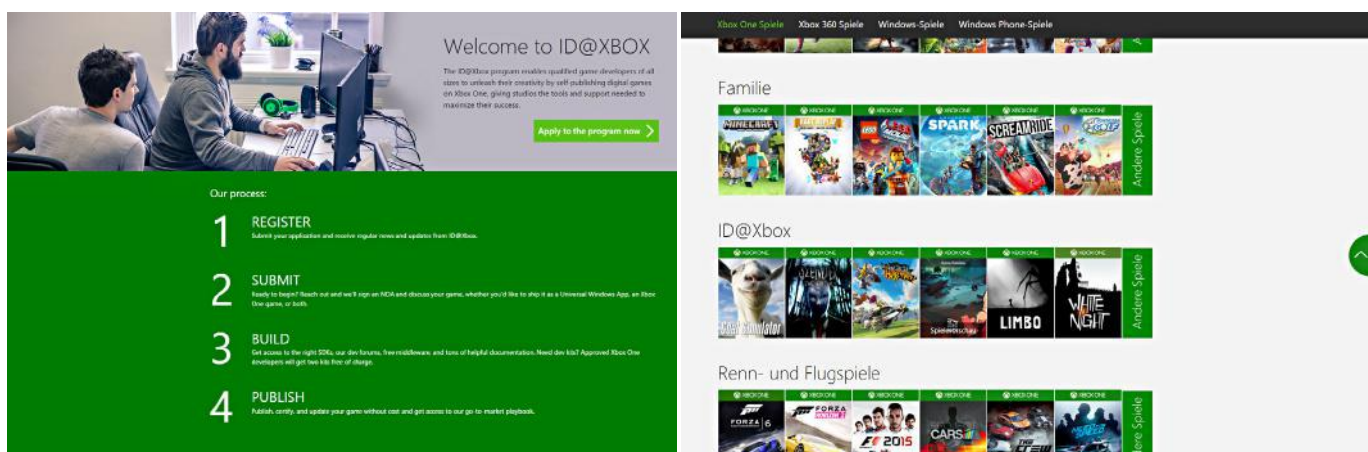
But all in all, for many studios the chances of being truly independent will outweigh the risks. And there are new ways of finding partners for your project. Deck13, as well as other larger developers, offers support in the areas of marketing, PR, production, release, and other fields. You can just call us and pick what services you require. In return, we take a share of the income, but it's not the major



Lords of the Fallen (left) was the first game that Deck13 developed for next gen consoles, using their Fledge engine. Now the team is working on The Surge (right).



This chart by SteamSpy shows how the amount of released games has increased on Steam as now Indie developers can do that on their own - the same situation will come true for consoles sooner or later.



Joining programs like ID@Xbox makes it very easy for independent developers to publish their games on consoles and Microsoft is even advertising those games on its store page - the same is true for Sony and Nintendo.

share, as it was with traditional publishing contracts, and you're free to decide which services you need. Deck13 is just one of many seasoned developers stepping into this field of partnership. As each of these studios can only take on a certain amount of partner developers at a time, there's plenty of space for more players on the market. In any case it's worth a try. Why do it on your own if you can do it together with a partner? Especially, if the partner has experience with developing and releasing for the »next generation« (or now rather: current generation) platforms.

The Road Ahead

The new consoles are established, they have gained decent market shares, their online stores are thriving. Indie developer programs are in place. That's not a bad time to start a project on those platforms. And while big titles with big ambitions and big budgets will

most likely still need big publishers to back them (and things will just stay as they were for many years), smaller independent projects, can now secure more fitting partnerships that allow your studio and your project to stay independent. Where getting a contract and getting funded was the toughest task in the past, today it is getting the players' attention so that they recognize your game out there. But today you are free to choose a partner who helps you just in the area where you need help, so you don't need to end up in a do-or-die situation where you might be forced to sign a crappy contract just because it's your only chance to become a part of the industry.

Jan Klose



PORTING VAN HELSING TO XBOX ONE

The team of Neocore Games calls themselves a traditional PC developer and still the team took the rocky road of porting their Hack'n'Slay Van Helsing to Xbox One. It was a rough journey but in the end it was worth it.



Balázs Farkas
is Communications Manager
at NeocoreGames.

Balázs Farkas, Communications Manager, is working at NeocoreGames since 2015, where his main responsibilities are written communication (mainly the devblog) and research. He holds a Bachelor's degree in Library and Information Science, as well as a Master's degree in Knowledge Management.

The end of 2015 marked a new milestone for NeocoreGames – we finally released the first console port of »The Incredible Adventures of Van Helsing«. Our game has appeared in the Games With Gold program on Xbox One, and essentially doubled our audience. This is nothing unheard of, but we're a small (but growing), independent Hungarian studio, with a portfolio of PC exclusives, so this was a great learning experience.

We have worked with publishers before, but with the Van Helsing franchise, we were on our own since its first release on Steam. Publishers often help with porting to other platforms, but independent developers who can't afford third parties need to make the necessary steps themselves.

Porting games wasn't something that required our immediate attention, nor was it something that we were experienced in, but judging by the success of our game on PC (and the demand for similar games on consoles), we decided that we'll get into it. It was much

more difficult and complex than we anticipated, we weren't quite prepared for all the requirements of the console market. So here's how these things go.

With other Platforms in Mind

First things first, we have our own game engine. This is a rare thing as developers nowadays have the luxury of using very capable



multiplatform engines, even using them for free (up to a point), but if a developer studio decides to use their own game engine, they need to prepare it for porting.

The Incredible Adventures of Van Helsing is an Action-RPG that was developed with a PC gaming mindset. While the divide between PC gaming and console gaming is starting to vanish, it still exists, especially in Hungary where console gaming gained popularity at a much slower pace due to various reasons. Most of the senior developers in this country are more comfortable with PC games.

This was true for NeocoreGames as well. Previously we made games in genres that are traditionally for the PC (real-time strategies with RPG elements), and we were used to a kind of mental picture of a gamer sitting in front of a monitor with the convenience of clicking on things with a mouse.

You can see where this is going, but fortunately, early in the development of Van Helsing we were getting to know the console gaming mindset as well. PlayStation 3 and Xbox 360 were quite capable machines, and mobile gaming was on the rise (that was around 2011 and 2012).

So we kept this in mind, and we started experimenting with porting to OSX (this was successful), Linux, iOS and Xbox 360, but we soon encountered difficulties.

For iOS, we managed to complete a port, technically speaking, but we soon realized that the game interface wouldn't work and all the skill trees and gameplay mechanics needed to be redesigned according to resolution, screen size and ratio for multiple models of iPads. As for the Linux version, we just didn't have the capacity to test multiple distributions extensively, so we never released it.

We were always considering a controller support, though. Playing an isometric Action-RPG with a controller was a neat idea, but there were artifacts of the PC gaming mindset – some skills needed a cursor to point, what needed to be addressed (later on console ports as well), and little things like this made the development always slightly more complicated.

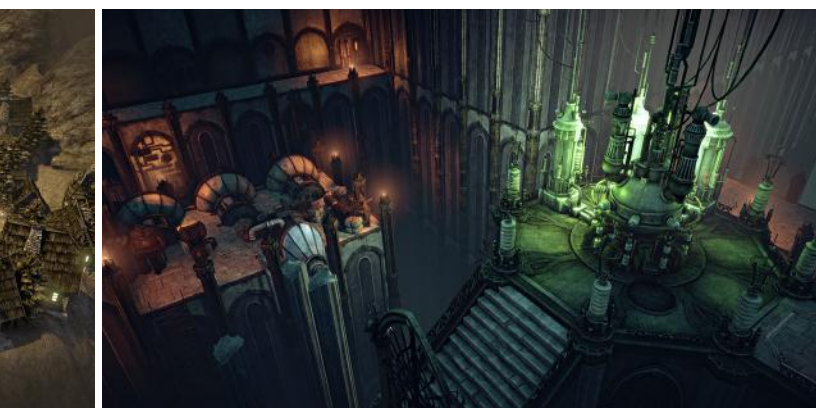


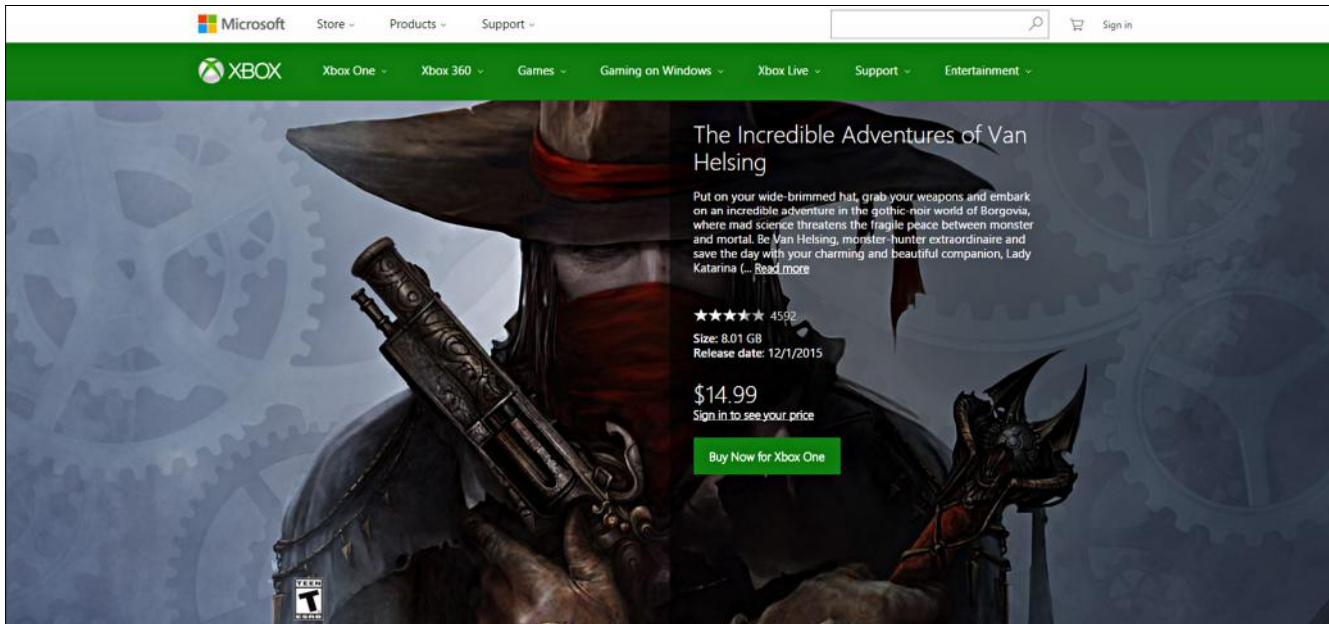
The Incredible Adventures of Van Helsing is a Hack'n'Slay comparable to Diablo. The team released three games within the franchise so far.

Preparing for the Next Generation of Consoles

Good news or bad news, you decide: We started porting our game to consoles on the turning point of generations. PlayStation 4 and Xbox One were coming, and they were coming fast. We already announced (or at least hinted at) ports for the previous generation of consoles, so the communication to our fans got awkward for a while, and we were unsure about how this would play out.

Originally the team planned to release the game for PlayStation 3 and Xbox 360. But NeocoreGames quickly realized that the performance of the next gen consoles would fit the game better.





The Incredible Adventures of Van Helsing is available for \$14,99 in the Xbox Store, the same price that PC users have to pay on Steam. In December 2015 the game was available for free for Xbox Live Gold users.

But the porting itself provided the answer: Van Helsing needed much work to make it run on PlayStation 3 and Xbox 360. For optimal performance, we needed to redesign and scale down much of our graphical content, but we had no capacity to do so. Eventually, we gave up on the idea to release the game on PlayStation 3 and Xbox 360. We looked forward to the next generation of consoles, as they were much more powerful and flexible machines.

This helped much indeed. But these consoles have very strict guidelines and quality control to ensure the comfort of players, and once we started working on the console ports, meeting the requirements of the certificates was taking up 70 percent of the time spent on developing them.

These aren't unreasonable requests, mind you, but for a PC developer, they are sometimes unexpected. These include cloud saving, suspended game (you need to ensure your game doesn't crash or quit when the console goes to rest or sleep mode), different requirements for PlayStation Network and Xbox Live features, the framework for achievements and trophies, the different aspects of multiplayer, interface, controllers, and so on.

Again, these are all very, very reasonable requests. But we were often surprised as we did not consider them previously. This is not because of lack of research or laziness, as we were aware of these features (the guidelines of the console manufacturers are very clear), but we were surprised how these features can mess up our work that we considered done (our carefully selected font looked nothing like on PC for example), and our engine wasn't quite on the same wavelength, so to speak. This meant extra work as well as extra time.

The Gravity of the Task

But how much extra work exactly? By the time we arrived at Xbox One's certificate process, the whole NeocoreGames team consisted approximately of 40 to 50 people. This number seems big, but it includes marketing, localization, side projects, support, communication, etc. Considering all this, we are a small team for such a big project (and we can safely say that Van Helsing got really big in just two years).

We have two coders working on the console ports, full time. We have one or two internal testers, depending on what stage we are at in the development, but we work with a third party company that specialises in testing the games, so this helps. We also have an interface designer, and one or two people communicating with Microsoft and Sony (there are technical and business related questions).

And how much extra time? Well, this is a bit unorthodox, since we made small mistakes, oversights in the process, and our engine needed much attention, and ... it took us several months to get a former PC exclusive released on Xbox One. With the sequels and our future projects, it won't take this much time, it will go much faster. We have learned practically everything during this work.

But, yes. Our first big console release took months. This is partly due to the fact that the development of The Incredible Adventures of Van Helsing III and eventually Final Cut (a free upgrade that includes all three campaigns in a much bigger game) overlapped with the development of the port, but still. PC developers who are considering console ports with a custom engine, need to prepare for the gravity of the task. It gets easier, it gets faster, but as a developer, you need to be very careful

NeocoreGames

NeocoreGames is an independent video game development company that was founded in 2005 and that is located in Budapest, Hungary. Currently the team is working on »Warhammer 40,000: Inquisitor – Martyr« that was announced for PC, Mac, Xbox One and PlayStation 4. In addition, the team is said to be still working on »Broken Sea«, a roleplaying game that was announced in 2012. Before that Neocore Games worked on the King Arthur series and the series of games that are dealing with the theme of Van Helsing.



to see if you have the capacity, knowledge and time needed. Fortunately, we did. And we can safely say, that it was worth it.

After Release

We were lucky enough to have gotten accepted for the Games With Gold program. This is something that helps gain audience for first time console developers. Since we have sequels and a spinoff on Steam already, this will help us find an audience for them on Xbox as well.

Other than that, communication and support, for now, seems the same as on PC. One exception is the question of patches and updates, as they need to go through the same certification process and tests as the main game. There's no workaround, there are no beta patches, so if you're going to release your game and your patch, it better be top notch, because it takes a bit more time to work on them.

All things considered, it's an exciting process in an exciting time for console gaming, and we look forward to release our other games as well, with the hope that our future projects (including »Warhammer 40,000: Inquisitor – Martyr«) can be released on three platforms simultaneously. That would be great.

Conclusion

If traditional PC (Steam) developers are thinking about porting their game to other platforms, they need to be familiar with all the tasks that are needed to make a successful and working port. It boils down to the following principles:

- Make sure you create or license a game engine that meets all the requirements of different platforms (popular open source engines are developed and updated with this in mind).
- Consider the hardware capabilities and possibilities of the targeted platform, including suspended game mode, sleep mode, removal of different cables, etc.
- Consider the features of different controllers, motion sensors, audio, VR, etc.
- Consider that multiplayer works differently on various platforms, and that online features can be used in single player games as well (cloud saving, leaderboards, etc.).
- Design your game interface with the resolution, font set, screen area of the target platform in mind.
- Make sure that the team can afford the extra time and work that is needed (remember, this could mean months!). Make sure that your work will be worth it, ask yourself the question: is there an audience on the targeted platform that will be interested in your game?
- Consider the different mindsets of PC gamers and console gamers.



A comparison with the PC version of the game (lower image) shows how the team adapted the user interface to console controls.

- Read the quality and certification guidelines of the target platform thoroughly, and make sure you meet every requirement before submitting.
- Ensure that there are people in your team assigned for support and communication, especially around the time of release and patches.

And that's our wisdom on the subject as of now. Hopefully it helps others. **Balázs Farkas**





ERFAHRUNGEN BEI DER PORTIERUNG VON PC-SPIELEN AUF DIE PLAYSTATION 4

Gaming Minds hat sein Strategiespiel »Grand Ages: Medieval« 2015 nicht nur auf dem PC, sondern auch auf der PS4 veröffentlicht. Daniel Dumont blickt zurück und erklärt, welche Probleme die Portierung mit sich brachte und wie sich Sonys Konsole weiterentwickelt hat.



Daniel Dumont
ist Mitgründer und
Creative Director
der Gaming Minds Studios.

Daniel ist seit mehr als 16 Jahren Spieleentwickler. Er entwickelte bei Ascaron die »Patrizier«-Reihe weiter und entwarf »Port Royale« sowie »Darkstar One«. Im Jahr 2009 gründete er zusammen mit Kay Struve und Kalypso Media die Gaming Minds Studios, die seither an Spielen wie »Patrizier 4«, »Port Royale 3«, »Rise of Venice« und zuletzt »Grand Ages: Medieval« arbeiteten. Sein Schwerpunkt liegt auf der Visionierung sowie auf der Konzeption von Computerspielen.

Unser Team entwickelt seit fast 20 Jahren Spiele für verschiedene Plattformen, darunter sind drei Titel für die Xbox 360 und zwei für die PlayStation 3. »Grand Ages: Medieval« ist der erste Titel, den wir für die PlayStation 4 entwickelt haben.

Ein Lob an Sony: Grundsätzlich hat sich die Entwicklungsumgebung von Sony seit der PlayStation 3 deutlich weiterentwickelt. Die Tool-Chain ist sehr solide, die Dokumentation umfangreich und in fast allen Bereichen angemessen. Der Entwicklersupport von Sony reagiert normalerweise auch bei Fragen mit niedriger Priorität innerhalb weniger Stunden.

Grafikanpassungen

Wegen der modernen Hardware der PlayStation 4 waren keine speziellen Texturen, 3D-Modelle oder Shader notwendig. Selbst die Assets

für die hochdetaillierte PC-Grafik haben auch auf der PS4 gut funktioniert. Nur die Texturen und Shader mussten wir vor der Verwendung in PS4-spezifische Formate konvertieren.

Unsere Engine verwendet eine Abstraktion der Rendering-API, die sich grob an DirectX 11 orientiert, und ließ sich relativ problemlos um einen Renderer für die PlayStation 4 erweitern. Um den zeitlichen Aufwand der Portierung zu verringern, haben wir bei Grand Ages: Medieval auf die Verwendung einiger fortgeschrittener Features wie Compute Shader verzichtet (vgl. dazu das Schaubild in [Bild 1](#)).

PS3-Erbe

Das User-Interface im Spiel und viele der PSN-spezifischen Features sind noch wie auf der PS3, daher sind viele Erfahrungen aus der letzten Generation noch gültig (etwa Trophies, Leaderboards, PSN-Userverwaltung, PSN-Store-Anbindung).

Da wir das Spiel mit dem PC als Leadplattform entwickelt haben, hat der Gamepad-Support aber vor allem bei der Konzeption und Programmierung des Interfaces (zum Beispiel HUD, Dialoge, 3D-Karte etc.) einiges an Arbeit bedeutet, zumal unser Ziel ein typisches Konsolen-Interface war. Normalerweise entwickeln wir das Maus- und das Gamepad-Interface gleichzeitig. Im Falle von Grand Ages: Medieval war anfangs aber noch nicht klar, ob wir eine Gamepad-Unterstützung brauchen würden, daher haben wir die erstmal beiseitegelassen und mussten sie dann eben nachträglich einbauen. Das ist insgesamt mehr Arbeit und normalerweise nicht zu empfehlen, denn wenn beide Designs gleichzeitig gemacht werden, lassen sich die Kompromisse in der Steuerung leichter finden.

Grundsätzlich halten wir uns für den Gamepad-Einbau an zwei Regeln, die in dem

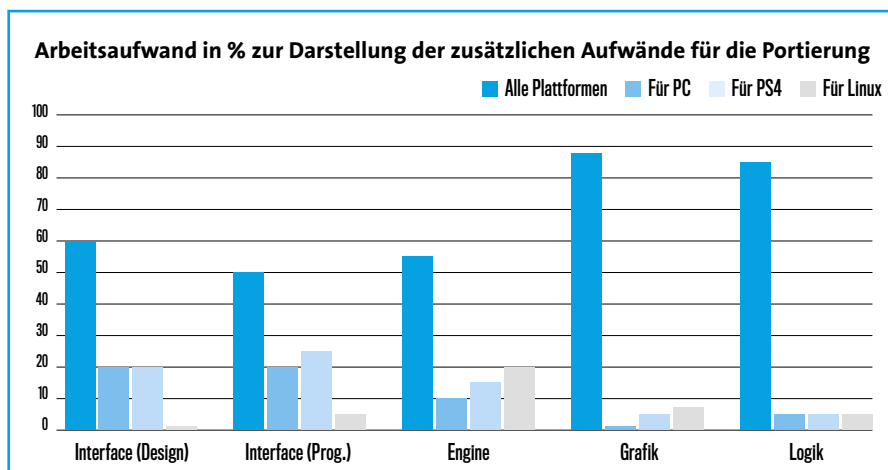


Bild 1: Der Graph zeigt, wie sich der Arbeitsaufwand für verschiedene Entwicklungsschritte auf die unterstützten Plattformen verteilt.



Bild 2: Die beiden Bilder zeigen das PC- und Konsolen-Interface im Vergleich.

Bild-Beispiel (**Bild 2**) noch einmal verdeutlicht werden:

1. In Dialogen darf es nur einen klaren Fokus geben. Anders als bei der PC-Version darf man auf Konsole nicht erwarten, dass der User in einem Dialog verschiedenste Elemente unterschiedlicher Größe anklickt.
2. Es sollte nach Möglichkeit keine Buttons im Dialog geben; vor allem die vielen kleinen Buttons, die in PC-Dialogen manchmal üblich sind (zum Beispiel für »vor« und »zurück«) müssen mit dem Gamepad gelöst werden. Grob gesagt ersetzt man einen Button im PC-Dialog durch eine Gamepad-Taste, die dann in der Legende im HUD angezeigt wird.

Zum Vergleich nehmen wir einen PC-Dialog mit vielen Buttons und das Konsolen-Pendant ohne Buttons (siehe **Bild 2**) her:

Die Buttons im PC-Dialog sind in Bild 2 mit 1 bis 7 nummeriert. Sie sind von unterschiedlicher Größe und Machart. Manche bestehen nur aus Text, manche aus Symbolen, andere sind Grafiken. Außerdem sind sie über den ganzen Dialog verteilt. In der Konsolenvariante sind nur noch die großflächigen Bild-Buttons 5 und 6 übriggeblieben. Sie werden mit dem linken Stick angesteuert und der X-Taste ausgewählt. Die Buttons 1, 2, 3, 4 und 7 hingegen sind in die Legende verschoben worden, denn sie werden direkt über eine Gamepad-Taste angesteuert.

Hardware

Die Ähnlichkeit der PlayStation-4-Hardware zu PCs und auch die Integration des Compilers in Microsoft Visual Studio haben die Portierung deutlich vereinfacht. Die Optimierung der CPU-Nutzung stand für die PS4-Version mehr im Fokus als auf dem PC, wovon aber natürlich auch die anderen Plattformen profitierten.

Zum Beispiel verwendet unsere User-Interface-Library ziemlich viel ActionScript-Code. Diese Programmteile waren oft für Framerate-Einbrüche verantwortlich. Hier haben wir durch genauere Analysen und Experimente ermittelt, welche GUI-Elemente dafür verantwortlich sind, und diese dann überarbeitet beziehungsweise ersetzt.

Sony stellt gute Profiling-Tools zur Verfügung, mit denen man die Performance detailliert überprüfen kann. Das CPU-Analyse-Tool kann zum Beispiel den Programmablauf über viele Frames aufnehmen und dann untersuchen, welche Funktionen auf welchem Thread für einen Framerateneinbruch verantwortlich sind.

Insgesamt haben wir aber im Vergleich mit der Portierung von »Sacred 2« oder »Port Royale 3« für Xbox 360 und PlayStation 3 deutlich weniger Zeit für die Optimierung aufwenden müssen.

Submission

Die PS4-Mastererstellung ist etwas einfacher geworden und die Submission-Testläufe sind deutlich verkürzt – das hilft besonders am Ende des Projekts und nach dem Release. Dennoch war die Submission ein nervenaufreibender Meilenstein: Die Checklisten sind stark überarbeitet worden und daher sind hier die Erfahrungen aus früheren PS3-Submissions nur teilweise hilfreich.

Bei der Submission hatten wir zum Beispiel einen kritischen Fehler, der durch ein Missverständnis der Definition von »user generated content« (UGC) entstanden ist. In unserem Spiel kann der Spieler Städte, die er kontrolliert, umbenennen. Dieses Feature fällt unter UGC, und es muss daher sichergestellt werden, dass junge Spieler, bei denen die Jugendschutzeinstellung UGC verbietet, im Mehrspielermodus die von anderen Spielern erfundenen Stadtnamen nicht sehen können.

Wir hatten auch einen Fehler in der IP-Adressermittlung, der bei uns und den Teststudios nicht auftrat, da wir keine Tests im Mehrspielermodus gemacht hatten, bei denen die User hinter verschiedenen Routern angemeldet sind. Leider konnten aufgrund dieses Fehlers bei Release nur sehr wenige Kunden Mehrspielerpartien beitreten, und wir mussten kurzfristig ein Update bereitstellen.

Updates sind auf der PS4 sehr einfach: Das Patch-System ist unkompliziert und hat viele Features, die Submission-Tests sind sehr schnell, und die Bereitstellung ist mit wenigen Klicks erledigt.

Daniel Dumont



Da die PlayStation 4 technisch ähnlich aufgebaut ist wie der PC, waren grafisch nur wenige Änderungen an Grand Ages: Medieval nötig.



TECHNISCHE HERAUSFORDERUNGEN EINER PLAYSTATION-PORTIERUNG

Mit dem Erfolg auf dem PC bringt Kalypso Media sein Strategiespiel »Dungeons 2« auch auf die PlayStation 4. Warum Unity hier eine gute Wahl ist und welche technischen Probleme auf PC-Entwickler zukommen können, erklärt das Team der Realmforge Studios.



Christian Schlütter
ist Games Producer
bei Kalypso Media.

Christian Schlütter startete seine Karriere in der Gamesbranche als Volontär und Redakteur beim Spielmagazin »PC Games«. Seit 2010 arbeitet er als Games Producer beim Wormser Publisher Kalypso Media.

Nachdem Dungeons 2 am 24. April 2015 für PC, Mac und Linux erschienen war, wurde relativ schnell klar, dass das Münchener Entwicklerstudio Realmforge damit Extraarbeit bekommen würde – und zwar auf die bestmögliche Art. Spieler und

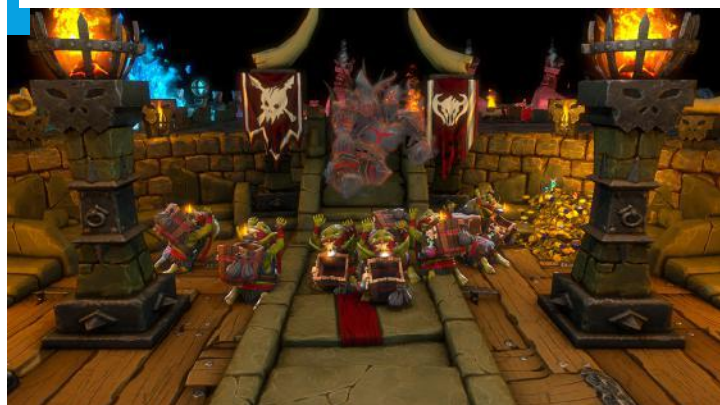
Kritiker waren angetan vom Mix aus Dungeon-Management und Echtzeit-Strategie, und auch mit den Verkäufen konnte Publisher Kalypso Media zufrieden sein. Gemeinsam entschied man, parallel zu den bereits geplanten DLCs für die Desktop-Versionen auch eine Portierung auf die PlayStation 4 anzustreben. Für die Konsolenumgebung war Dungeons 2 damit aber in zweierlei Hinsicht ein Sonderling: Zum einen war es ein Strategiespiel, zum anderen ein Projekt, das ursprünglich nicht mit Blick auf eine Konsolenveröffentlichung entwickelt wurde. Beide Punkte sollten

Realmforge in den folgenden Monaten vor Herausforderungen stellen.

Grundlegende Änderungen der Steuerung

»Geht nicht davon aus, dass ihr die Controller-Belegung eins zu eins mappen könnt!«, rät Immanuel Scholz, der bei Realmforge unter anderem die Programmierung der Controller-Steuerung übernommen hat, wenn man ihn nach Tipps für die Portierung eines PC-Spiels auf die PlayStation 4 fragt. Abgesehen von den offensichtlich notwendigen Änderungen (wie etwa keine Mauseingabe, nur limitierte Anzahl an Buttons und so weiter) kommen bei der PS4-Entwicklung nämlich zusätzliche Probleme auf einen Entwickler zu – gerade bei Strategiespielen. Einfach die Tastenbelegung der PC-Version zu nehmen und eins zu eins auf die Buttons des PlayStation-4-Controllers umzumünzen, das wird nicht funktionie-

Genau wie Dungeons (2011) ist auch Dungeons 2 eine Hommage an den Klassiker Dungeon Keeper, wurde aber um einen Strategiepart in der Oberwelt erweitert, der an Warcraft erinnert.





ren. Dafür gibt es zu viele kontextsensitive Areale, zu viele Sonderfälle, zu vieles, das mit Sticks und D-Pad erledigt werden muss. Eine grundlegende »Von-Hand-Programmierung« der einzelnen Befehle ist also zielführender. Zudem sollte von vornherein eingeplant werden, dass neben der eigentlichen Steuerung auch das entsprechende GUI und die Texte angepasst werden müssen.

PC-Strategiespieler sind es zum Beispiel gewohnt, zu vielen Schaltflächen und Grafiken Mouseover-Texte und -Erklärungen angezeigt zu bekommen. Genau diese aber sind nicht ohne weiteres auf der Konsole umsetzbar. Realmforge zeigt die notwendigen Informationen auf der PlayStation 4 nun kontextsensitiv in einem Rahmen am unteren Bildschirmrand an (siehe Bild 1). Dieses Areal wiederum steht nur zur Verfügung, weil die dort zuvor platzierten Aktions-Schaltflächen auf ein übersichtliches Ringmenü verteilt werden. Dieses zeigt, abhängig von der derzeit ausgewählten Einheit oder dem derzeit gewählten Raum, die möglichen Aktionen an.

Hinzu kommt dann die textliche Anpassung. Tutorial-Texte wie »Klicken Sie mit der rechten Maustaste auf dieses Objekt« müssen ausgetauscht und an die neue Plattform angepasst werden. Als mausintensives Spiel läuft Dunge-



Bild 1: Realmforge entschied sich für die PS4-Version dafür, Aktions-Schaltflächen in einem Ringmenü unterzubringen, die sonst per Maus oder Hotkey ausgeführt werden.

ons 2 zudem auf der PlayStation 4 in eine Falle, die den meisten Jump'n'Runs und Shootern der Konsolenwelt erspart bleibt: Es gibt keinen Hardware-Cursor. »Auf dem PC nimmt sich die Unity-Engine einfach die Informationen des Hardware-Cursors und gibt sie an das Spiel weiter. Auf der PlayStation 4 geht das nicht. Also mussten wir dafür selbst eine Routine schreiben«, erklärt Immanuel die Schwierigkeit, dem Spiel überhaupt Informationen darüber zu geben, wo der Spieler nun gerade hinzeigt beziehungsweise hinklickt. Resultat ist die »overriddenMousePosition«:

```
public static LocationService location { get { return Input.location; } }
public static bool mousePresent { get { return Input.mousePresent; } }
public static int touchCount { get { return Input.touchCount; } }
public static Touch[] touches { get { return Input.touches; } }

public static Vector2 overriddenMousePosition = new Vector2(float.NaN, 0);
public static Vector2 mousePosition
{ get { return float.IsNaN(overriddenMousePosition.x)
    ? overriddenMousePosition : (Vector2)Input.mousePosition; } }
public static void ResetOverriddenMousePosition()
{ overriddenMousePosition = new Vector2(float.NaN, 0); }

public static bool compensateSensors
{
    get { return Input.compensateSensors; }
    set { Input.compensateSensors = value; }
```



Technische Einschränkungen

Einige Mühen bereitete den Codern von Realmforge die nötige »Ahead-of-Time-Compilation« auf der PlayStation 4. Die Ursache hierfür liegt im C#-Code, in dem Dungeons 2 mit der Unity-Engine erstellt wurde. »Generische Datenklassen wie eine Liste werden auf den PC-Plattformen vom JIT-Compiler erst zur Laufzeit kompiliert«, so Korbinian Abenthum, Technical Lead von Realmforge. Das Problem: Aus Sicherheitsgründen verbietet die Architektur der PlayStation 4 es, dass eine Applikation während der Laufzeit Programmcode erstellt und ausführt. Vielmehr müssen sämtliche Listen-Varianten bei Programmstart, also »ahead-of-time«, bekannt sein und kompiliert worden sein. Die PC-Version von Dungeons 2 setzte aber stark auf die Erstellung und Verwendung solcher Daten während der Laufzeit.

Jeden einzelnen dieser Fälle zu finden und zu ändern, erwies sich als Fleißarbeit, wie Programmierer Jörg Böhnel zu berichten weiß: »Das Schwierige daran ist, dass so etwas nicht beim Kompilieren im Unity-Editor als Fehler angezeigt wird. Man merkt es erst dann, wenn einem das Spiel während einer Partie an einer

bestimmten Stelle abstürzt. Und teilweise kann das in sehr obskuren Fällen vorkommen. Zum Beispiel wenn eine einzelne Einheit eine bestimmte Spezialfähigkeit auf ganz spezifische Art einsetzt. Tritt dann ein Absturz auf, heißt es: »Crashlog analysieren, Fehler beheben und neu kompilieren.«

Aufwändige Optimierung

Erwartungsgemäß war die Optimierung eine der größten Aufgaben bei der Portierung von Dungeons 2 auf die PlayStation 4. Auch wenn die Next-Gen-Konsolen weitaus mehr Kraft mitbringen als ihre Vorgänger, so verlangt ihre spezielle Hardware-Architektur doch Anpassungen gegenüber der PC-Entwicklung. Die CPU der PlayStation 4 besteht aus acht Kernen, aufgeteilt in zwei Cluster. Kerne 1 bis 4 bilden den ersten Cluster, Kerne 5 bis 8 den zweiten. Die Kunst besteht darin, sein Spiel so zu verändern, dass die einzelnen Berechnungen (wie Rendering, KI, GUI und so weiter) optimal auf die verschiedenen Kerne verteilt sind, keine Operation zu lange auf eine andere wartet und die Prioritäten der jeweiligen Threads zu jeder Zeit richtig gewählt sind. Letzteres ist leichter gesagt als getan. Denn aus Stabilitätsgründen definiert die Unity-Engine für viele Operationen die Priorität bereits vor.

»Wir mussten also quasi an Unity vorbei arbeiten, um die Threads so zu priorisieren, wie es für Dungeons 2 am besten war«, erklärt Jörg. Die Cluster-Architektur wirft ein weiteres Problem auf: »Ist ein Kern mit einer Operation ausgelastet, tendiert die PlayStation 4 dazu, den korrespondierenden Kern aus dem anderen Cluster hinzuzuziehen. Dabei tritt nur eine winzige Verzögerung auf. Passiert das aber mehrere Hundert Mal pro Sekunde, ergeben sich merkliche Performance-Probleme. Wir haben also versucht, die Operationen so gut wie möglich auf die Kerne des ersten Clusters zu verteilen, damit das System gar nicht erst auf die Idee kommt, den zweiten Cluster zu Rate zu ziehen (siehe Bild 2)«, so Julian Moschüring, der bei Realmforge maßgeblich für die Optimierung der PS4-Version zuständig ist. Ihm fällt auch die undankbare Aufgabe zu, die Update-Routinen von Dungeons 2 für die Konsolenversion zu überarbeiten.

Die Grundeinstellungen von Unity bestimmen, dass alle Objekte im Spiel (»game objects«) kontinuierlich aktualisiert werden. Auch wenn sie nicht benutzt werden, werden game objects also nicht abgemeldet, was Rechenzeit sparen würde. Für viele Spiele ist dies kein Problem, gerade wenn fast nur statische Objekte benutzt werden. Dungeons 2 allerdings bietet neben der Echtzeitstrategie-Oberwelt auch noch einen komplett ausbaubaren Dungeon.

Jedes einzelne Tile in diesem Dungeon kann zu jeder Zeit ausgegraben, bebaut oder sonst



Die Screenshots zeigen die PC-Version (unten) und wie sich das User Interface im Vergleich auf der PS4 verändert hat (oben).

wie verändert werden. Jeder Charakter kann die unterschiedlichsten Aufgaben übernehmen und an jeder Stelle sein. Eine Menge Dinge wie Gold, Mana und Fässer können von Charakteren herumgetragen werden. Sprich: Jeder Dungeon besteht aus gut über 30.000 game objects, die mitberechnet werden müssten, selbst wenn sie gerade nicht benutzt werden oder nicht einmal sichtbar sind.

Julian und seine Kollegen mussten also Unitys Update-Routinen so optimieren, dass alles, was gerade nicht gebraucht wird, abgemeldet oder nur rudimentär aktualisiert wird. Das betrifft »nav agents« für stillstehende Kreaturen genauso wie Partikelsysteme und Lichtquellen, die gerade nicht sichtbar sind.

»Im Laufe der letzten Monate konnten wir so die ursprünglichen rund 3.000 Update-Calls pro Frame auf 50 reduzieren«, berichtet Julian. Die dadurch erreichten Performance-Gewinne kommen durch Patches auch der PC-Version zugute.

Die passende Engine und wichtige Vorbereitungen

Auch wenn Unity in der Praxis nicht das One-Click-Wunder ist, das viele sich erhoffen, hat sich die Engine doch als richtige Wahl herausgestellt. Gerade die direkte Kommunikation mit den Engine-Entwicklern ist Realmforge positiv aufgefallen. »Der PS4-Support von Unity über Foren und Tickets ist wirklich super!«, so Jörg. Und er hat auch noch einen ganz grundlegenden Tipp für alle angehenden PlayStation-Programmierer: »Besorgt euch frühzeitig eine statische IP!« Ohne diese wird man nämlich bei Sony gar nicht erst als PS4-Entwickler registriert. Und je nach Internet-Provider kann sich das als monatelange Odyssee herausstellen.

Christian Schlütter



Das Technikteam von Realmforge

Senior Programmer Julian Moschüring, Jörg Böhnel und Immanuel Scholz (v.l.n.r.) sowie Technical Director Korbinian Abenthum (nicht im Bild) bilden beim Münchener Entwickler Realmforge das technische Rückgrat und haben in den letzten Monaten Dungeons 2 vom PC auf die PlayStation 4 portiert.

Korbinian ist einer der Gründer der Realmforge Studios und hat dort in dunklen Zeiten als einziger Programmierer an einer modifizierten Ogre-Version herumgebastelt. Diese Tage sind zum Glück vorbei, heute konzentriert er sich auf das große Ganze. Nach einer langen Odyssee durch verschiedene Engines ist für ihn Unity der Lichtblick am Horizont.

Julian ist 2009 bei Realmforge eingestiegen und kümmert sich seitdem um alles, was mit Grafik und Performance-Optimierung

zu tun hat. Die Grafiker bei Realmforge huldigen ihm, damit er sie mit neuen Features versorgt. Shader, visuelle Effekte und Multi-Threading sind genau sein Ding.

Immanuel ist seit 2010 an Bord und hat schon an Dungeons 1 mitgetüftelt. Er verschwindet nur zu gerne in den Tiefen der Gameplay-Programmierung, kümmert sich aber nebenbei auch um KI, Wegfindung oder baut neue Zauber und Fertigkeiten ins Spiel ein – wenn ihn Game-Designer mit genügend Gummibären oder anderem Süßzeug bestechen.

Jörg ist ebenfalls seit 2010 mit dabei. Wenn er nicht gerade an der Client-Server-Architektur arbeitet, den Assemblercode studiert oder seinen beeindruckenden Bart pflegt, dann ist er in Krisenzeiten immer dort zur Stelle, wo es brennt.

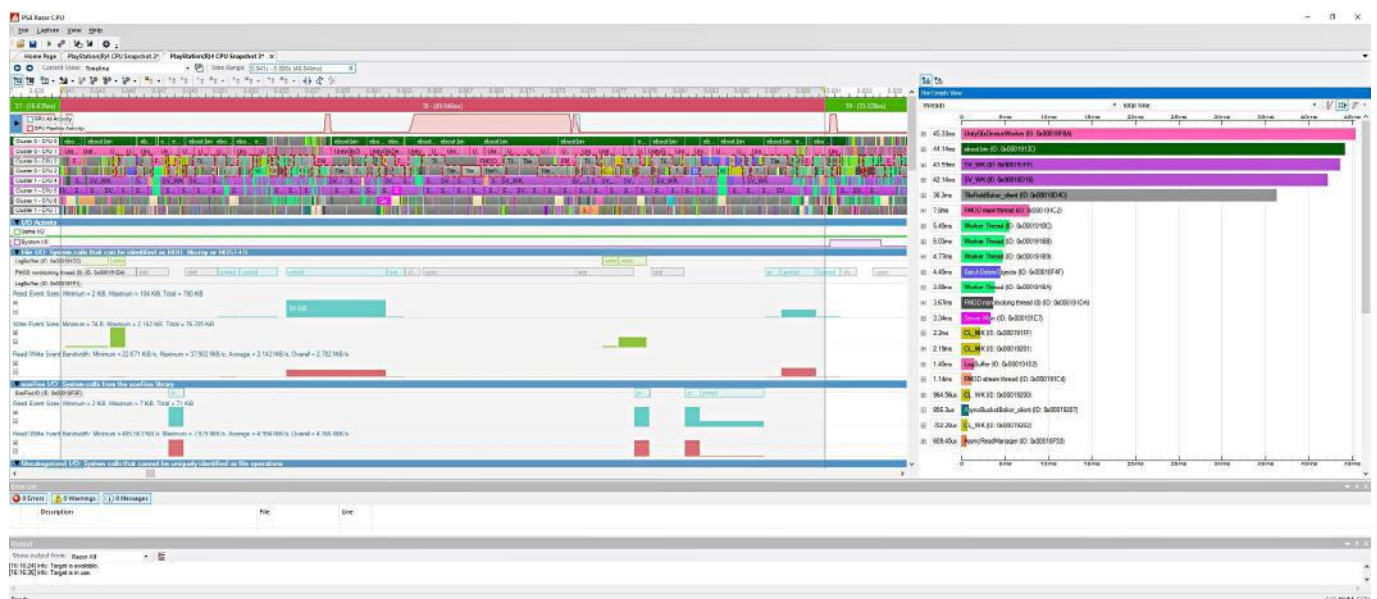


Bild 2: Realmforge versuchte, Operationen so gut wie möglich auf die Kerne des ersten Clusters zu verteilen, damit das System gar nicht erst auf die Idee kommt, den zweiten Cluster zu Rate zu ziehen.



Making Games 02/2016

ALS INDIE AUF DIE GROSSEN KONSOLEN

Black Forest Games beschreibt, wie sie »Giana Sisters« für PS4 und Xbox One veröffentlichten und welche Hürden es dabei zu meistern galt.



Mischa Strecker
ist Product Manager
bei Black Forest Games.

Er hat eine lange Karriere im Games-Journalismus und Produkt Management hinter sich, mit Stationen bei CyPress, Axel Springer, Yahoo und Web.de. Seit 2012 kümmert er sich bei Black Forest Games in Offenburg um PR, Marketing und Communities.



Neue Artworks oder Logos benötigt man in der Regel nicht für eine Konsolenversion ...



... außer man veröffentlicht das Spiel in China.

Kurz zurückgeschaut: 2012 starteten wir unser erstes Kickstarter-Crowdfunding. Ziel war es, Giana wieder zum Leben zu erwecken. Zunächst als PC-Spiel, aber es war immer als Fernziel auch ein Konsolen-

Release geplant. Kein Kinderspiel, wir haben zwar Erfahrung mit Konsolenspielen, hatten aber noch nie selbst eins veröffentlicht. Seit Anfang 2015 ist unser Mädchen nun auch auf Xbox One und PS4 unterwegs – und wir arbeiten dran, diesen Erfolg mit »Rogue Stormers« zu wiederholen.

Ein Blick auf Steam zeigt: Die Einstiegs-hürden sind niedrig. Auch kleine Teams mit wenig Erfahrung können dort relativ leicht veröffentlichen. Das war vor ein paar Jahren noch anders – Steam stellte Entwickler vor hohe Hürden und verlangte die Vermittlung eines Publishers. Inzwischen sind die Regeln locker, es gibt Green Light und Early Access, und der Katalog von Steam ist auf immense, unüberschaubare 30.000 Spiele und Apps angewachsen.

Zulassung als Entwickler

Auf den Konsolen sieht das anders aus, aber auch hier wurde der Zugang in den letzten Jahren deutlich erleichtert. Um überhaupt als Entwickler zugelassen zu werden, muss man sich bewerben. Bei allen drei Plattformen gehört dazu zunächst einmal ein Formular, das die Basics abfragt: Ist man eine eingetragene Firma, mit eigenen, zugangsgesicherten Büroräumen, einem sicheren Netzwerk, und gibt es schon ein Portfolio an veröffentlichten Spielen? Etablierte Studios mit Track-Record sind hier im Vorteil.

Sind diese Hürden genommen, ist man noch lange nicht am Ziel, aber darf sich immerhin schon mal »Zertifizierter Entwickler« nennen. Alle Hersteller bieten inzwischen kostengünstige Programme, um Indies mit den notwendigen Dev-Kits auszustatten. Microsoft überlässt Entwicklern über das ID@Xbox

Programm zwei Konsolen kostenlos, inklusive einer Unity-Lizenz. Sony und Nintendo verkaufen die Kits weiterhin zu Preisen um die 1.000 Euro (früher: ca. 3.000), bieten aber auch Leihprogramme für Indies an.

Zulassung des Games

Wenn die Firma die erste Hürde genommen hat, heißt das aber noch nicht, dass das eigene Spiel sie auch nimmt. Konsolen sind nach wie vor geschlossene Systeme – die Hersteller stellen Qualität vor Quantität. Um das zu gewährleisten, existieren klare Spielregeln.

Technical

Requirements Certification

Eine der schwierigsten Hürden, um ein Spiel auf eine Konsole zu bekommen, sind die Technical Requirements Certifications (TRC bei Sony / XR bei Microsoft). Jeder Plattforminhaber stellt diese in der ein oder anderen Form und unterschiedlichem Umfang zur Verfügung. Diese Dokumente geben technische – aber auch fürs Gameplay – Vorgaben, die ein Spiel erfüllen MUSS. Der Inhalt reicht über die Handhabung von Speicherständen, Vorgaben für Tastenbelegungen, über Spieler-Einladungen bis hin zu direktem Hardwarezugriff (Spoiler: Ist verboten!). Diese Dokumente gilt es zu studieren. Mehrmals. Akribisch. Unklarheiten – die es bei einigen Punkten gibt – lassen sich zwar in der Regel im direkten Kontakt mit den Platform Ownern klären, aber die Gespräche ersetzen nicht die Einarbeitung.

Flaschenhals Nr. 1 bei der Portierung oder Entwicklung eines Spiels ist und bleibt die Einhaltung der TRC. Unabhängig von der Plattform.

Vom PC zur Konsole – ein Spiel wird portiert

Das Submission-Verfahren der einzelnen Hersteller ist – im Vergleich zu früher – wesentlich einfacher, was sich vor allem zeitlich bemerkbar macht. Ein Beispiel: Der Port von Giana Sisters: Twisted Dreams auf die PS4 war innerhalb von zwei Monaten abgeschlossen.

Der Port auf die PS3 nahm knapp ein Jahr zuvor ein Vielfaches der Zeit in Anspruch.

Gerade Anfänger in der Konsolenentwicklung unterschätzen diesen Punkt immer wieder. Die Einhaltung der TRC kann einen immensen Arbeitsaufwand bedeuten. Das betrifft nicht nur offensichtliche Tasks wie die Optimierung der Rendering-Engine, die auf die jeweilige Konsole angepasst werden muss. Auch Controller Inputs, Standards in der Tastenbelegung oder das Benutzer-Interface müssen auf die jeweiligen Herstellerspezifikationen angepasst werden. Eigenheiten wie die Ausgabe auf Fernsehgeräten (Overscan, anyone?) wollen berücksichtigt und wie buchstäblich Hunderte weitere Details beachtet werden.

Man kann es nicht oft genug wiederholen: TRC sind komplex – und ohne Erfahrung kann es mehrere Monate dauern, bis man sein Projekt zertifiziert hat.

Insgesamt läuft der Submission-Prozess bei allen Herstellern nahezu identisch ab. Als Entwickler reichen wir ein Submission Package des Spiels ein, bekommen Feedback in Form eines Bug Reports (priorisiert nach Schwere und der Vorgabe oder Empfehlung eines Fix – zum Beispiel »must fix« (»mf«) für Fehler, die auf jeden Fall behoben werden müssen). Das korrigierte Paket reichen wir erneut ein und warten auf Feedback. Wash – Rinse – Repeat.

Unterschiede betreffen hauptsächlich die Vorgehensweise und das Toolset. Im Falle Sonys steht man als Entwickler vor einer weiteren Besonderheit: Man durchläuft den Submission-Prozess jeweils für jede Region, in der man veröffentlichen will, aufs Neue. Microsoft agiert hier »globaler«. Hat ein Spiel den Prozess erfolgreich durchlaufen, kann es weltweit veröffentlicht werden. Allen Herstellern gemein ist, dass man nicht davon ausgehen sollte, dass die eingereichten Versionen gleich intensiv getestet werden: Es kann zum Beispiel vorkommen, dass im ersten Feedback nur ein »mf« Bug auftaucht. Im nächsten Bug-Report stehen dann unerwartet acht neue. Damit muss man rechnen – in der Regel kann man es vermeiden, wenn man sich an die TRC-Dokumente hält.

Nicht unerwähnt sollte dabei bleiben, dass die QA der Plattforminhaber nicht die eigene QA ersetzen kann. Zwar sind die Submissions in der Regel kostenfrei, jedoch kann es passieren, dass man die Kosten trägt, wenn man das System missbraucht.

Dieser Prozess beginnt übrigens mit jedem neuen Patch von vorne.

Kommunikation: Wichtig! Hilft!

Wie oben erwähnt kümmern sich Microsoft, Sony und Nintendo inzwischen besser um die Entwickler – in der Regel bekommt man verschiedene Ansprechpartner für jeden Teilaspekt im Release-Prozess. Sie helfen bei Fragen in der Entwicklung, aber auch beim Marketing oder bei der Zusammenarbeit mit



Ziel der Kickstarter-Kampagne war es, Giana wieder zum Leben zu erwecken. Als Fernziel war immer ein Port von PC zu Konsole geplant.

den örtlichen Behörden in Ländern, in denen man ein Spiel veröffentlichen will. An dieser Stelle Kudos an SCE Asia. Ohne die intensive Hilfe von Liou, unserem Ansprechpartner vor Ort, der die Kommunikation mit den Behörden übernahm und uns die Auseinandersetzung mit der Han-Schrift ersparte, hätten wir Giana Sisters: Twisted Dreams niemals in China veröffentlichen können.

Warum der Stress?

Neben dem technischen Aspekt kommen noch weitere Aufgaben auf Indies zu: Fragen zu Altersfreigaben, Marketing und Lokalisierungen. Community-Arbeit. Kommunikation mit der Presse. Messebesuche, um das Spiel zu präsentieren. Wenn man es sich leisten kann, die Arbeit mit Agenturen. Die Wahrheit ist: Für jedes »DayZ«, »Risk of Rain«, »Speedrunners« oder »Minecraft« gibt es zig andere Spiele, die leer ausgehen. Technisch ist self-publishing möglich – inzwischen sogar plattform-unabhängig. Es stimmt auch, dass Indie-Titel beliebt sind. Allerdings ist nicht einmal harte Arbeit und die Gewissheit, »alles richtig gemacht zu haben«, ein Garant für den Erfolg.

Wenn also der Aufwand – und Risiko – so hoch sind und das Abenteuer am Ende zum Reifall werden kann, wieso machen wir uns dann den Stress überhaupt?

Für uns kann ich nur sagen: Weil es sich eben doch lohnen kann, und weil wir die Arbeit lieben. Technisch gesehen hat sich das Portieren der Games durch die PC-nahe Architektur der aktuellen Konsolengeneration wesentlich vereinfacht. Die Konsolenhersteller gehen zum Teil aktiv auf Indies zu und haben die Einstiegshürden massiv gesenkt. Und: Die Plattformen sind noch nicht so überlaufen wie zum Beispiel Steam. Spiele auf dem Xbox Marketplace, dem PlayStation Store oder dem Nintendo eShop haben eine deutlich höhere Sichtbarkeit und dadurch bessere Chancen, verkauft zu werden. Außerdem erweitern wir dadurch natürlich auch die Zahl der potenziellen Käufer und erreichen andere Käuferschichten.

Zu guter Letzt bringt es uns als Entwickler weiter, wenn wir uns möglichst breit aufstellen: sowohl für Auftragsarbeiten als auch für unsere Eigenproduktionen. **Mischa Strecker**

国家新闻出版广电总局文件

新广出审〔2015〕1081号

关于同意出版运营进口游戏机游戏 《吉娜姐妹 扭曲梦境 导演剪辑版》的批复

上海市新闻出版局:

你局沪新出科数〔2015〕508号文收悉。

经研究,同意华东师范大学电子音像出版社有限公司出版、上海东方明珠文化发展有限公司运营进口游戏机(PSP)游戏《吉娜姐妹 扭曲梦境 导演剪辑版》(网络下载)。请要求出版运营单位做好以下工作:

- 一、认真做好内容自查工作,不得增加国家法律和法规规定的禁载内容。
- 二、在游戏开始的面面显著位置标明该游戏的批准文号、游戏出版号、著作权合同登记号。

— 1 —

三、在游戏开始的画面中全文登载《健康游戏忠告》。

四、安装不良内容过滤软件,确保不出现违禁内容和违禁情况。

五、与著作权有关的事宜,应依照《著作权法》及国家有关法规执行。

六、督促出版单位收到批复后,在15个工作日内将其为该游戏分配的游戏出版号信息报新闻出版广电总局数字出版司。



抄送:华东师范大学电子音像出版社有限公司,上海东方明珠文化发展有限公司。

国家新闻出版广电总局办公厅 2015年9月24日印发

— 2 —

Am Ende eines langen bürokratischen Prozesses steht das offizielle Dokument der chinesischen Zensurbehörde, mit dem die Veröffentlichung gestattet wird.



MULTIPLATTFORM-ENTWICKLUNG MIT UNITY

Das Team von KING Art Games hat sein Fantasy-Adventure »The Book of Unwritten Tales 2« auf insgesamt zehn Plattformen veröffentlicht. Was noch vor wenigen Jahren für ein kleines Studio undenkbar gewesen wäre, ist heute dank Multiplattform-Engines wie Unity möglich – wenn man die Produktion entsprechend organisiert und potenzielle Probleme kennt.



Jan Theysen
ist Game Director
bei KING Art Games.

Jan ist Game Director sowie Autor von »The Book of Unwritten Tales 2« und hat an den verschiedenen Benutzeroberflächen und Steuerungssystemen gearbeitet. Er spielt von PC über Konsole bis hin zu Mobile auf allen Plattformen, die er in die Finger bekommen kann.



Arne von Ohsen
ist Programmierer
bei KING Art Games.

Arne ist mit seinem Team verantwortlich für die Portierung von »The Book of Unwritten Tales 2« auf zehn Plattformen. Er hat dazu unzählige Versionen gebaut und Terrabytes an Daten quer durch die Welt geschickt. Zurzeit entwickelt er KING Arts Multiplattform-Toolkit.

Wenn man ein Spiel nur auf einer Plattform herausbringt, müssen sämtliche Entwicklungskosten, all das Geld für Assets, Animationen, Sprachaufnahmen und Werbung, mit dieser einen Version des Spiels eingespielt werden.

Wir haben »The Book of Unwritten Tales 2« (BoUT 2) für PC, Mac, Linux, Xbox 360, PS3, Wii U, Xbox One, PS4, Android und iOS (Tablets & Smartphones) veröffentlicht. Das sind zehn Möglichkeiten, vorhandene Fans auf ihrer aktuell bevorzugten Plattform anzusprechen, neuen Spielern das Spiel vorzustellen und Berichte in der Presse zu generieren.

Die neun Portierungen des Spiels haben die Entwicklungskosten um weniger als 15 Prozent steigen lassen, denn:

- Wir haben die Assets von Anfang an so angelegt, dass sie auf allen Systemen benutzt werden können.
- Wir haben bei Features, Steuerung und UI stets alle Plattformen im Hinterkopf gehabt.

- Wir haben auf plattformunabhängige Technologien/Formate und mit Unity auf eine bewährte Multiplattform-Engine gesetzt.

Dabei hat uns geholfen, dass wir zuvor bereits andere Projekte für Last-Gen-Konsolen und iOS/Android umgesetzt haben. Die (teilweise schmerzhaften) Erfahrungen aus diesen Projekten konnten wir bei BoUT 2 nutzen.

Nicht jedes Spiel eignet sich für jede Plattform

Natürlich sind zehn Plattformen für ein Spiel eine Ausnahme. Die Veröffentlichung von BoUT 2 fiel mit dem Konsolen-Generationenwechsel zusammen und nur wenige Spiele eignen sich so gut für einen Multiplattform-Release wie Point&Click-Adventures.

- Es muss möglich sein, das Spiel mit Maus, mit Controller und per Touchsteuerung gleichermaßen gut zu spielen.
- Das Spiel muss auf einer großen Bandbreite von Geräten (von zehn Jahre alten Xbox-360-Konsolen über schwachbrüstige Handys

bis hin zu aktuellen PCs) lauffähig sein und dabei gut aussehen.

- Die Benutzeroberfläche muss auf einem Fernseher an der Wand genauso funktionieren wie auf einem winzigen Display in der Hand des Spielers.
- Es muss auf jeder der Plattformen eine Zielgruppe für das Spiel geben.

Die Portierung eines Spiels muss nicht teuer sein. Trotzdem sollte man der Versuchung widerstehen, ein Game auf jede verfügbare Plattform zu schaufeln, nur weil man dazu in der Lage ist. Wenn sich ein Spiel zum Beispiel einfach nicht für Touch Devices eignet, sollte man seinen Fans diese schlechte Gaming-Erfahrung ersparen und darauf vertrauen, dass sie stattdessen eine der anderen Spielversionen kaufen.

Man sollte sich außerdem vom Gedanken des »kleinsten gemeinsamen Nenners« fernhalten. Ein Spiel, das auf PC oder den Konsolen schlecht aussieht, nur damit es auch auf Mobile Devices läuft, wird von den Spielern zu Recht nicht akzeptiert werden. Gleiches gilt für »vorportierte« Spiele, die sich anfühlen, als seien sie auf keinem System zu Hause, weil der Entwickler ständig nur darüber nachgedacht hat, wie er den Aufwand bei der Portierung gering halten kann.

Drei Steuerungssysteme

Wir alle haben uns schon mit schlecht portierten Controller-Steuerungen und Benutzeroberflächen auf dem PC herumschlagen müssen. Wir empfehlen daher jedem, der ein Multiplattformspiel machen will, von Anfang an **drei** individuelle Steuerungssysteme und **drei** individuelle Benutzeroberflächen einzuplanen. Es gibt keine Steuerung oder UI, die sich bei allen drei Eingabemöglichkeiten auf allen drei Geräteklassen optimal anfühlt.

Der Aufwand für individuelle Steuerungen und Benutzeroberflächen hält sich in Grenzen. Es verblüfft uns immer wieder, warum ausgerechnet an dieser Stelle bei vielen AAA-Games gespart wird. Schlechter Bedienkomfort oder umständliche Benutzeroberflächen können Spiele schließlich kaputt machen!

Über Folgendes sollte man sich Gedanken machen:

- **Was sind die Standards für Games auf den verschiedenen Plattformen?** Wie werden »Rechtsklicks« oder Tooltips auf Tablets gemacht? Oder wie verhält sich ein Controller-gesteuerter Charakter, wenn er links aus einer Szene geht und die nächste Szene von rechts betritt (und der Spieler immer noch nach links drückt)?
- **Welche technischen Auswirkungen haben die verschiedenen Steuerungen?** Wenn sich ein Charakter auf dem PC bei einem Mausklick schön weich umdrehen soll, derselbe Charakter bei Controller-Steuerung aber direkt auf die Eingaben des Spielers reagieren muss, um

sich nicht »schwammig« anzufühlen – was bedeutet das für das Animationssystem?

- **Was bedeutet das für Gameplay und für Leveldesign?** Wenn man ein Item mit dicken Fingern auf einem iPhone-Screen auswählen soll, wie groß muss dann der Klickbereich mindestens sein? Und ist er für Maussteuerung auf dem PC dann zu groß? Wie viel Platz muss es zwischen Kollisionsobjekten mindestens geben, damit man per Controller-Steuerung bequem hindurchnavigieren kann?
- **Gibt es gerätespezifische Eingabemöglichkeiten, die man nutzen will oder muss?** Macht es vielleicht Sinn, die Bewegungssensoren oder das Licht im PS4-Controller zu benutzen? Oder will man Sprachsteuerung bei der Xbox One unterstützen? Der



Wir haben bei der BoUT-2-Entwicklung Konsolen und Tablets immer im Hinterkopf gehabt, haben aber dennoch zunächst die PC-, Mac- und Linux-Versionen fertiggestellt und erst dann ernsthaft mit den Konvertierungen begonnen.



Weil es auf Tablets und Handys keinen Rechtsklick gibt, haben wir für diese Systeme ein Kontextmenü ins Spiel integriert. Außerdem haben wir fürs Inventar eine Drag&Drop-Funktionalität eingebaut, da iOS- und Android-Spieler das so gewohnt sind.

»Back«-Button auf Android-Geräten sollte irgendwas Sinnvolles im Spiel tun ...

Es ist nötig, sich mit diesen Dingen früh in der Entwicklung zu beschäftigen, idealerweise schon in der Preproduction. Man muss nicht zwangsweise alle Steuerungsmöglichkeiten von Anfang an umsetzen, aber man muss sie im Hinterkopf haben, um zukünftige Probleme vermeiden zu können.

Welche Steuerung dann zum Beispiel mit Controllern am besten funktioniert, kann man eigentlich nur durch Tests herausfinden. Man drückt unterschiedlich erfahrenen Spielern den Controller in die Hand und beobachtet, was sie anstellen. Feedback wie »Wenn ich das mache, hätte ich eigentlich erwartet, dass die Spielfigur Folgendes tut«, ist Gold wert.

Drei Benutzeroberflächen

Man sollte sich und den Spielern grundsätzlich einen Gefallen tun und skalierbare Benutzeroberflächen und Schriftarten in seine Spiele implementieren.

Texte, die auf PC eine angenehme Größe haben, müssen für Fernseher an der Wand auf 150 bis 200 Prozent vergrößert werden. Auch für Mobile Devices sind größere (Per-Pixel-) Schriften aufgrund der unglaublich hohen Auflösungen nötig.

Auf Touchgeräten müssen Buttons und sonstige Klickbereiche deutlich größer sein als zum Beispiel auf dem PC. Außerdem muss man sich Lösungen für Mouse-Overs/

Tooltips und Rechtsklicks einfallen lassen, die zwar bei Maussteuerung Standard, bei Mobile-Anwendungen aber tricky sind.

Bei einer Konsolen-UI passen nicht nur viel weniger Elemente in einen Schirm, auch die Menüführung ist häufig grundsätzlich anders als auf dem PC. Auch hier muss man sich intensiv mit weiteren Genvertretern auseinandersetzen und die Standards kennen.

Einige plattformspezifische Elemente, wie zum Beispiel Controller-Tastensymbole, sollten leicht austauschbar gehalten werden und man sollte sich mit den Vorgaben der Plattformanbieter beschäftigen. Es gibt Regeln, wie die Kreis-, Quadrat-, Dreieck- und X-Buttons auszusehen haben – Verzeihung, ich meinte »Kreuz« und »Tasten«, denn so sind die bevorzugten Bezeichnungen.

Bis auf die verschiedenen Skalierungen und kleinere Änderungen sind sich die drei Benutzeroberflächen bei BoUT 2 relativ ähnlich. Das liegt aber vor allem daran, dass Point&Click-Adventures sehr simple Spiele sind, wenn es ums User Interface geht. Bei Spielen aus anderen Genres sollte man den Aufwand für nötige Anpassungen nicht unterschätzen.

Qualität vs. Performance

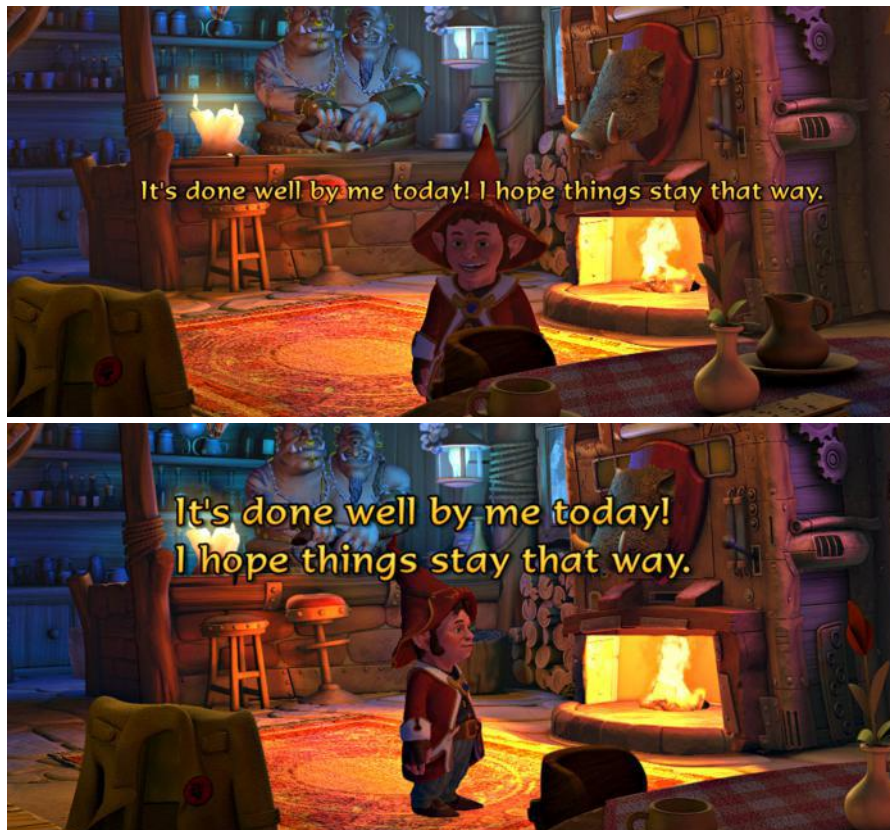
Aktuell sind normale Smartphones und Tablets qualitativ höchstens auf dem Niveau der letzten Konsolengeneration. Zwar haben sie mehr Arbeitsspeicher, dafür sind die Grafikchips in der Regel schwächer und haben deutlich mehr Pixel auf den Schirm zu bringen als die Xbox 360 oder PS3.

Die Entwicklung schreitet im Mobile-Bereich schnell voran. Es ist jedoch nicht damit zu rechnen, dass die Grafikqualität in den nächsten Jahren das Niveau der aktuellen Konsolengeneration erreicht.

Wenn man seinen Spielern auf allen Systemen die gleiche, hochwertige Optik anbieten möchte, dann darf man also nicht auf rohe Rechenpower setzen, sondern muss geschickt vorgehen. Spiele wie »Don't Starve« oder »FTL« leben von ihrem Stil und sehen auf jeder Plattform gleich aus – natürlich, denn technisch gesehen verlangen sie den Systemen wenig ab.

Bei BoUT 2 sind wir einen ähnlichen Weg gegangen und haben auf eine Technik namens »Projection Mapping« gesetzt. Dabei wird vorgerenderte und vorbeleuchtete Highpoly-Geometrie auf Echtzeit-Lowpoly-Geometrie gemappt. Das Ergebnis sind sehr detaillierte Szenen mit schöner Tiefenwirkung, die aber nur aus wenigen tausend vorbeleuchteten Polygonen bestehen (verglichen mit hunderttausenden dynamisch beleuchteten Polys, die für klassische Lowpoly-Versionen der Szenen nötig gewesen wären).

Die in der Berechnung der Hintergründe eingesparte Performance konnten wir in die Charaktere, deren Echtzeit-Beleuchtung, in Effekte und Schatten investieren. Der Aufwand hat sich gelohnt, sieht BoUT 2 doch auf



Textgrößen unterscheiden sich auf PC (oben) und Konsolen (unten) erheblich. Das wird vor allem relevant, wenn der Platz stark begrenzt ist.

allen Plattformen praktisch identisch (und wir hoffen: gut) aus.

Herausforderungen bei der Code-Portierung

Wirft man einen Blick auf die Multiplattform-Feature-Liste von Unity, entsteht schnell der Eindruck, alle angebotenen Plattformen seien nur einen Mausklick entfernt. Es stimmt zwar, dass Unity einem viel Arbeit abnimmt, jede neue Plattform erfordert aber auch immer mehr oder weniger große Anpassungen am Spiel selbst.

Auf den Desktop-Plattformen verwendet Unity die Mono Runtime für die vom Entwickler geschriebenen Scripts. Mono ist annähernd identisch zu Microsofts .NET Framework und bietet daher Entwicklern einen C# Compiler mit Unterstützung für viele komfortable Features wie etwa LINQ. Kern einiger dieser Features ist die Just-in-Time-Kompilierung (JIT), bei der zur Laufzeit neuer Code generiert wird.

Auf Plattformen, wo dies nicht unterstützt wird (beispielsweise iOS), wird der Code von Unity beim Bauen der Anwendung »ahead of time« (AOT) konvertiert. Wird in einer AOT-Umgebung ein JIT-Feature verwendet, führt dies zu einem Laufzeitfehler, der dann nicht selten zum Absturz des gesamten Spiels führt.

<code>var lastItem = list.Last();</code>	//will result (in some cases) in a crash
<code>var lastItem = list[items.Count - 1];</code>	//will run in all environments

Diese Fehler zu finden, kann schnell sehr zeit- und aufwändig werden, weil sie in einem Desktop-Build oder im Unity Editor nicht auftreten. Das Spiel muss also auf die Zielplattform gebracht werden, was je nach Umfang des Projekts sehr lange dauern kann. Es zahlt sich daher schnell aus, diese Arbeitsschritte zu automatisieren und zu beschleunigen, indem man beispielsweise ein kleines Testprojekt ohne große Assets verwendet.

Es gibt im Internet reichhaltige Non-NDA-Dokumente über Dos und Don'ts in Bezug auf

Dev-Kits & Ressourcen

Um auf den Konsolen arbeiten zu können, benötigt man spezielle Dev-Kits von Sony, Microsoft und Nintendo. Die Chancen, als professioneller Entwickler ein Dev-Kit zu erhalten, sind nicht schlecht. Allerdings sollte man Zeit mitbringen. Einige Ansprechpartner bei den großen Drei sind schlecht zu erreichen, und teilweise können zwischen der ersten Kontaktaufnahme und der Lieferung des Dev-Kits Monate liegen.

Als registrierter Developer für eine Konsole erhält man gegen NDA Zugang zu Entwicklungsressourcen, Dokumentationen

und Tools. Nichts davon dürfen wir hier beschreiben oder zeigen. Der NDA ist auch der Grund, warum man wenig öffentlich zugängliche Informationen finden kann, wenn man in Probleme läuft. Eine schnelle Google-Suche ist bei konsolenspezifischen Problemen normalerweise nicht möglich.

Unity bietet für registrierte Entwickler geschlossene Foren, in denen sie sich austauschen können. Das Niveau der Diskussionen ist in diesen Foren professioneller als in den öffentlichen Foren, weil in der Regel nur Profis an Dev-Kits kommen.

AOT. Es empfiehlt sich, die Problematik während der Entwicklung ständig im Blick zu haben.

Asynchronität und Fehlerbehandlung

Wenn man auf Desktop-Plattformen ein Savegame-System implementiert, das nur wenige Kilobyte an Daten schreiben muss, ist es am schnellsten (und bequemsten), dies in einer blockenden Operation zu tun. Das Schreiben so kleiner Dateien ist normalerweise kaum

bemerkbar. Leider führt eine solche Lösung auf vielen Plattformen zu Problemen.

Sicherer ist es, von Folgendem auszugehen:

- Alle Schreib-, Lese- und Löschvorgänge können im schlimmsten Fall Sekunden dauern.
- Diese Operationen sind nicht immer erfolgreich und der Benutzer muss dann aussagekräftige Fehlerinformationen erhalten.

Implementiert man sein Savegame-System so, dass alle Vorgänge asynchron ablaufen und im Zweifelsfall mit vernünftigen Fehlermeldun-



Bei »Battle Worlds: Kronos« mussten wir eine optisch abgespeckte Mobile-Version erstellen, weil wir uns im Vorfeld nicht genug Gedanken gemacht hatten – die Version kam erst als Kickstarter Stretch Goal dazu.

Eine neue Plattform – Die ersten Schritte für Programmierer

1. Dokumentation lesen und sich mit dem Submission-Prozess vertraut machen:
 - Was sind die technischen Anforderungen der Zielplattform an mein Spiel?
 - Wie läuft der Zertifizierungsprozess ab und was wird dabei wie getestet?
 - Wie muss ich mein Spiel für die Abgabe verpacken?
 - Gibt es dabei Vorgaben und Limitierungen?
2. Sich mit den Dev-Kits vertraut machen und ein Testprojekt bauen:
 - Was muss ich tun, damit ich ein Unity-Projekt auf der Konsole zum Laufen bekomme?
 - Was kann ich einstellen und wie wirkt es sich aus?
 - Wie erhalte ich Fehlerinformationen?
3. Das Zielspiel zum Laufen bekommen:
 - Gibt es Probleme mit meinem Spiel, wenn es AOT-kompiliert wird?
 - Gibt es Performance- oder Speicherprobleme?
 - Gibt es Shader oder Effekte, die auf der Zielplattform nicht funktionieren?
4. Buildserver aufsetzen, automatisierte Tests (Benchmark-System) bauen.

gen enden, ist das die optimale Voraussetzung für eine schmerzfreie Portierung.

Eine asynchrone Architektur beschränkt sich jedoch nicht nur auf das Savegame-System. Überschreitet das Spiel eine gewisse Datengröße, kann es erforderlich sein, das Spiel in mehrere Einzelteile aufzusplitten. Diese Teile werden installiert, während das Spiel schon läuft. Es kann also passieren, dass zum Beispiel ein Level, den der Spieler gerade betreten will, noch nicht installiert wurde. In diesem Fall muss der Spieler benachrichtigt werden und das Spiel muss auf die Installation des Levels warten.

Zusätzlich können Ereignisse eintreten, die mit dem eigentlichen Spiel nichts zu tun haben, auf die das Spiel aber jederzeit reagieren muss. So kann sich beispielsweise auf der Xbox One der aktuelle Benutzer stets von der Konsole abmelden. Optimalerweise wird in diesem Moment das Spiel angehalten und eine Nachricht angezeigt.

Einbinden der Plattform-APIs

Plattformspezifische Features wie Cloud Savegames, Achievements oder Ingame-Käufe machen die Verwendung der entsprechenden Plattform-API erforderlich. Diese APIs sind in der Regel in der nativen Plattformsprache geschrieben und können deshalb nicht ohne Weiteres mit Mono in Unity verwendet werden.

Selbstverständlich gibt es für die meisten dieser Funktionen fertige Mono-Implementationen – einige davon sind direkt in Unity integriert, andere können über Plugins dazugeladen werden. Wenn man von den Standards abweicht und exotische Features verwenden möchte, kann es erforderlich sein, ein eigenes, natives Plugin zu schreiben.

Der eigene Code, der die Plattform-API aufruft, muss für jede Plattform neu implementiert werden. Es ist also empfehlenswert, die API an einer zentralen, austauschbaren Stelle aufzurufen und diese Calls nicht im gesamten Code zu verteilen.

Speicheroptimierung

Speicheroptimierung spielt auf den Current-Gen-Konsolen eine geringere Rolle als auf den Last-Gen-Systemen. Trotzdem lohnt es sich, Zeit in die Speicheroptimierung zu stecken, denn von einem geringeren Speicherverbrauch und kürzeren Ladezeiten profitieren alle Versionen des Spiels.

- Alle Assets, die in einer Unity-Szene vertreten sind, werden beim Laden dieser Szene in den Laufzeitspeicher geschrieben. Man sollte also sicherstellen, dass nur genau die Assets in der Szene sind, die auch tatsächlich benötigt werden. Gesprochene Texte, Musik usw. sollten nach Möglichkeit »on the fly« nachgeladen – und dann auch wieder entladen werden.
- Die Import-Einstellungen aller Assets müssen korrekt sein. Um The Book of Unwritten Tales 2 auf den Last-Gen-Konsolen lauffähig



PlayStation 4 und Xbox One fordern, dass große Spiele schon gestartet werden können, während das Spiel noch heruntergeladen/installiert wird. Entweder hat man das schon zu Beginn der Entwicklung auf dem Schirm oder man hat am Ende der Entwicklung ein großes Problem ...

zu bekommen, war es zum Beispiel notwendig, die Texturauflösungen aller Hintergrundtexturen zu halbieren. Eine maximale Größe kann pro Plattform vergeben werden.

- Codestripping spart Speicher und beschleunigt die Ladezeiten des Spiels. Ist dieses Unity-Feature aktiviert, wird der Code während des Buildens analysiert und Klassen und Methoden, die nicht referenziert wurden, werden entfernt. Ein dynamischer Aufruf von entfernten Methoden (beispielsweise durch Reflection) führt allerdings nach dem Codestripping zu einem Laufzeitfehler. Deshalb ist es notwendig, diese Klassen und Methoden vor einem Build in einer Blacklist festzuhalten.
- Werden in kurzen Abständen viele neue Objekte instanziiert (zum Beispiel durch das Instanzieren von Objekten in einem Update Call), wächst der Mono Heap. Der Heap hat eine feste Größe im Hauptspeicher. Überschreitet die Größe des Heaps sein aktuell zugewiesenes Budget und kann der Garbage Collector nicht genug Speicher frei machen, wird neuer Speicher vom Hauptspeicher alloziert. Die Größe des Heap Budgets kann in Unity nur wachsen. Es gibt keine Möglichkeit, einmal zugewiesenen Heap-Speicher wieder an das System zurückzugeben. Dies kann zu schwer diagnostizierbaren Problemen führen, die erst nach langer Spieldauer auftreten.

Performanceoptimierung

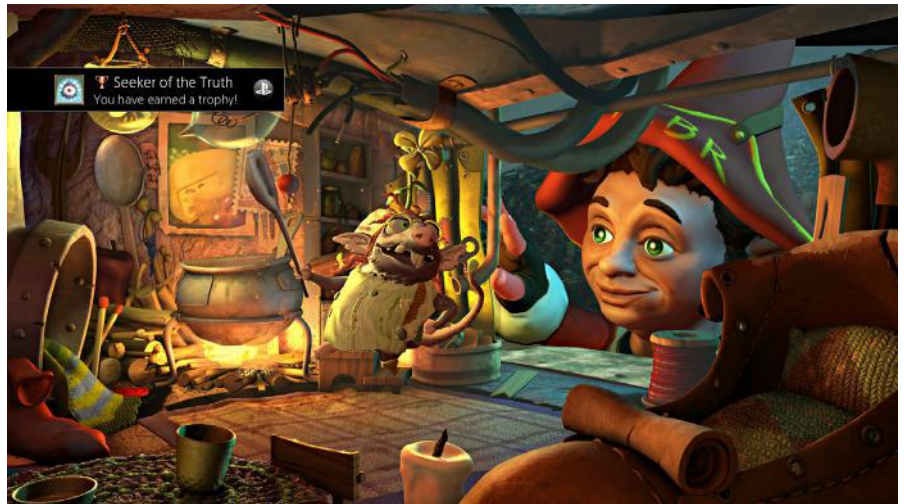
In The Book of Unwritten Tales 2 bieten wir vier Qualitätsstufen an. Auf Plattformen mit vorbestimmter Hardware ist das Spiel fest auf eine der Qualitätsstufen eingestellt. Bei Plattformen mit unterschiedlicher Hardware, wie dem PC oder Android Devices, kann sich der Benutzer für eine Qualitätsstufe entscheiden.

Abhängig von der eingestellten Qualität werden dann global Posteffekte wie Tiefenunschärfe oder Antialias deaktiviert oder reduziert. Außerdem finden je nach Qualität unterschiedliche Shader für unsere Hintergründe oder Charaktere Anwendung.

Wir haben zusätzlich die Möglichkeit geschaffen, Objekte, Effekte und Shader für eine Qualitätsstufe zu taggen. So können je nach Qualitätsstufe höherwertige Partikeleffekte oder detailreichere Objekte verwendet werden.

Die letzte Möglichkeit, bessere Performance auf schlechter Hardware zu erreichen, ist, die Auflösung des Spiels zu reduzieren. Dieser Schritt führt gerade auf mobilen Geräten zu enormen Verbesserungen, reduziert aber auch die Qualität des Spiels deutlich.

Der Aufwand für die Performance- und Speicheroptimierung der aktuellen Konsolengeneration hat sich bei BoUT 2 sehr in Grenzen gehalten. Weil wir mit der PC-Version bereits eine High-End-Variante und mit der Xbox-360-Version eine Low-End-Variante hatten, mussten wir lediglich optimale Einstellungen für die PS4 und Xbox One finden. Unser auto-



Die Spiellogik ruft eine »Grant Achievement«-Methode auf, wenn der Spieler ein Achievement erspielt hat. Die Methode informiert dann die Achievement-Systeme von Steam, Xbox 360, Xbox One, PlayStation 3, PlayStation 4 und iOS-Game-Center.



BoUT 2 lädt Dialoge, Animationen, Soundeffekte und Musik erst dann, wenn sie wirklich benötigt werden. Eine Optimierung, die für die speicherarmen Last-Gen-Konsolen nötig war, die aber auch auf PC und den aktuellen Konsolen Ladezeit spart.

matisierter Benchmark (siehe Kasten) hat uns dabei sehr gute Dienste geleistet.

Konsolen-Submissions

Ein PC-/Mac-/Linux-Spiel ist dann »fertig«, wenn Entwickler und/oder Publisher sagen, es sei fertig. Ein Disc Image wird ins Presswerk geschickt, eine Version bei Steam hochgeladen – und das war's. Bei Konsolenspielen sieht das anders aus. Die werden vom Plattformhalter zunächst in einem Zertifikationsprozess genau unter die Lupe genommen.

Welche Tests durchgeführt werden und wie diese im Detail ablaufen, ist für registrierte Entwickler detailliert dokumentiert. Abhängig davon, ob das Spiel Online-/Multiplayer-Funktionalität besitzt, gibt es über hundert Test-Cases, die alle umgesetzt und getestet werden müssen.

Man sollte sich mit dem Zertifikationsprozess vertraut machen und genau wissen, was gefordert wird, sowohl im Spiel als auch in den Begleitdokumenten für die Submission. Im Kern ist die Absicht der Plattformanbieter, eine



Current-Gen-Konsolen: Hochaufgelöste Texturen, weiche Schatten, Antialiasing, Licht- und sonstige Effekte, Echtzeit-Spiegelungen.



Niedrigste einstellbare Qualitätsstufe für Mobile: 50 Prozent Texturgröße, Mobile-Komprimierung für Texturen, kein Antialiasing, keine Schatten, vereinfachte Licht- und sonstige Effekte, vereinfachte Shader.

gleich bleibende Qualität und Bedienbarkeit der Konsolenspiele sicherzustellen.

Je nach Auslastung des Zertifikationsteams kann eine Submission einige Wochen dauern. Werden Fehler gefunden, muss der Zertifikationsprozess von Neuem komplett durchlaufen werden. Es sollte also genug Zeit für diesen Prozess eingeplant werden, und es ist nicht unrealistisch, von mindestens einem Fehlschlag auszugehen.

Bei der Konsolenentwicklung von *The Book of Unwritten Tales 2* hat sich der Einsatz einer externen QA sehr ausgezahlt. Es gibt diverse Firmen, die bei den Konsolenherstellern registriert sind und sich darauf spezialisiert haben, Spiele auf die Anforderungen des Zertifikationsprozesses hin zu testen.

Verwaltungsaufwand nicht unterschätzen

Auch wenn der technische Aufwand für eine Portierung eines Spiels auf eine neue Plattform heute geringer ist als jemals zuvor, darf

man den erheblichen Verwaltungsaufwand nicht unterschätzen.

Ähnlich wie jede weitere Sprachversion oder Spieledition zusätzlichen Organisationsaufwand bedeutet, macht auch jede zusätzliche Plattform Arbeit.

- Der Testing-Aufwand des Spiels vervielfacht sich. Will man alle Versionen eines Spiels parallel rausbringen, braucht man plötzlich mehrere Male so viele Tester.
- Spielversionen und Patches müssen unter Umständen für zig verschiedene Spielversionen gleichzeitig gebuildet und getestet werden – und wie immer am besten bis gestern!
- Zig Gigabytes an Daten müssen hochgeladen werden. Klingt banal, aber wenn man 60 GB an Builds über die Nacht hochladen will, hat man besser eine schnelle und stabile Leitung oder kann den Upload auf verschiedene Standorte verteilen.
- Man muss mit etlichen Parteien kommunizieren. Mit den Plattform-Haltern (im Falle

KING Art Games

KING Art Games gehört mit 15 Jahren zu den alteingesessenen Spieleentwicklern in Deutschland und arbeitet mit seinen rund 35 Entwicklern an PC-, Konsolen-, Mobile und Browsergames. »The Book of Unwritten Tales 2« ist das fünfte Adventure der Bremer und erhielt 2015 den Deutschen Entwicklerpreis als »Bestes Konsolenspiel«.



Jede Version von BoUT 2 haben wir mehrfach beim QA-Dienstleister Quantic Lab testen lassen, insgesamt über 20 Testrunden (verglichen mit zwei bis drei, die wir bei einem PC-only-Titel veranschlagen würden).

Der Lebensretter: Automatisierte Benchmarks

Unity erlaubt es, schnell und fast ohne Vorkenntnisse gut aussehende Szenen zu erstellen. Was für Grafiker großartig ist, ist für Coder, die Speicherverbrauch und Performance im Auge behalten müssen, häufig ein Problem. Regelmäßig werden zu »teure« Effekte oder Objekte eingebaut, performancekritische Dinge falsch eingestellt und Features unbemerkt kaputt gemacht.

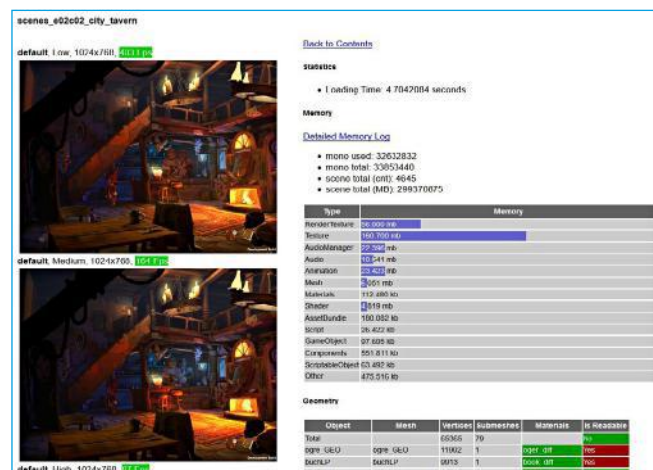
Damit wir in der Entwicklung den aktuellen Speicherverbrauch und die Performance jederzeit im Auge behalten können, haben wir unser Spiel mit einem automatischen Benchmark-Modus versehen.

Wird das Spiel mit dem entsprechenden Commandline-Argument gestartet, beginnt es, alle Szenen, die im Spiel vorkommen können, nacheinander zu laden. Jede Szene wird in allen Qualitätsstufen dargestellt und es wird je ein Screenshot gemacht. Fehlende Texturen, kaputte Shader oder Beleuchtungsprobleme lassen sich so schnell entdecken.

Außerdem wird die Framerate festgehalten und der Speicherverbrauch der Szene analysiert. Dazu werden zunächst über »Resources.FindObjectsOfType(typeof(UnityEngine.Object));« alle Objekte erfragt, die sich aktuell im Unity-Speicher befinden, und über »Profiler.GetRuntimeMemorySize(object);« gemessen.

Alle so erfassten Daten werden dann in einem für alle Teammitglieder zugänglichen HTML-Report zusammengefasst. Der Benchmark wird auf einem dedizierten Rechner, der grob die Mindestanforderungen des Spiels erfüllt, nach jedem erfolgreichen Build des Projekts automatisch gestartet. Wir haben so jederzeit einen detaillierten und maximal 1 bis 2 Stunden alten Bericht über den Speicherverbrauch und die Performance unseres Spiels.

Sehr hilfreich war der Benchmark auch beim Hardware-Kompatibilitätstest auf Android-Devices. Er wurde auf Dutzenden Geräten parallel ausgeführt und wir erhielten mit wenig zeitlichem Aufwand detailliertes Feedback, welche Szenen auf welchen Systemen Probleme machten.



Der komfortable HTML-Report unseres automatischen Benchmarks: Pink eingefärbte Objekte zeigen ein Shader-Problem an. Gerade auf Mobile Devices gibt es viele Grafikchips, die aus unterschiedlichen Gründen nicht mit jedem Shader klarkommen.

von Sony sogar mit Europa und Nordamerika separat), mit digitalen Verkaufsplattformen, mit Unity, mit dem Publisher, mit den Testern – jede zusätzliche Plattform bringt neuen Klärungsbedarf mit sich. Und weil die Partner häufig in unterschiedlichen Zeitzeonen sitzen, darf man nicht jederzeit mit einer sofortigen Antwort rechnen.

- Mehr Plattformen bedeuten auch mehr PR-Aufwand, mehr Werbematerial und später mehr Spieler mit plattformspezifischen Problemen oder Fragen ...

einem Plattform-Toolkit zusammengefasst. Dieses Toolkit kann in ein Unity-Spiel eingebunden werden und stellt die Schnittstelle zu den diversen Zielplattformen dar. Es nimmt Mitarbeitern mit wenig Erfahrung auf den verschiedenen Plattformen eine Menge Sorgen ab und verringert das Bug-Potenzial.

Wir werden in absehbarer Zeit wohl kein Spiel mehr für zehn Plattformen herausbringen. Aber für fünf bis sieben Plattformen fühlen wir uns auch in Zukunft gut gerüstet!

Jan Theysen, Arne von Öhsen

Lessons Learned

Entscheidet man sich für die Entwicklung eines Multiplattform-Spiels, gibt es viele neue Faktoren, die beachtet werden müssen. Ist man sich dieser Mehrarbeit bewusst und hat die potenziellen Fehlerquellen stets im Blick, hat man mit Unity eine Engine, die einem beim Erreichen neuer Plattformen viel Arbeit abnimmt.

Wir haben die Erfahrung gemacht, dass die Umsetzung eines Spiels für mehrere Plattformen allen Versionen des Spiels zugutekommt. Eine Controller-Steuerung, skalierbares User Interface, umfangreiche Speicher- und Performance-Optimierungen und vernünftige Fehlermeldungen werden überall gern gesehen. Na ja. Fehlermeldungen nicht. Aber ihr wisst, was wir meinen.

Um die Entwicklung von Spielen für viele Plattformen in Zukunft noch einfacher zu gestalten, haben wir unsere Erfahrungen in



Unser aktuelles Spiel, »Die Zwerge«, wird nicht für Last-Gen-Konsolen und nicht für Mobile Devices herauskommen, also »nur« auf fünf oder sechs Plattformen (mal schauen, was Nintendo NX kann).



Making Games 02/2016

RELEASING AN INDIE GAME ON 3 CONSOLES AT ONCE AND FAILING FINANCIALLY

After releasing a former game jam project on Ouya, the team of Juicy Beast decided to port »Toto Temple Deluxe« to three consoles at the same time, which turned out to be a financial fail instead of a big success. Here is what the team learned after the release.



Yowan Langlais
is Co-Founder, Art Director
and UI Designer
at Juicy Beast.

Yowan is the co-founder, art director and UI designer at Juicy Beast (www.juicybeast.com), a small indie studio born in 2009 in Montreal, Canada. He specializes in UI, UX and game design, but has always been fascinated by minimalism, elegance and computer generated art in general.

[@wyel](#)
[@juicybeast](#)

In May 2013, we attended Toronto Game Jam and made a local-multiplayer game called »Toto Temple«. About a year later, we released a first version called »Toto Temple Deluxe« on Ouya. We kept improving the game afterwards and released the final version of Toto Temple Deluxe on PS4, Xbox One, Wii U and Steam on September 29th, 2015. The game didn't sell very well, and we think we know why. Here is what I will talk about on the following pages:

1. Why invest two years on such a small game?
2. Was the game at least fun?
3. Gameplay analysis
(how about those other popular games?)
4. Marketing experiments
5. Why didn't the game sell?
6. Next time

1. Why Invest Two Years on such a Small Game?

The truth is, we never actually planned to do anything with Toto Temple (the game jam version). We were presented with a chain of opportunities, and we simply decided to take each one of them as the development went on.

Year 1: Working with Ouya

The first opportunity we took after the Toronto Game Jam (where the game ended up being one of the favorites), was taking Ouya's money to make a full-fledged game out of Toto Temple. At the time, Matt Thorson's »Towerfall« was Ouya's killer app, and they wanted to cash in on that local-multiplayer trend (who wouldn't). We took the money and made a bigger version of the game, which we then called Toto Temple Deluxe.



The deal was simple, get a bigger game (and more experience) in our portfolio without spending our own money. The game went live on Ouya, people enjoyed the Deluxe version, we even received really nice feedback from the press at E3 (at Ouya's booth).

Here we were, a couple of weeks after the Ouya release, with a »console friendly« game in our hands, which technically didn't cost us anything to make. For us, this was something unusual: We used to make small Flash and mobile games, and we now had a »big« game that could be played in the living room with actual controllers. It felt like a big deal, as we could remember actually saying a couple years ago that »we would probably never be able to make console games«. At the time, we didn't have the skills to develop for consoles, and Unity wasn't really a thing, yet.

Year 2: Porting to Consoles

This is where we took the opportunity to get in touch with the three giants. We arranged meetings at GDC and pitched Toto Temple Deluxe to the three console owners.

They all liked the game and things moved forward pretty quickly after that. We officially became PS4, Xbox One and Wii U developers, which still felt like a really big deal. We were in the big leagues now.

Being total beginners in the console market, we quickly realized that we would never get an interesting exclusivity deal with any of three console owners, since the game was already out on Ouya.

Knowing that, we decided to release the game on all three consoles. The decision was mainly based on two points:

- Our thought process back then was that if we can't strike a good exclusivity deal, then let's do all three consoles for three times the revenue.
- We also thought that we probably wouldn't get the media's attention three times for the same game (the three releases), so let's release the game on all three consoles at the same time. That way, we would get decent attention one time and speak to all console owners at once.

2. Was the Game at least Fun?

We genuinely think Toto Temple Deluxe is a lot of fun, under the right circumstances. We designed the game to be as family friendly as possible, meaning that anyone should have a minimum amount of fun, no matter how skilled their opponents are (auto-balance system, etc.). It's also designed to be played with real people, in the same room, on the same couch. The more yelling, the better!

We also witnessed genuine fun and engagement from players who tried the game in public events, like at PAX and Comiccon.

3. Gameplay Analysis

We don't think the game didn't sell well because it wasn't fun. On the other hand, Toto Temple Deluxe is still quite different from those other popular local-multiplayer games.

Intuitive Controls

One of the first things you notice when you first play Toto Temple Deluxe is how tricky it can be to effectively move around in the game (by using the dash mechanic).

Even if we did our best to explain the controls in the most intuitive way we could think of, there's still a difficulty curve that you need to overcome to really feel at ease while moving around.

It definitely comes in the way of our goal to make the game family friendly. We should either have made the controls simpler, or aimed at a more »experienced« crowd from the start. Not both.

In Towerfall, for instance, everyone can be deadly by simply pressing the »shoot« button once. It's that simple. In Toto Temple Deluxe, you need to:

1. Get close to the goat carrier.
2. Use a combination of two buttons to dash towards the goat carrier.
3. Hope that you were aiming right.
4. If not, you need to turn around and start over.

It's definitely not a problem once you get the hang of it, but it's not making the game as easy to get into for new players.

About Toto Temple Deluxe

Toto Temple Deluxe is a fast-paced, local-multiplayer party game in which players must steal an egg-laying goat from their friends and try to keep it on their own head for as long as possible.

Dashing Gameplay

Infinitely dash in any direction to swiftly move around and powerfully head-butt the goat carrier to steal its four-legged booty. Then you have to escape your opponents.

Classic Mode

In Classic mode the player has to gather 3,000 points first by holding on to the goat, collecting coins hatched from goat eggs and harnessing the power of power-ups.

Bomb Mode

In Bomb Mode the player has to be the first to grab the explosive goat and keep it long enough for it to detonate near his/her rivals. If the goat is lost, the timer resets. This mode can be played as teams or in free-for-all.



Toto Temple Deluxe is a game that is meant to be played in local multiplayer. It was created during a game jam and was later developed into a full game.



The Show

Tricky controls can actually be a good thing if they're funny to watch. Our friends from Breakfall did just that with their game »Starwhal«. The game is pretty hard to control, but it's part of the show, as you watch your starwhal wiggle its way towards your opponent's heart.

In Toto Temple Deluxe, it's actually pretty hard to make sense of what's happening on the screen if you don't know the game already.

Toto Temple Deluxe doesn't really give a »good show« by default. You never see spectacular kills like in Towerfall or Starwhal, and not many actions are worth their own replay or slow motion sequence (there are, but they are harder to spot). It's not making the game less fun to play, obviously, but it's making it less fun to watch. It might not be that attractive to YouTubers and Twitchers for this reason.

Intensity Variations

In most competitive local-multiplayer games where you can »eliminate« players in some way (like Samurai Gunn, Nidhogg, Towerfall, Starwhal, etc.), you often find significant variations in intensity.

The tension goes up as you're fighting or trying to survive, but once you or the other player is eliminated, there is this drop in intensity that lets you breath (and celebrate) for a couple seconds, which feels right. On paper, it would probably look like in [figure 1](#).

In Toto Temple Deluxe, the intensity graph probably looks more like in [figure 2](#). You get micro rewards or satisfactions every time you steal the goat or strike a power-up, but the game doesn't stop nor celebrate it that much, it just keeps going. Your real reward is at the end, when you finally reach 3k points and win the match.

We don't think this kind of intensity is necessarily a bad thing, but it forced us to reduce the time of a match to a maximum of about two minutes. Anything passed that would really

drain you out mentally, and a single game was usually more than enough.

All of these elements are probably not responsible for the game not selling well, but they do have an effect on the actual feel and accessibility of the game.

4. Marketing Experiments

We tend to do everything in-house (like contacting media, official websites, video trailers, community management, etc.). We've never been fans of spending money on marketing for things we could do ourselves. We also try to come up with creative solutions to market our games to counter-balance our low marketing budgets.

For Toto Temple Deluxe, we really tried to do more than what we usually do. We went to PAX for the first time (remember, we used to make Flash games), and we also paid for PR for the first time as well. While going to gaming events and paying for PR services is pretty traditional, we also experimented a bit with other things like real goats and automatic Twitch raffles.

Paying for nothing PR at PAX East

Our first experience with PR wasn't super useful in the end, to be honest. Even if we managed to get those services for a reasonable price (being »new customers« to a super friendly PR company), the end results left us pretty unsatisfied.

The services they provided were good, don't get me wrong. We managed to get a couple interesting meetings scheduled at PAX (big websites), but the snowstorm that happened that week made some of them cancel. When we got back, the PR company helped us re-schedule new online meetings with the sites who canceled, but none of them answered our multiple emails.

One other big website that actually came to the meeting wrote an article about the upcoming game following the event, but they never bothered writing a review once the game was released. We contacted them multiple times with no luck, which is weird, considering they already showed an interest in the game.

Giving away \$17,000 worth of Toto Temple Deluxe at PAX East

To put things in perspective a bit, we originally designed our PAX demo so that matches would be really quick. That way, we'd have a better flow and have more players try the game. Since matches wouldn't last long enough for the crowd to feel any major engagement, or for hype to build up, we needed a better way of getting spectators more engaged in the game as they were looking at it.

Normally, only the four current players would be really engaged in the game. Maybe they were with a friend, and maybe that friend would get engaged and cheer a bit, but we felt like it wouldn't be enough to build enough engagement or hype overall.

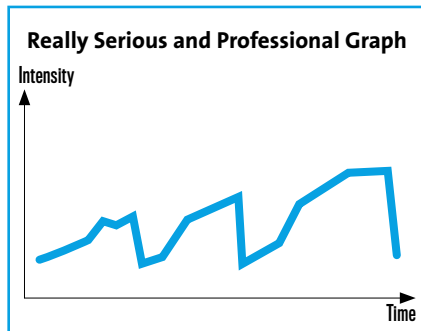


Figure 1: More variations in tension during gameplay means more time to breathe for the player.

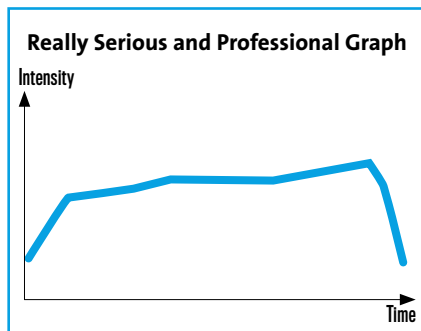


Figure 2: In Toto Temple Deluxe tension is more or less always on the same level and the real reward comes at the end if the player wins the match.



»Starwhal« is pretty hard to control, but it's part of the show, as you watch your starwhal wiggle its way towards your opponent's heart.

We then heard of a strategy that the guys from Vlambeer were doing at events for »Nuclear Throne«. They would have hourly challenges where everyone watching would get a free copy of the game if the current player managed to beat the game. That would normally encourage people to stay and look at the game. They would have something to win (or lose), so they should feel engaged.

We thought it was really clever and it would totally fit our needs, but we decided to push the idea a bit further. Instead of asking players to come back every hour to have a chance of winning a free copy of the game, we decided that every – single – match would be an opportunity to win a copy of the game.

With that in mind, we needed something to link the giveaways to. Ideally, something that wouldn't happen every match, but only once in a while so we don't end up giving away 50,000 copies. That's when we came up with the showdown system:

Toto Temple Deluxe has two main game modes. The Classic mode, where you fight for an egg-laying goat to make points, and the Bomb mode, where you fight for an explosive goat in order to kill your opponents with the blast.

Since we needed something to link the giveaways to, we figured we could use the bomb mode for that. Here's how the system works:

The winner of each match (in Classic mode) would be automatically thrown in a 1 vs. 1 bomb showdown against one bot (set at hard). If that player manages to kill the bot two times, everybody watching would get a free copy of the game.

It was also a good way of having players experience both modes and discover the game's content without having to come back to play a match in the other mode.

This particular experiment (the showdown system) was really successful at the event. We attracted big crowds and got noticed by a lot of players. The only problem is that we didn't get noticed by that much press, which would have helped promote the game later down the road.

The showdown system worked so well with players that we decided that we replicated it at future events (like the Montreal Comiccon), and received the same type of response (only on a smaller scale).

It's definitely something we would try to adapt to our future games, and we encourage anyone to remix this idea to fit their needs.

The Real Goat Bundle

When the game came out, we had a plan to join a good cause by promoting the »Oxfam Unwrapped program«. Oxfam are offering to buy real animals (like chickens, goats, donkeys, etc.), to provide communities in need around the world. It's a sort of donation, basically.

What we did is we committed ourselves to buy real goats in the name of anyone who would purchase the »Real Goat bundle«. The



The showdown system helped that the booth of Juicy Beast at PAX East was pretty much crowded.

buyers would also get four copies (Steam keys) of Toto Temple Deluxe to share with their friends.

Despite being really well adapted to the theme of the game (goats) and being original to some extent, the Real Goat bundle didn't sell at all. We think it's probably due to three factors:

1. The steep price of the bundle (\$60 USD), which basically just covers the price of the goat. We make no money with the bundle, which actually fits the »good cause« idea. We're giving away copies of our game in exchange for a donation to a charity organization.
2. We only felt comfortable giving away PC versions of the game. The game is more enjoyable and popular on consoles, but we're not allowed to distribute console codes that way.
3. The game doesn't support online-multiplayer, so four copies of a local-multiplayer game might feel less interesting a deal if it doesn't let you play with your friends online. We were aiming to give buyers an opportunity to simply gift the game to their friends, but we completely understand the ambiguity.

Bot Bet Battle

Following the release of Toto Temple Deluxe, we organized a one-day Twitch event called »Bot Bet Battle«.

We had the game's bots fight themselves in a classic tournament, and viewers could vote for the bot they thought would win. Everything was AI controlled, so we had no input over which bot would win.

The number of votes would update in real time (or almost), the names of the voters would pop on the screen for everyone to see, and the tournament would start when enough votes were entered to fill the »votes bar«.

At the end of the tournament, everyone who voted for the winning bot would automatically receive a Steam key in the chat system (as a whisper, obviously). You basically had a chance out of four to win a free copy of the game.

We would also ramp up the amount of necessary votes from round to round, so that viewers would invite their friends to accelerate the process.



A yellow warning sign like this which the team used at PAX East is a pretty good way of getting people's attention!



The »Real Goat bundle« consists of a donation of \$60 to Oxfam, who provide a real goat to communities in need and players get four steam keys for Toto Temple Deluxe in return.



In a Twitch event called »Bot Bet Battle« the matches were played by bots and spectators could bet on which one would win to win a steam key themselves.



When Toto Temple Deluxe was released some of the developers who were present in the game as cameos helped to promote the game via Twitter.

We gave away about 1,400 Steam keys during the event. Knowing that a single account couldn't win twice, and that a lot of viewers never won a key, we know we reached a lot more than 1,400 viewers.

We promoted the event on our social pages and a bunch of related subreddits, so there was probably a decent percentage of »new customers« in the lot.

Even though the event was fun, and that we probably managed to reach new followers, we barely noticed any bump in the game's sales.

Cameos

This one took us a lot of time and grew out of proportion pretty quickly. When we started working on the Deluxe version, we knew we wanted to include cameos from other games as a replacement for the goat (what you're fighting for). By other games, we mean games made by friends of ours and other developers we admire and respect. We also wanted to include people in there (not just game characters), like the developers themselves.

All of this was supposed to be a small and cute addition to the game. We were adding unlockable cameos from games we love, and everyone from the team started to pitch in. This is where everything exploded. We created around 150 cameos (and by we I mean JP, our artist), ranging from game characters to their developers themselves.

We reached out to every single developer to introduce our idea and get an approval if they liked it. Some of them were already our friends, which was easy, but a lot of them were big developers we never spoke with before. Sometimes it took a lot of effort and contacts to get the missing email addresses.

We wanted to do things in a clean way, so we wrote a personalized agreement for each of them, stating that the cameos were just cosmetic additions, and that we would never sell them or make money directly from them, etc.

The agreement itself let the devs change the name of the cameos, decide on which consoles they would appear, pick the secret code to unlock them, and if they agreed with the whole idea or not. A digital signature was required, and boom, everything was set (no PDF, no emails).

A couple of months later, on the actual release date, we've seen a lot of those devs share their cameos (and their secret code to unlock them) on Twitter and Facebook. Not all of them shared though. There was obviously no way to predict who would share and who wouldn't, but we still saw a LOT of tweets that day.

It's pretty hard to calculate how much visibility we gained through the cameo system, as we can't access other dev's Twitter stats, but we probably reached a lot of people who wouldn't have been aware of the game otherwise. What we can calculate though, is how many new connections we made with developers we admire, and we're pretty happy about it.

Newsletter

A newsletter system is nothing new, but we also toyed with it for the first time. Okay, maybe the second time. Remember that part where we gave out free Steam keys at events? Well they weren't really free. We actually asked people to subscribe to our newsletter in order to receive the key.

We even created a sweet little backend system (with the help of our good friend Devine Lu Linvega) that automatically distributes the keys without having us send a single email. We just need to upload the keys to a database, and the system handles everything.

1. Users need a unique code printed on the »key card« they won.
2. They join our newsletter by entering their email address and unique code in a Mailchimp form.
3. Mailchimp sends the info back to our backend system, where we validate the code and register the email if it's good.
4. Users automatically receive an email with their key in it.

We managed to gather 1,437 email addresses around Toto Temple Deluxe (not just with free keys), which all received a single email on the release date (we don't like spam). Out of these 1,437 emails we sent, 493 (34 percent) were opened, and 72 (5 percent) got clicked. Based on certain benchmarks, these numbers aren't that bad!

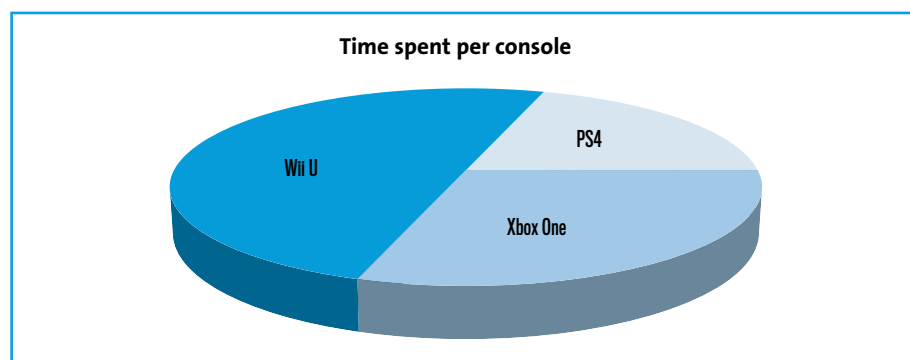


Figure 3: This chart shows how the team spent their time on the actual console ports.



Some examples of cameos that made it into the game.

5. So, why didn't the Game sell?

The game was fun and players were genuinely engaged at events. We reached out to a lot of press with the help of a PR company, but barely got any coverage. We tried different and original marketing tactics, but they didn't result in many changes in the sales. So, what happened?

There's a good possibility that the game didn't have good enough »hooks« to captivate people's attention. Sure, »fighting your friends to put an egg-laying goat on your head« sounds funny and weird, but it doesn't really give you a clear idea of what it's like to play Toto Temple Deluxe.

At the end of the day, we think the biggest factor is because it's a local-multiplayer game with no online play. The game is aiming at a pretty niche audience by requiring actual human friends to play, and we can't ignore the impact it has on sales. A quick look at comments on YouTube, Reddit and such, and it's obvious that a lot of people are simply not buying the game for this very reason (that along a lack of solo experience).

To confirm this idea, one of our contacts from a very well-known gaming site explained to us why they wouldn't write an article on Toto Temple Deluxe. Apparently, it's simply because local-multiplayer news doesn't generate enough clicks to be worth it. We totally appreciated the honesty!

On top of all this, we also think the game came out almost two years too late. Back in 2014, when the local-multiplayer boom was happening, Toto Temple Deluxe's development felt more logical.

Today, it feels a bit out of place, as we think a lot of players might have bought popular local-multiplayer games in the past, then realized that they would play them less and less frequently. We definitely can't blame them, since it's pretty much the same for us.

Obviously, these are all based on our current knowledge and experience, and probably don't apply to all local-multiplayer games. If you're

reading this and you're thinking we missed a point, or have a different opinion on the subject, we'd love to hear it!

Porting to 3 Consoles at Once

Releasing the game on the three consoles at the same time felt like a good idea at the time. We were only thinking about it from a revenues and media coverage standpoint, which was obviously a big mistake (for a local-multiplayer game released two years too late with no online mode, at least).

The Actual Ports

The worst part in all of this is probably that our only programmer, Alex, did all the heavy lifting by himself (the guy's a freaking machine). Making the game run smoothly on all three consoles took him a good amount of time and energy, especially during the certification process.

To give you an idea of the difference between the three consoles, [figure 3](#) shows how we spent our time on the actual ports over the second year of development.

Considering that we made about equal revenues on each consoles so far, you can see that one was definitely more expensive to port to. This might change in the future though, as this was our very first time working with consoles. We also received a lot of help from our friends from Chainsawesome Games for the Xbox One port, which saved us a lot of time and headaches!

Another factor that greatly extended the port process was having to jump back and forth between the different versions of the game. Yes, Unity makes it easier to port to multiple platforms, but the requirements for each console were so different that we ended up having three different Unity projects / branches to support them.

We lost a LOT of time patching little mistakes here and there, like details we added or fixed on one version but forgot to include in the other, etc. With only two team members skilled enough to handle the »technical stuff«, you can surely imagine how this inefficient

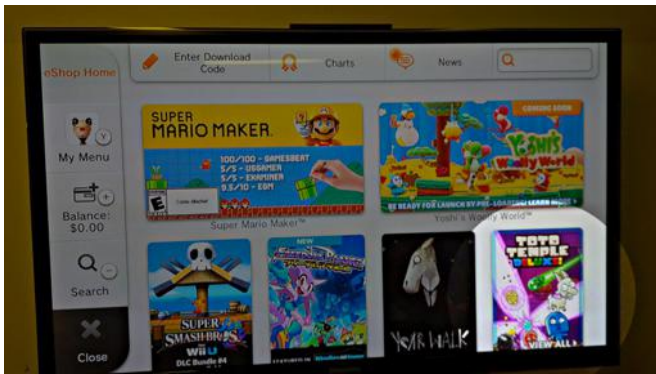
Juicy Beats



Juicy Beast was co-founded in 2009 by longtime friends Yowan Langlais and Dominique Lemire-Nault. Programmer Alexandre Dazé-Hill and illustrator-animator Jean-Philippe Côté joined the ranks shortly after the legal birth of the company. Fresh out of school and with no work experience whatsoever, the four friends took the big leap and released over ten Flash games in the following years, including the successful »Burrito Bison« series and »Knightmare Tower«. After releasing Burrito Bison on iOS with the help of developer Ravenous Games, they slowly shifted their attention towards new markets, and replaced their old friend Flash with Unity. Now with the ability to export to almost every platform, they completely revamped Knightmare Tower and released it on Steam, Ouya, Android and iOS, where it got featured by Apple as one of the Best Games of 2013.

In May 2013, Juicy Beast participated in the Toronto Game Jam (TOJam) where they made a local-multiplayer party game called »Toto Temple«. The jam game got Ouya's attention and they offered their financial help to develop a full-fledged version of Toto Temple. In July 2014, the new and bigger version now called Toto Temple Deluxe was released on the Ouya store.





Of the three console manufacturers, Nintendo was the only one which featured Toto Temple Deluxe in their store.

workflow became extremely messy during the certification process.

Burning Out

We spent two years, full time, working on Toto Temple Deluxe. The first year was about developing the game and its content, while the second year was about porting it to three different consoles at once. That second year is where we started to feel the burn.

We're a team of creative people, and we get our motivation and energy by exploiting that creativity. We like to create things, invent worlds and mechanics. That second year of development was all about paperwork, unusable console portals, technical problems, bugs and desperate searches for information (consoles and their documentation are not really easy to navigate).

Once we shipped the game on September 29th, we all went home and did nothing for at least a week. Porting Toto Temple Deluxe really drained us out, especially for Alex who did an inhuman job all by himself (seriously, a machine).

The process of porting the game was hard by itself, but following up a year of non-creative work with a launch that doesn't make much noise, nor sell a lot, was pretty hard on all of us. We're slowly getting back in shape for a new project (winning the prize of the most addicting game with »Right Click to Necromance« for Indie Speed Run shook things up a bit), but we can still taste the bitterness of the unsuccessful launch.

No Deal, no Help

Back in the days, when we were young and naive, we pictured console storefronts as big marketing machines with lots of users and traffic (remember, we were doing Flash games). We realized pretty quickly that it's not how it works!

Since we couldn't really strike an exclusivity deal with any of the consoles, with the Ouya version already available, and we were porting to all three consoles at the same time (no »semi-exclusivity«), Toto Temple Deluxe probably wasn't perceived as a really interesting deal for any of the consoles. Sure, they were getting a new game for their system, but that was pretty much it.

Because of that, we weren't really able to get a featured spot on any the storefronts, except for Nintendo who actually featured us on both the American and European front pages. Thanks, Nintendo!

What's also interesting is that being featured twice on the Wii-U-eShop actually ramped up the sales of the game to almost equal numbers with the other two.

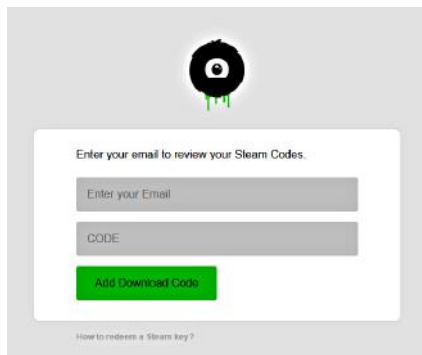
6. Next Time

We've learned a lot with Toto Temple Deluxe, and there's a lot of things we'd do differently (or not at all) for our next game:

1. **No more local-multiplayer:** They're easy to make in a game jam, but super hard to market if you don't add online play.
2. **Strike a deal first:** If we're to make another game on console, we'll make sure to strike an exclusivity deal first, so we can get a bit more help and visibility from the console owner.
3. **Only one console:** It goes without saying that we will only do one console if we strike an exclusivity deal, but even without that, we wouldn't port to three consoles at once ever again. It's a LOT of hard work that didn't really pay well in the end.
4. **Design from a marketing perspective:** Visibility is getting harder and harder to come by, so we will try to design our games with marketing in mind from day one. Having a solid hook and making sure that there's a potential audience for your game is super important, as Ryan Clark from Brace Yourself in his blog »What Makes an Indie Hit?: How to Choose the Right Design« explained.

I think it's easy to think of Toto Temple Deluxe as a failure. Even worst, a waste of time and resources. Sure, the game only generated a fraction of what we spent on it, but we've learned a lot. Now it's time to get up, dust off our shoulders and get the creative juice flowing again!

Yowan Langlais



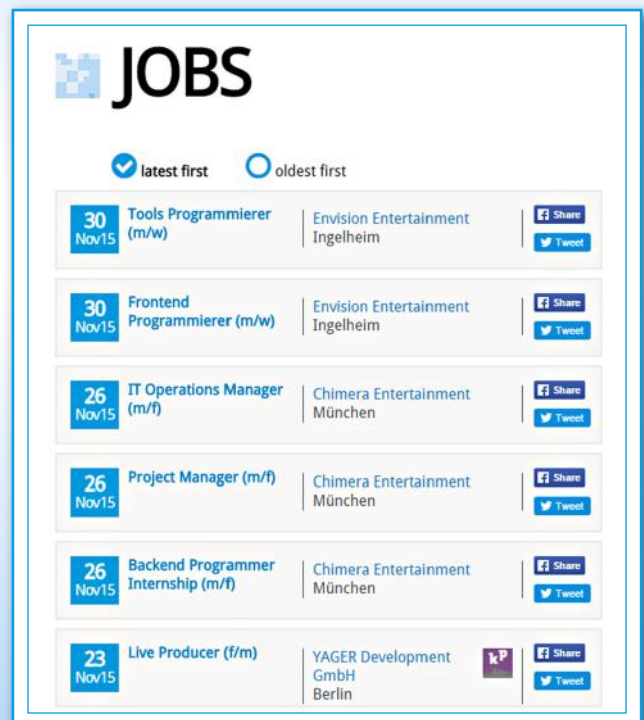
Anybody who won a steam key in all the different marketing actions had to register to Juicy Beast's newsletter, what was a good way to get in contact with new customers.



JOB EXCHANGE

Are you looking for new employees?
Do you want to hire the best talent in the industry?
Then the job exchange on makinggames.biz is the perfect match for you:

- Makinggames.biz is the **BIGGEST WEBSITE** on game development in Central Europe with the perfect target group for your jobs!
- High reach thanks to the **CROSSPROMOTION ON GAMESTAR.DE AND GAMEPRO.DE**, Germany's leading B2C-websites on videogames.
- **GREAT VALUE** – choose a single job posting for 95 Euros per month or a flatrate for 595 Euros per month, offering a subscription model with bonuses like free ads.
- A **FREE TRIAL OFFER** lets you post up to 3 jobs free of charge for a runtime of 14 days and guarantees a **RISK FREE BOOKING**.



For more information visit:
www.makinggames.biz/jobexchange



Or get in contact:
Sebastian Weber

Phone: +49 (0)89 / 244 136 650

E-Mail: sebastian.weber@webedia-group.com

underground

ACTUALLY FREE ON AMAZON UNDERGROUND BEYOND FREEMIUM AND IN-APP-PURCHASES

A new business model from Amazon, benefitting Android developers and customers alike as the players get apps for free while developers are getting paid on how long their apps are used. Carlos Lievano describes how the system works and how you can publish your apps on Amazon Underground.



Carlos Lievano
is Senior
Developer Marketing Manager
at Amazon Appstore.

Carlos Lievano is Senior Developer Marketing Manager at Amazon Appstore, engaging games and apps creators with the distribution and monetisation opportunities offered by Amazon. He is also an experienced entrepreneur, with projects including a Tech Ed business and a mobile games studio.

The Amazon Appstore allows developers to distribute and sell their Android and HTML5 web apps to millions of customers in 236 countries and territories. It is available on a multitude of devices, and is pre-loaded on Android phones on select carriers in the U.S. and Europe. And now the Amazon Appstore has over 500,000 apps globally.

Amazon has tools and services that help you monetize your apps and keep users engaged. It's easy to get started. You can test your app in under a minute and most Android apps just work on Amazon devices, with no additional development required. Developers can get started in minutes at developer.amazon.com/apps.

Amazon Underground

In August 2015, Amazon introduced the new Android shopping app Amazon Underground to customers in the U.S., UK, Germany, and France and just expanded to Spain, Italy, and 14 other European countries and territories at the beginning of December 2015. While Amazon Underground boasts all the well-known features and categories of the regular Amazon mobile shopping app, it also comes with an exciting subsection benefitting Android users: apps that are actually free and there's no premium subscription like Amazon Prime required to use Amazon Underground. More than 17,000 EUR in paid apps, games and in-app

items were available to customers on Amazon Underground at the time of the December expansion to additional European countries.

With Underground, Amazon wants to provide their customers with an experience to shop, watch, and play in one place. In addition to millions of physical goods they can discover and purchase, unlimited hours of their favorite movies and television shows they can stream, customers can also browse through entertaining audiobooks, useful apps and addictive games – now with the additional feature of apps and games that are 100 percent free – marked as actually free. All Android customers need to do is download the App directly from Amazon at amazon.de/underground, as Google doesn't allow an external app that offers games or apps to be included in Google Play. Fire HD and HDX users don't have to do anything – Amazon Underground is already preloaded.

Being Actually Free

Customers won't have to pay a single cent to download popular premium apps like »Office Suite Professional 8« and »Monument Valley«. What's more, they can also download popular titles that come with In-App-Purchases, like »Frozen Free Fall«, »Star Wars Rebels: Recon Missions«, »Angry Birds Slingshot Stella«, and many more – except they won't be expected to pay for any of the items needed to get ahead in the game if they got the app from Amazon Underground. In times when fewer



»Colossatron« from developer Halfbrick is one of the actually free games in Amazon Underground and could see an increase of downloads and revenue since then.

users are willing to shell out for IAPs or higher priced premium apps without knowing if they will actually use it, our actually free model can be a game changer.

What does this mean to Developers?

This is made possible by introducing a brand new business model which will allow developers to turn 100 percent of their users into revenue generating customers. How? If developers agree to waive their regular download or in-app-fees, Amazon will pay them instead on a per-minute-played basis. Payments may vary in different regions. In Europe, the current rate is 0.0018 Euro. This means as soon as the users start playing the game or using the app, developers start getting paid for each minute their users spend with the Amazon Underground app. The decision whether to go premium or freemium is obsolete for developers who choose Amazon Underground. Instead of focusing on a monetisation model, they can put all their energy into creating a truly engaging mobile experience, either through an addictive game, a useful lifestyle app or kid-friendly and ad-free content. Developers no longer have to convince users to pay a premium price or wait around for them to convert to paying customers as they become fans of the game – as long as they keep the customers engaged, there is no need to sell them anything.

The fact that the selection of free apps, games, and in-app purchases on Amazon Underground have tripled since release to more than 1,500 apps by the launch in the additional European territories in December is proof that developers are adopting this new business model. Revenues are subject to increase for different reasons:

- You'll monetise all your users: If your game is free to play, instead of monetising a small percentage of users that convert via in-app purchasing, you'll get paid for every minute all of your users are playing.
- You'll increase user engagement: By taking away the frictions associated with payment walls and in-app purchasing, no user will leave your game at that point in which those mechanisms start to work against you.

- You'll get new users: If your game is premium, you're reducing the price barrier to all users that would have been interested, but aren't eager to pay upfront for their games. Moreover, the game mechanics that you had in place to influence users into spreading the word out, will become more effective, by becoming a game anyone can afford.

Real-Life Example: Halfbrick's Colossatron

The results for developers are already showing. For instance, developer Halfbrick, creator of blockbuster games like »Jetpack Joyride« and »Fruit Ninja«, with worldwide downloads of more than 500 million and 1 billion, respectively, across all platforms, has released many popular mobile games over the years using traditional monetisation models like premium, freemium, and ad-supported. According to Shaniel Deo, founder of Halfbrick, »We had varying degrees of success. [Amazon Underground] really piqued my interest when I heard about it. This is something that I think is innovative, and allows us to reach a different audience, while at the same time generate revenue from that audience.«

»Colossatron« was the first game Halfbrick published to Amazon Underground, and Deo



Amazon automatically adds Amazon Underground branding to any assets uploaded through the dev portal so you don't have to do this manually



This figure shows how downloads and revenue changed after Halfbrick published »Colossatron« on Amazon Underground.

has been pleased with the customer response. »Session lengths went up, and the overall hours that we saw in the Underground version (compared to the regular version of the game) were incredible – hundreds of percent higher. It was really amazing to see. All of that translates into more hours played in total, and ultimately, revenue.« In fact, monthly downloads nearly tripled after the game became available through Amazon Underground, and total revenue rose by more than 750 percent.

Deo believes one of the keys to the success of Colossatron makes it well-suited to Amazon Underground. »It's one of those games that has a real »stickiness« about it; people engage with it and play it for a long time.« Still, even Deo was surprised by the bump in revenue, though he's not complaining. »We wouldn't have been able to do that without this unique business model. I've been very happy with the performance.«

Now several games from Halfbrick's catalog are available on Amazon Underground, and Deo believes there are opportunities to get even more from the platform. »There's a lot of scope to really look at the strength of the model: the fact that you can monetise your entire audience. I would definitely recommend getting your apps onto Amazon Under-

ground; it's another viable platform and it's incremental revenue.«

How to turn an App into an Amazon Underground App

Amazon Underground represents an easy opportunity to convert non-paying users into paying ones through deeper engagement and longer sessions spent in the app.

The four basic steps to convert a freemium app for Amazon Underground are:

1. Make free all your IAP items.

You have two options to execute this step. Those options are:

Option A – Re-create and re-submit to the Amazon Appstore using new SKUs for the Amazon Underground version of your app. Set each re-created item to its original IAP price; Amazon will apply strikethrough pricing to your items to show that they are now free. You will also need to update your code base to use the new SKUs. Keep in mind that testing for your app will not start, unless you have successfully submitted your IAP items before submitting your app. To create and submit the new SKUs, you need to follow these steps on the Amazon Apps & Games Developer Portal:

1. Sign in to your account on the Amazon Apps & Games Developer Portal
2. Go to the My Apps view
3. Locate your app and choose the Manage in-app items option from the app drop down menu
4. Choose the in-app item you want to create (Consumable, Entitlement, or Subscription)
5. Configure the specifics around the purchasable item
6. Specify the SKU
7. Click Save
8. Repeat steps 4 to 7 for each purchasable item you want to create

After you have created all the new SKUs, you will be ready to submit your application.

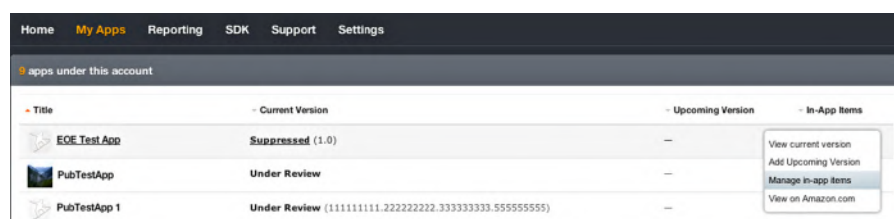
Once submitted, it is subject to the flow chart depicted in [figure 1](#). The following considerations are important regarding the different application status:

- You can cancel your app submission while your app is in SUBMITTED status by clicking the Cancel review button.
- If an issue is encountered with your app or a purchasable item defined for your app, your app status will change to PENDING, and you will be messaged with an explanation of the issue encountered and additional information as appropriate. Additional testing will be suspended until you submit an updated version of the app or purchasable items.
- Once your app has been approved for publication, the status will change to APPROVED and your app will be live in Amazon Appstore.

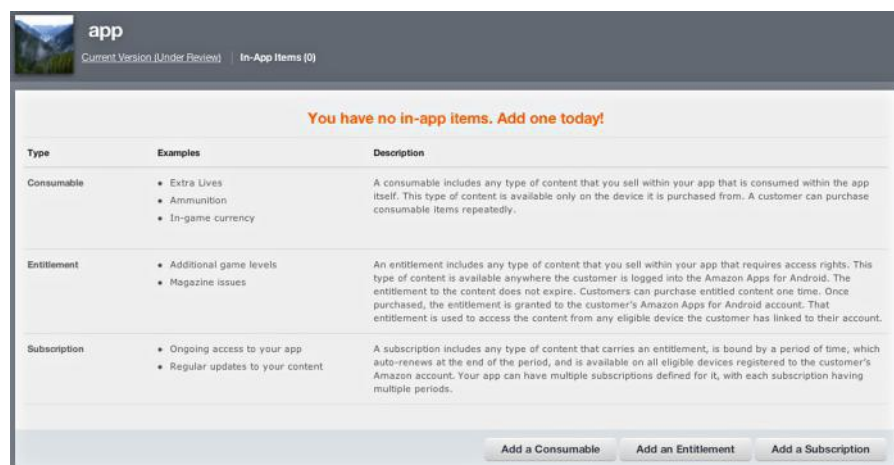
Note that if you are not using Amazon's IAP API because your app uses a different in-app



Halfbrick published several games from their portfolio on Amazon Underground.



Step 3: Managing in-app items for your app



Step 4: Choosing the in-app item you want to create.

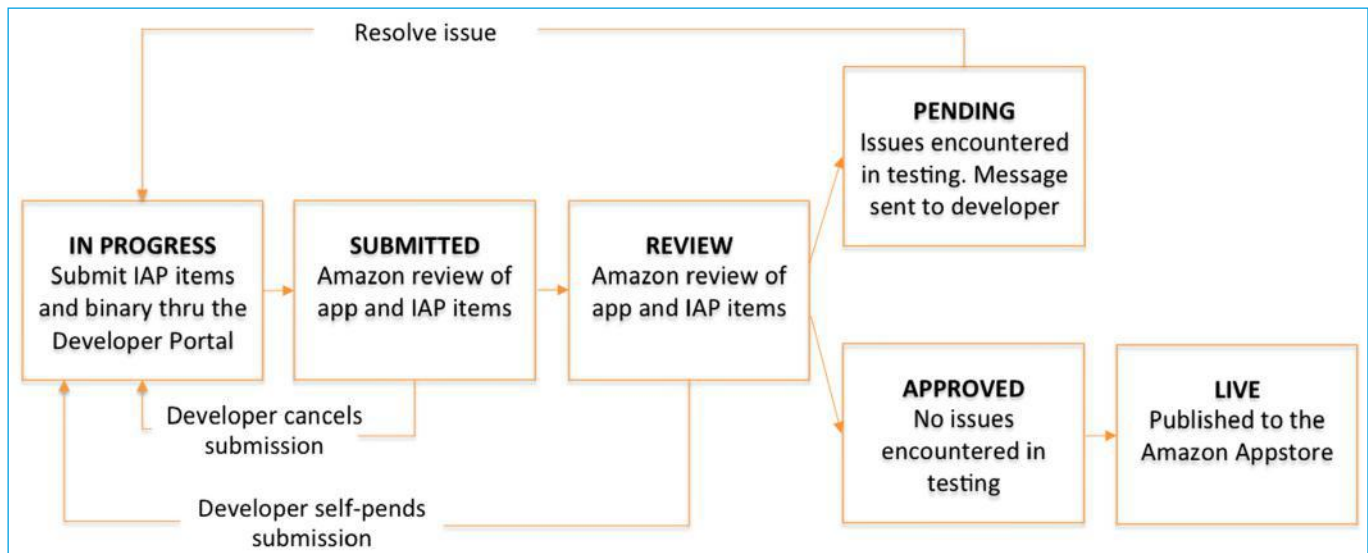


Figure 1: The App Submission Flow Chart shows the status your app will go through in the submission process.

purchasing API, such as Google's In-App Billing (IAB), you will need to implement Amazon's IAP API to use this option. This implementation will also be necessary if you want your app to be made available to run on Amazon devices, such as Fire tablets, as Google Play's IAB will not work on Amazon devices. It's an opportunity to reach new users and be discovered by all customers using Amazon Appstore.

If your app uses Google's IAB, and you wish to port your code to the Amazon IAP API for the first time, you will need to follow this steps:

Step 1: Configure your `AndroidManifest.xml` file to support Amazon's IAP. Configure your `AndroidManifest.xml` file to define the IAP Response Receiver for your app. The IAP Response Receiver ensures that the intent communication from the Amazon Client is intercepted by your app. Note that if you are supporting both Google Play's IAB and Amazon's IAP, you do not need to remove elements related to IAB; they will simply be ignored by IAP. In your `AndroidManifest.xml` file, add the appropriate `<receiver>` tag for the IAP ResponseReceiver:

```

1 <application>
2 ...
3 <receiver android:name="com.amazon.inapp.iap.ResponseReceiver">
4 <intent-filter>
5 <action android:name="com.amazon.inapp.purchasing.NOTIFY"
6 android:permission="com.amazon.inapp.purchasing.Permission.NOTIFY">
7 </action></intent-filter>
8 </receiver>
9 ...
10 </application>
  
```

Step 2: In your app, implement logic to mediate between IAB and IAP. You can use the same code base for your app, regardless of where your app is hosted. Just add logic to your app to

determine whether the app was downloaded from the Amazon Appstore or from Google Play. Depending on where the user downloaded the app, execute the appropriate methods for either IAP or IAB.

The following example code determines whether a package was installed from Google Play or the Amazon Appstore:

```

1 PackageManager pkgManager = context.getPackageManager();
2
3 String installerPackageName =
4 pkgManager.getInstallerPackageName(context.getPackageName());
5
6 if(installerPackageName.startsWith("com.amazon")) {
7 // Amazon
8 } else if ("com.android.vending".equals(installerPackageName)) {
9 // Google Play
10 }
  
```

Step 3: Add and implement the Amazon IAP API. For the most part, Amazon's IAP works similarly to Google Play's IAB. When you create the path in your code to implement the IAP API, you should be able to use a similar logic

flow to IAB, but will need to account for different class and method names.

The following table maps the most frequently used IAP methods to their IAB equivalents:

PurchasingService method	PurchasingListener callback	Response object	Equivalent IAB call
<code>getUserData()</code>	<code>onUserDataResponse()</code>	<code>UserDataResponse</code>	N/A
<code>getPurchaseUpdates()</code>	<code>onPurchaseUpdatesResponse()</code>	<code>PurchaseUpdatesResponse</code>	<code>getPurchases()</code>
<code>getProductData()</code>	<code>onProductDataResponse()</code>	<code>ProductDataResponse</code>	<code>getSkuDetails()</code>
<code>purchase()</code>	<code>onPurchaseResponse()</code>	<code>PurchaseResponse</code>	<code>getBuyIntent()</code>
<code>notifyFulfillment()</code>	N/A	N/A	<code>consumePurchase()</code>

Note that Amazon's `notifyFulfillment()` call and Google Play's `consumePurchase()` call correlate with each other somewhat. Calling `consumePurchase()` on a managed item for IAB will return the item's state to »unpurchased«, essentially converting it to a consumable. After that, the receipt will no longer appear in the response from subsequent calls to `getPurchases()`. Amazon will return the receipts for consumable items in calls to `getPurchaseUpdates()` until the app calls `notifyFulfillment()`.

Step 4: Test your app. To test the API integration, download and install the App Tester tool provided by Amazon in their developer portal. This tool can also test Mobile Associates APIs, if your app uses those, as well. After testing your app locally, you can use the Live App Testing service, also provided by Amazon, to beta test your app in a live production environment with a select group of users.

Option B: Remove the IAP integration from your app, but provide an alternate mechanism for users to obtain IAP items for free. For example, build a »store« that does not actually make any IAP calls or has any IAP items on the backend but whose items can be purchased through non-monetary ways. Your users could then »buy« items from your store, just as they normally would do for IAP items, but the »currency« for the store could be game experience, time played, or something else. When developing ways to unlock IAP items, keep in mind that you will not be permitted to require users to watch ads to unlock IAP items.

Implementing Option B involves the following high-level tasks:

- Remove the IAP SDK from your code.
- Modify your purchase execution call to fulfill former IAP items to your users for free: Add your own logic for what each former IAP button does, such as a »Buy« button. (This logic is handled by the IAP SDK for non-Amazon Underground apps.) Fulfill the item to the customer.

We encourage you to update the icons and strings for your IAP items for Amazon Underground to emphasize to users that they are now receiving these items for free, which will help to communicate to your users the value of the items that they are now receiving for free.

2. Repackage your existing app

In order for your game to work with Amazon Underground you will need to create a new APK for your app. This APK will need a unique package name that includes appending the

string ».underground« to the original package name for the app. For example, »com.company.example.underground« would be the package name for the Amazon Underground version of »com.company.example.«

3. Ease migration of existing customers (optional)

While not required, it would be a good user experience if you ensured that customers of your current app can keep and use any saved data in the new Amazon Underground version as well. These are some of the data and items that users generally expect to be available when migrating to a different version of your app:

- Game progress (levels achieved, high scores, rewards gained)
- Individual or character progress (a level 20 knight vs. a level 1 knight)
- In-app currency
- Competitive progress (leaderboard standing)
- Settings and preferences (colors, control layouts, defaults, sound options)

Now, we will present you with some of the tools that enable you to achieve this, as well as a suggested workflow for the migration between versions. Keep in mind that this isn't something you will only want to do to migrate a user into Underground, but also to ensure that users do not lose their progress achieved while your app was part of Amazon Underground.

There are different tools – cloud-based sync solutions – you can use to migrate the user progress across different version of your app. You may find that the best solution for your app is to use a combination of the following options:

Amazon Solutions:

- Amazon GameCircle with Whispersync for Games: Use GameCircle to share game achievements and leaderboards across the non-Underground and Amazon Underground versions of your title. Use Whispersync for Games to sync user data, including settings, preferences, and pure game state data.
- Amazon Web Services S3 and Cognito: Use Amazon Web Services (AWS) S3 and the Data Transfer API from the AWS Mobile SDK to create your own cloud solution, or use Amazon Cognito to sync user identity and mobile app data in the AWS Cloud.
- Amazon Cloud Drive: Amazon Cloud Drive is available as a paid service that can store data, similarly to DropBox or Google Drive.
- Login with Amazon: Use Login with Amazon to manage user identity support between app versions.

Identity Services: Identity services generally work in similar ways, so we will use Login with Facebook as an example: Use the Facebook SDK to provide identity support. This solution requires you create two separate Android App IDs, one for the non-Underground version, which you already have if you are using Facebook Login, and a new one for the Amazon Underground version. Each of these IDs uses the same Facebook Hash Key but will need separate package names. To locate your Facebook Hash Key, select any application from the developer console and navigate to the Binary File(s) tab. In the Appstore Certificate Hashes section both SHA-1 and MD5 hashes can be found.

Proprietary Solutions: If you have your own proprietary cloud save solution, be aware that using the package name at run-time to access data may not work for sharing data between the different Amazon versions of your app. You might need to modify your app's code to transform the Amazon Underground version of the package name to match the non-Underground version, or you may need to modify your cloud solution to accept an alternate identifier for identifying the app and version.

Android app preferences, settings, and files created by an app are stored locally in Shared Preferences, Internal Storage or SQLite databases. A new version of an app, such as an Amazon Underground version, does not have visibility to these data stores. To share this data, the non-Underground version of your app must store data in external storage – in the Android sense – so that the Amazon Underground version of your app can also access these items.

The following process is a suggested workflow for migrating locally stored data and files from a non-Underground version of your app to an Amazon Underground version. Remember to follow general security best practices when working with externally-stored files. To migrate these data and files:

1. Update the non-Underground version of your app so that each time that the state is changed in private app storage, the appropriate data is also stored in external storage.
2. In the Amazon Underground version of your app, on startup, check external storage for shared data, and import this data into the private storage of the app. If this data import is lengthy, communicate to the user that the app is importing the previous app state.
3. After importing the data into the Amazon Underground version of your app, delete the data from external storage.

Note that the techniques described here also apply to Amazon devices, such as Amazon Fire tablets.

4. Create new marketing material for the Underground version

You will need to upload new screenshots and a new app description to the developer portal.

It's important to explain the value you're giving customers and call out any additional features of your app as part of the Amazon Underground program. Also be watchful of changes that involved changes to the in-game experience shown in the screenshots, especially if one is showing the in-app store. Amazon automatically adds Amazon Underground branding to any assets uploaded through the dev portal.

5. Upload the new APK

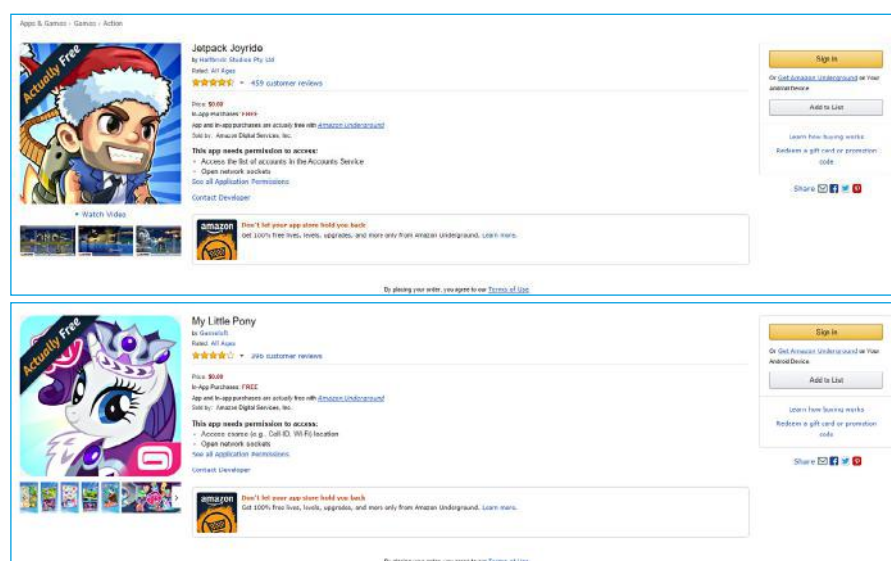
Simply upload your new APK, description, and screenshots, and submit for approval through Amazon's developer portal (developer.amazon.com/apps). Once the app passes testing, your app will be live to all Amazon Underground customers.

For any premium apps, you will be following the same steps, only skipping the first one. Getting an existing premium app converted for Amazon Underground is quite easy – developers won't even have to change the game's code to get it ready for Amazon Underground – though of course they are free to do so, as long as the Underground version is similar or better than the regular premium app.

Conclusion

With Amazon Underground, Amazon has created a one-stop-shop for all their customer's needs – from physical products to TV shows, films, audio books to apps and games. Moreover, with the new business model for apps and games, customers receive actual value – for free. A release on Amazon Underground increases customer loyalty and engagement, promotes discoverability, and supplements revenues. Preparing your Android and Fire apps for Underground is just as simple as covered here. And since Amazon's developer program is free to register, you can benefit from this new opportunity now.

Carlos Lievano



All games on Amazon Underground are discoverable on the Amazon website and are identifiable as actually free.

LOCALISING COMEDY

Comedy is one of the biggest localisation challenges. Mark Estdale explains the importance of game assets, context and the necessity of changes to avoid a weak game localisation.



Mark Estdale
is CEO
at OMUK.

Mark is the pioneer and evangelist of Game Immersive Voice Recording and specialist on the adaptation of acting for espionage. He is a voice director turned audio engineer, audio technology and tools expert. He joined the industry in 1994 as a consultant to build the first dedicated voice studio for games in the UK. Many studios and over 600 titles later Mark runs OMUK, one of the most successful international voice production teams in London and Los Angeles.



The example of »Zero Wing« for Sega Genesis shows why Japanese to English translations are infamous in the industry.

Getting comedy right in localisation requires a game asset rich approach that provides the freedom to deviate from the original language. If this is not understood and resourced then the localisation will be weak.

In his book »Is that a fish in your ear?« Professor David Bellos discusses what he calls the »myth of literal translation«. He argues that the key to translating comedy »lies NOT in preserving the literal text of the original but instead ABANDON the idea of perfect fidelity and try to find humour that rings some of the same bells as the original.« Finding what »rings bells« needs intimate knowledge of the context of the game.

Context is King

The best solution therefore is to translate whilst playing the game. But doing this takes time and there is a distinct possibility the game won't be ready when translation is required to start. Without context you will create bugs. You need to decide between the cost (including time) of preparing assets to give rich context for the start of the localisation process and fixing localisation bugs in testing. There is a clear statistical correlation between quality of contextual information and the number of bugs.

If the game isn't ready, the next best thing is to use raw game assets for context: character

art, object art, scene art; and any audio assets: recorded voice, music, and sound effects. It's also a good idea to write contextual notes – especially when wordplay is involved.

Localising Verbal Humour

Localising verbal humour is difficult, and in some instances, impossible. The humour in puns or wordplay comes mainly from exploiting the multiple meanings of similar sounding words. As these similarities frequently don't exist between languages, it is important to remember that »perfect fidelity« isn't the aim, capturing the essence of a moment is. This is the core difference between pure translation and localisation. Translation focusses on the words and localisation on the objective and momentum.

A developer who has worked on a game for two years will know the context intimately, a translator coming in fresh won't. The more context you can give the better. Also having a short and fast chain of communication between the translators and the developer is a big advantage for dealing with questions effectively.

Changes are necessary

Making changes to enable a scene to work in a different language needs to be understood by the whole team. Humour is full of cultural references so adapting the scene to make the

humour work in a different market is an essential ingredient in localisation requiring trust between the creative team and the localiser.

Notorious in the industry are Japanese to English localisations. The most notable reason for cringe-worthy localisations is perhaps the fact that historically Japanese publishers and developers have retained the last word of localisation veto. They often over-control the creative input of the localisation partner, consequently constraining and losing out on the creative cultural input of the localiser. Freedom to localise is freedom to adapt. Constrain it at your peril.

Escape the Literal

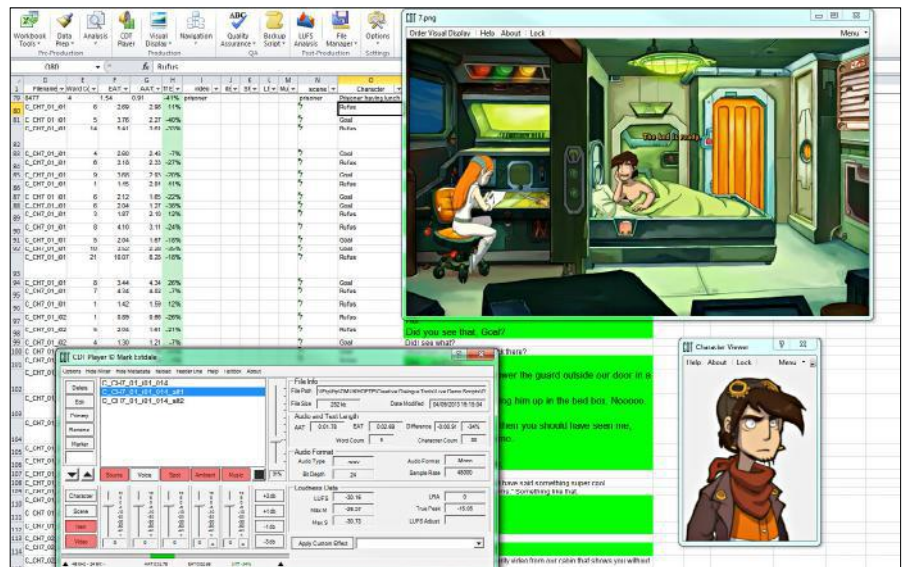
When recording the English version of »The Book of Unwritten Tales 2«, a translation bug was flagged by the developer in a response to the character Wilbur making a comment »It doesn't look like ice«. In German the response to Wilbur was »Weil es hier unten völlig windstill ist! Ohne Wellen und Blasen im Eis sieht gefrorenes Wasser aus wie ... Wasser.« The localisation in English was »So?« This is a translation bug but NOT a localisation bug. If the German was translated literally the scene would not have worked as well in English. Focussing on the intention, and NOT the literal meaning, is about preserving the momentum and tension in a scene above all else. The change would not have been possible without knowing the context of the scene. The context brings meaning to the moment and makes it possible to escape the literal.

Scene: A little ladder stands next to a large step-ladder and says to the step-ladder: »you're not my real father.« This is a wordplay in English that ties step-ladder with step-father. How would you make it work in German? The choice is to find a different joke that works or to remove the joke. A literal translation would not work.

Daedelic's »Deponia« series of games written by Jan Müller-Michaelis are full of German language puns and wordplays that sadly don't translate between German and English. So if you play the English versions you will find either completely different puns or no pun at all. I was very lucky to have Jan join me in the studio for recording the English versions of the games. His input was priceless as he could explain his sometimes obscure humorous intent. We had time to experiment with the English. The whole studio team including the actors joined in the process. The collaborative effort was extremely valuable. Without knowledge of the context and intent it would have been impossible to get the results we did.

The Monkey Wrench

A while ago I was speaking with a non-English adventure game developer about some really odd puzzles he had in his game. Like most adventure gamers he was influenced by LucasArts.



For the localisation of Daedalic's »Deponia« series, writer Jan Müller-Michaelis gave valuable input.

Whilst defending his puzzles he referred to »Monkey Island 2«. He said »If you think my puzzles are weird, what about Guybrush having to have sex with a monkey to stop a waterfall.« At the time I had no idea what he was talking about so I checked it out.

The funny thing is he thought the puzzle really cool even though he completely missed the simple American wordplay. If you're unfamiliar with the game, Guybrush needs to close a valve to turn off the waterfall. In Britain a spanner does the job, but in the US it's a Monkey Wrench, hence the monkey. A classic wordplay. While the intentional humour in the writing had been missed, it didn't really matter that much as the playing experience wasn't lost. The original joke was completely lost in translation but it didn't matter for the gamer. The essential thing is to keep the momentum of the playing experience.

I spoke to some of the original Monkey Island team about the monkey animation and they admitted that the sex animation was done that way on purpose. It just happened to slip by the censor.

Fidelity to the Intention

The keys to successfully localising comedy lie in putting fidelity to the intention of a scene above fidelity to the words that create it. Attention to detail will help you faithfully capture the essence and intention of a scene. As developers, share as much as you can as this helps everyone understand the context and make the writing and performance work. This may mean adapting your pipelines but the impact will be significant. It is context that makes it possible to escape from a literal translation. Without it you have only the words to work from. Finally, it's worth noting that this holds true for all aspects of localisation and voice recording, not just localising comedy.

Mark Estdale



Knowing the context of each scene in »The Book of Unwritten Tales 2« was extremely important for the localisation process.



The original joke of the monkey wrench in »Monkey Island 2« was lost in translation.

7 WEGE, UM DEN SPIELER BEI DER STANGE ZU HALTEN

Ein Spiel kann so gut sein, wie es will – wenn es den Nutzer nicht zu fesseln vermag, hat es ein essenzielles Ziel nicht erreicht. nextMedia.Hamburg stellt sieben Faktoren vor, die die Aufmerksamkeit sichern können.



Alexander Karl
ist Redakteur
für nextMedia-Hamburg.

Alexander Karl, Jahrgang 1989, schreibt für nextMedia.Hamburg, die Hamburger Initiative für die Medien- und Digitalwirtschaft. Sie wird getragen von der Freien und Hansestadt Hamburg, der Hamburgischen Gesellschaft für Wirtschaftsförderung mbH, dem Verein Hamburg@work sowie engagierten Unternehmen. Ziel der Initiative ist es, die Spitzenposition der Medienmetropole Hamburg zu sichern und die Rahmenbedingungen für die hiesigen Unternehmen zu verbessern. Ein Projekt der Initiative ist auch gamecity:Hamburg, mit über 2.000 Akteuren das größte regionale Netzwerk der Games-Branche in Deutschland.

Es ist das Ziel jedes Spiels: die User zu fesseln und ihre Aufmerksamkeit möglichst lange an das Geschehen zu knüpfen. Mit diesem Ziel sind Computerspiele nicht allein, denn bereits die alten Griechen fragten sich, wie ihre Texte und Geschichten die Menschen unterhalten können. Zwar hat sich seit damals vor allem technologisch viel getan, aber die Mechanismen der Aufmerksamkeitsökonomie sind doch ähnlich. Welche Elemente Videospiele brauchen, um die Aufmerksamkeit ihrer Spieler zu sichern, weiß der Hamburger Professor Gunther Rehfeld. Für nextMedia.Hamburg, die Standortinitiative für die Medien- und Digitalwirtschaft, hat er sieben Wege kommentiert, wie der User in Spielen bei der Stange gehalten werden kann.

1. Die Story

»Um was geht's da?«, das ist die typische Frage, wenn man seinen Freunden von einem neuen Spiel erzählt. Je komplexer ein Titel ist, desto schwieriger ist diese Frage oft zu beantworten. Welche Elemente eine gute Geschichte jedoch ausmachen, darüber gibt es unter-

schiedliche Meinungen. Das wohl bekannteste Schema für den Aufbau von Geschichten ist die Heldenreise, die der Mythologie-Experte Joseph Campbell aus einer Vielzahl alter Mythen herausgearbeitet hat. Dabei umfasst die Heldenreise 17 Stationen und beginnt mit einer Aufgabe für den Heroen, führt ihn durch Abenteuer und bringt ihn schließlich mit neuem Wissen in seine alte Welt zurück.

»Gerade in narrativen Action-RPGs lässt sich das Prinzip wiederfinden«, erklärt Gunther Rehfeld. »Ein Beispiel sind die »Bioshock«-Titel von 2K Games, in denen sich der Protagonist einer Vielzahl von Aufgaben stellen muss. Aber die Heldenreise lässt sich auch leicht über viele Titel stützen, weil Campbell seine Stationen der Heldenreise weit gefasst hat.« Für den Bereich des filmischen Drehbuchs hat übrigens Christopher Vogler ein Handbuch zu der »Odyssee des Drehbuchschreibers« verfasst, das auch für Spieleentwickler eine lesenswerte Anleitung zum Ablauf und Aufbau der Heldenreise sowie eine Schilderung der Kern-Archetypen bietet. »Es empfiehlt sich aber auch, die asiatischen narrativen Strukturen zu kennen, wie sie in der chinesischen Erzählung »Die Reise nach Westen« vorkommen. Der Heldenbegriff ist dort buddhistisch-daoistisch geprägt.« Wer also ein internationales Publikum ansprechen möchte, sollte den Blick über den westlichen Tellerrand wagen.

2. Die Charaktere

Apropos Heldenfigur: Jedes Spiel hat einen oder mehrere Protagonisten. Außer es handelt sich um »Tetris«. Aber die großen, aufwändigen Produktionen legen ihren Fokus auf einen oder eben mehrere Charaktere, die durch die Story führen. In den vergangenen Jahrtausenden haben sich hier bestimmte Typen herausentwickelt, die Carl Gustav Jung als Archetypen zusammengefasst hat. »Dabei reicht das Spektrum vom klassischen Helden bis zum Schatten, also dem Bösewicht oder der inneren Unzufriedenheit. Archetypen unterscheiden sich somit



Die Charaktere in einem Spiel können ein wichtiger Aspekt sein, damit sich der Nutzer langfristig mit einem Titel beschäftigt. Dabei müssen sie aber nicht nur in Gut und Böse klassifiziert sein, auch Antihelden wie in »GTA 5« üben einen gewissen Reiz aus.



Die Quests, die ein Spieler erfüllen muss, können eine fesselnde Wirkung haben. Ein gutes Beispiel ist hier »City of Heroes«, denn dort können Spieler sogar eigene Aufgaben entwickeln.



Einige Spiele setzen auf Mysterien im Hintergrund der eigentlichen Handlung, durch die der Nutzer noch tiefer in die Story einsteigen kann – wie etwa »Lost Ember«.

einerseits in ihren charakterlichen Merkmalen, andererseits aber auch in ihrer Funktion für die Handlung. Oft ist der Held die Figur, die durch das Spiel führt, während der Schatten ihm als Bösewicht in die Quere kommt. Es gibt aber hier nicht nur Gut und Böse, sondern ebenfalls ambivalente Figuren. Sogar ein Antiheld wie Rufus in »Deponia« kann eine gewisse Sympathie erreichen«, erklärt Professor Rehfeld. »Auch die Verbrecher bei »GTA« sind typische Antihelden, die man als Spieler gern mal mimt.«

3. Das Spielziel

»Der Spieler sollte immer genau wissen, wo er ist und wo er hin will. Wo bin ich, wo komme ich her, wo will ich eigentlich hingehen – das sind die zentralen Fragen, die immer beantwortet werden müssen«, so Rehfeld. Das heißt jedoch nicht, dass keine Rätsel auftauchen dürfen; nur danach muss immer wieder Klarheit geschaffen werden. Dafür sind ein gutes UI-Design sowie ein klarer Aufbau der Spielelemente zentral. Eine symbolische Aufladung von Elementen, wie etwa Schwertern, mit einem erzählerischen Hintergrund ist eine Möglichkeit – vergleichbar mit dem Schild des Achill in der Ilias. Denn gerade bei narrativen Titeln ist die Motivation des Spielers entscheidend, und wie im wahren Leben treiben einen Ziele an, von denen es in Spielen meistens verschiedene gibt, um das diversifizierte Publikum anzusprechen. »Das gilt für MMOs, aber ebenso für »Clash of Clans«. Auch vermeintlich sinnloses Blöckchenschieben in Casual Games, um den persönlichen Highscore zu knacken, kann ein Spielziel sein.«

4. Die Quests

Eine Faustregel für gute Quests gibt es nicht, wohl aber unterschiedliche Formen: Von der Sammelquest über die Storyquest bis zur Begleitungsquest stehen unterschiedlichste Möglichkeiten zur Wahl, Aufgaben zu stellen. »Gerade Sammelquests, in denen man zum Beispiel Spinnen oder Ratten aufsammeln soll, sind leicht zu erstellen, bieten allerdings oft nur wenig narrative Relevanz. Storyquests, wie in »GTA« einen Wagen von Punkt A nach B zu bringen, formulieren hingegen ein klares Ziel. Schön umgesetzt sind die Begleitquests bei »Guild Wars 2« in den regionalen Quests gewesen«, meint Rehfeld. Einen interessanten Ansatz, der sich jedoch eher auf einer Metaebene befindet, bot »City of Heroes«. »Da wurde der Quest-Generator ins Spiel integriert«, weiß Professor Rehfeld. Auf Basis der Kern-Questmechaniken konnten die Spieler in »City of Heroes« eigene Quests entwickeln und die anderer Spieler bewerten. Dennoch werden die Questtexte von den Spielern meist nicht gelesen, sondern einfach nur weitergeklickt. Denn das eigentlich Wichtige sind ja die Aufgabe und die anschließende Belohnung.

5. Die Belohnung

»Die Belohnung nimmt im Spiel eine zentrale Rolle ein, um voranzukommen. Das kann innerhalb des Spiels der Levelaufstieg oder das Knacken des Highscores sein.« Doch auch Belohnungen wie zum Beispiel neue Schwerter für den Helden können eine wichtige Triebkraft sein. »Kreative Belohnungssysteme

nextMedia.
Hamburg

nextMedia.Hamburg ist die Hamburger Initiative für die Medien- und Digitalwirtschaft. Sie wird getragen von der Freien und Hansestadt Hamburg, der Hamburgischen Gesellschaft für Wirtschaftsförderung mbH, dem Verein Hamburg@work sowie engagierten Unternehmen. Ziel der Initiative ist es, die Spitzenposition der Medienmetropole Hamburg zu sichern und die Rahmenbedingungen für die hiesigen Unternehmen zu verbessern. Ein Projekt der Initiative ist auch gamecity:Hamburg, mit über 2.000 Akteuren das größte regionale Netzwerk der Games-Branche in Deutschland.

ne»xt
MEDIA.HAMBURG



Eine Community kann essenziell für die Spielerbindung sein. Im Falle von »World of Warcraft« rekrutierte Blizzard dort sogar Angestellte.



Gunther Rehfeld
ist Professor für Grafik
und Bildbearbeitung
an der HAW Hamburg.

Gunther Rehfeld, Jahrgang 1961, studierte Kunst an der HfBK Hamburg. Nach seinem Studium war er zunächst als Freelancer, primär für neue Medien, tätig. Nach einer Professur für Medienkommunikation und Mediengestaltung an der FH Kaiserslautern, wechselte er im Herbst 2001 als Professor für Grafik und digitale Bildbearbeitung an die HAW Hamburg, wo er bis heute lehrt. Zu seinen Schwerpunkten gehören die Identitätsbildung durch Narration in Gemeinschaften im Spiel und durch Gestaltung von Lebensräumen. 2007 erfolgte die Einrichtung des Gamecitylab in Kooperation mit der Fakultät IT, 2010 entstand der MA-Teilstudiengang »Games«, anschlussfinanziert durch die gamecity-Hamburg. Zudem ist er Lehrender im Games-Master (www.gamesmaster-hamburg.de). 2013 veröffentlichte er das Buch »Game Design und Produktion: Grundlagen, Anwendungen und Beispiele« im Carl Hanser Verlag.

bietet etwa Blizzard Entertainment in seinen Titeln, aber auch die Hamburger Browsergames-Entwickler von Goodgame Studios oder InnoGames, die geschickt den Highscore-Gedanken einsetzen«, so Rehfeld. Dabei gibt es unendlich viele Belohnungen, die verschiedene Typen von Usern unterschiedlich ansprechen. Das Finden von Gegenständen, das Erreichen von selbst gesteckten Zielen und so weiter, was dann wiederum mit dem Gewinn von Gegenständen, Gold, sozialer Anerkennung, Ruhm, Ehre, Achievements, besseren Plätzen in Rang- und Highscorelisten, Lob von Mitspielern, der Steigerung der eigenen Skills und Ähnlichem aufgewertet wird. Biologisch gesehen ist hier mitunter die Dopaminausschüttung von großer Bedeutung, wenn der Spieler bestimmte solcher Punkte im Spiel erreicht.

6. Die Mysterien

Einige Spiele setzen auf Mysterien im Hintergrund der eigentlichen Handlung, durch die der Nutzer noch tiefer in die Story einsteigen kann – das Hamburger Startup Mooneye Studios arbeitet gerade in »Lost Ember« daran. Dort soll der Spieler mit Hilfe von verschiedenen Tieren den Geheimnissen einer vergessenen Kultur auf die Spur kommen. Und wer »Age of Mythology« gespielt hat, fragt sich vielleicht, was es

mit Zeus, Odin oder Isis auf sich gehabt haben soll. Doch die Wahrscheinlichkeit, dass ein Spieler plötzlich die Sammlung antiker Sagen von Gustav Schwab liest, hält Gunther Rehfeld für gering. »Der Medienwechsel zum Buch ist eher unwahrscheinlich, mit etwas Glück kann aber der Sprung zu Wikipedia gelingen.« Wer jedoch ein Spiel mit zahlreichen Verweisen auf die Antike oder das Mittelalter plant, muss sich trotzdem nicht sorgen: »Es kann gut funktionieren, wenn die Mysterien sinnvoll eingesetzt werden. Allerdings glaube ich nicht, dass alle Spieler es interessant finden werden.« Sich als Nutzer in den im Hintergrund liegenden Mysterien zu verlieren, ist also eher das Sahnehäubchen, sofern man an der Story davor schon Gefallen gefunden hat.

7. Die Community

»Die Einbeziehung der Community ist sehr wichtig für die Entwicklung und Vermarktung der Spiele«, erläutert Rehfeld. Besonders gut geklappt hat das bei »World of Warcraft«. »Community-Leader und angesehene Spieler wurden da ins Boot geholt, um das MMO zu entwickeln. So konnte noch zielgerichteter auf das Publikum hin produziert werden. Aber auch bei dem Release einer unfertigen Version – Stichwort: Steam Greenlight – ist es wichtig, der Community sehr genau zuzuhören.« Und da sollten dann alle Kommunikationskanäle beachtet werden, von Social Media bis hin zu einschlägigen Foren. Zentral ist da natürlich, den Input zu filtern und auf seine Relevanz zu prüfen. Doch auch nach dem Release sollte die Verbindung zum Kunden nicht abreißen: »Je besser man auf die Kundenwünsche und Probleme eingeht, desto besser kann man sein Produkt weiterentwickeln. Gerade die Leute, die sich intensiv mit einem Spiel auseinandersetzen, werben auch dafür – oder können genauso gut das Image eines Titels ankratzen.« Das gilt sowohl für die realweltlichen Peergroups als ebenso für soziale Netzwerke.

All diese Faktoren führen zu dem, was das zentralste Element für jedes Spiel ist: der Spaß. »Aber das, was Spaß macht, ist bei jedem Nutzer und bei jedem Spielprinzip anders«, hält Rehfeld fest.

Alexander Karl



Verweise auf die Antike wie in »Age of Mythology« oder auf das Mittelalter können durch den realen Hintergrund Faszination im Spieler auslösen, sodass er sich tiefergehend mit der Materie beschäftigen möchte.

OPEN WORLD AUF MOBILE MIT UNITY

»TruckSimulation 16« ist eine LKW-Simulation für Mobilgeräte, in der die Spieler in einer riesigen offenen Spielwelt umherfahren können. Christoph Ender vom Entwickler kunst-stoff erklärt, wie sein Team diese technische Herausforderung mit Unity bewältigt hat und welche Probleme es zu lösen galt.



Für eine LKW-Simulation wie »TruckSimulation 16« ist das Gameplay klar definiert: Der Spieler wird fahren. Daher sollte die Umgebung ansprechend und nicht repetitiv sein. Das Autobahnnetz von Mitteleuropa ist die vorgegebene Spielwiese. Jede Straße sollte eine gewisse Abbildung der Realität in Verlauf, Landschaftsgestaltung und markanten Punkten bieten. So eine große, nichtprozedurale Spielwelt braucht wesentlich mehr Speicher, als der Arbeitsspeicher eines Smartphones Platz bietet. Daher wird sie in kleine Häppchen aufgeteilt und während des Spiels bei Bedarf, also dynamisch nachgeladen.

Dynamisches Nachladen

Im Idealfall werden diese Häppchen (patches) der Landschaft komplett in einem 3D-Modellierprogramm erstellt. So erhält man die volle Kontrolle über Detailgrad und Sichtbarkeit der verschiedenen Bereiche und kann Abbruchkanten zum Beispiel an Straßengraben sauber anfertigen. Dieses Vorgehen ist für die Größe

unserer Welt leider ein zu großer Aufwand und damit zu teuer. Es darf außerdem nicht vergessen werden, dass die Level Designer mit einem 3D-Modellierprogramm umgehen können müssen.

Wir haben uns trotz begründeter Leistungswarnungen für das in Unity eingebaute Terrain entschieden. Diese Terrains definieren sich durch ein Höhenfeld und lassen sich schnell im Editor erstellen, ändern und mit Skripten automatisieren. Außerdem lassen sich die Renderqualität von Geometrie (pixelerror) und jene von Texturen (base-distance) sehr gut der Leistung des Gerätes anpassen. Der Kompromiss aus Qualität und Speicherverbrauch lag bei einer Auflösung von einem Datenpunkt pro Einheit und Meter.

Die Welt mit ihrer Landschaft, den Häusern, Straßen und so weiter wurde in quadratische Kacheln der Größe 1024 mal 1024 Meter unterteilt (**Abbildung 1**). Ein Quadratkilometer ist groß genug, um eine gewaltige Stadt oder ein Autobahnkreuz bequem darin zu platzieren, und nicht zu klein, um den Level Designern das Leben schwer zu machen. Im Nachhinein



Christoph Ender
ist Technischer Direktor
der kunst-stoff GmbH.

Chris arbeitet seit 2011 (mit Unterbrechungen) bei kunst-stoff und hatte die technische Leitung für das Projekt »TruckSimulation 16« inne. Als großer Fan von Rennsportspielen fährt er den Racetruck besonders gern. Nach seinem Informatikstudium mit Schwerpunkt Computergrafik zog es ihn nach Berlin, wo er seit elf Jahren bei lokalen Spieleentwicklern arbeitet. Er kann auf einen Track-Record von 25 Spieletiteln zurückblicken.
www.xing.com/profile/Christoph_Ender2



TruckSimulation 16 umfasst eine offene Spielwelt mit Mittel- und Westeuropa als Vorlage und mehr als 20 Städten. Ein so großes Areal auf mobilen Geräten mit Unity umzusetzen, war keine leichte Aufgabe.

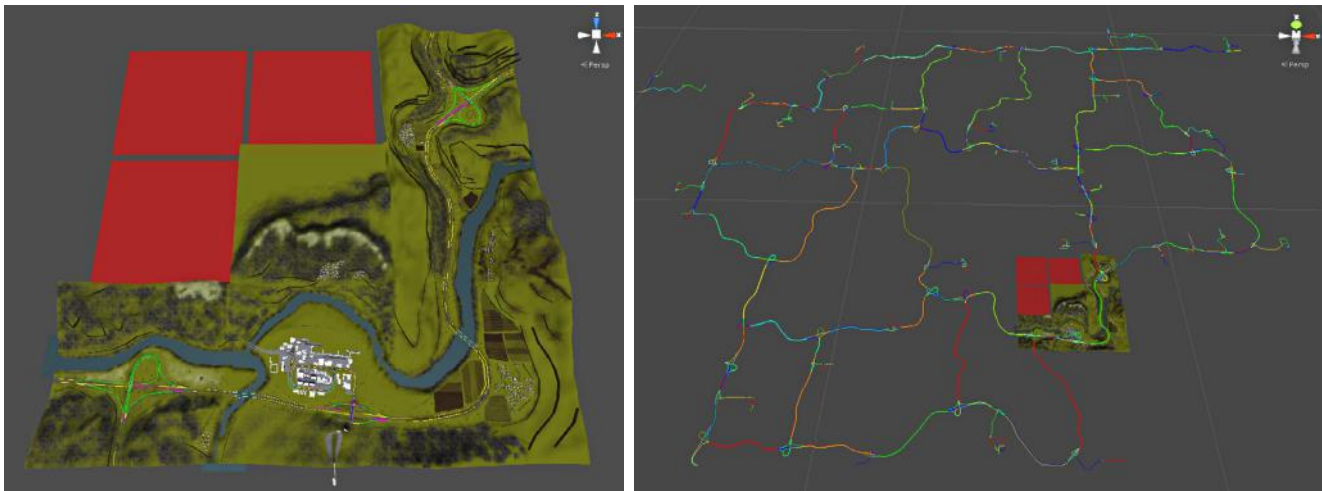


Abbildung 1: Das Terrain mehrerer Landschaftsszenen kann gleichzeitig editiert werden.

wären etwas kleinere Kacheln aber besser gewesen, um den Speicherverbrauch und die Nachladezeit zu verringern. Bei einer offenen Spielwelt ist es üblich, drei oder acht Nachbarkacheln zu laden und anzuzeigen. Nach einigen Tests beschränkten wir uns jedoch auf nur eine Nachbarkachel.

Das ist möglich, weil wir durch unsere Straßenführung die Kameraposition einschränken können. Es fällt dem Spieler praktisch nicht auf und spart viel Aufwand und Arbeitsspeicher. Die Sichtweite in der höchsten Grafikqualität haben wir auf – für ein mobiles Spiel erstaunliche – 600 Meter festgelegt.

Für das Durchfahren einer Landschaftskachel braucht der Spieler mit dem 200 km/h schnellen Racetruck nur rund 20 Sekunden. Ein Mobilgerät der eher geringen Leistungsklasse (wie das iPhone 4s oder Samsung S3) schafft es gerade noch, rechtzeitig im Hintergrund die nächste Kachel zu laden und anzuzeigen. Sollte es etwa durch fremde Apps massive Leistungseinbußen geben, so würde der LKW in die Tiefe stürzen. Um den Spielfluss nicht zu unterbrechen, müssen Kacheln im Hintergrund laden – also asynchron. Das unterstützt Unity, indem jede Kachel in einer eigenen Szene gespeichert wird. In jeder dieser Szenen gibt es nur einen Wurzelknoten (root), um sie einfach entladen zu können. Alternativ könnte man auch mit einem Pool aus instanziierten Objekten und einem eigenen Ladesystem arbeiten. Beim Setzen des Höhenfeldes eines Terrains bleibt eine Unity-Anwendung leider immer ein paar Millisekunden stehen. Beim Nachladen von Szenen kommt es deshalb auf langsamen Geräten hin und wieder zu kleinen Rucklern. Und wir haben die Rechnung ohne »Mr. Physics« gemacht. Die meisten Szenen beinhalten Hunderte, wenn nicht sogar Tausende statische Collider für Leitplanken, Straßenlaternen, Häuser und so weiter. Die interne Integration in das Raumunterteilungsverfahren von PhysX dauert wohl einige Millisekunden.

Die Landschaft

Leider ist das Editieren des Terrains leicht umständlich, kleinere Tools aus dem Asset Store helfen hier. Verschiedene Terraingeneratoren wie »Terrain Composer« wurden ausprobiert, waren aber nicht überzeugend. Der Boden besteht aus einer Mischung von kleinen, wiederholbaren Texturen (splat maps). Die Verwendung von Satellitenfotos ist technisch machbar und wurde getestet, würde aber viel Arbeitsspeicher, sehr viel Medienspeicher und viel zu viel Zeit zum Laden benötigen.

Im Laufe des Projektes taten sich mehrere Unschönheiten mit Unity-Terrain auf:

- Die virtuelle Kamera im Unity-Editor hat eine nicht änderbare Einstellung, infolge derer das Terrain Artefakte rendert, die eine genaue Platzierung von Objekten erschweren.
- Die eingebaute Unterstützung von Bäumen ist auf Mobilgeräten unbrauchbar. Ist solch ein Baum auf einem Terrain gesetzt, halbiert sich die Bildwiederholfrequenz (frame rate) auf dem Gerät, wenn man die Kamera rotiert.
- Es können nur wenige Bodentexturen pro Kachel verwendet werden. Bei mehr als drei bricht die Leistung ein.
- Auf einigen Android-Geräten mit Mali-Chips wird die Bodentextur fehlerhaft als grobes Schachbrettmuster gerendert. Als Abhilfe sind nur ungerade Splat-Map-Indizes zu verwenden. Wir haben die praktisch überall verwendete Grastextur auf die zweite Stelle geschoben.
- Das Terrain bietet eine performante Kollisionsabfrage, was das Durchfallen des LKWs durch den Boden verhindert. Diese wurde auf Mobilgeräten in seltenen Fällen erst nach einigen Frames aktiv, sodass der Truck dennoch durchfiel.

Als Test wurde das fertige Unity-Terrain in einem aufwendigen automatischen Prozess in statische Geometrie umgewandelt (Abbildung 2). Die Zahl der Polygone wurde mit dem Mesh-Simplifizierungswerkzeug »Cruncher« stark

kunst-stoff
GmbH

kunst-stoff ist ein unabhängiger Spieleentwickler aus Berlin. Das Team arbeitet an Titeln für PC und Mac sowie für Konsolen und mobile Plattformen. Zu den bisherigen Veröffentlichungen zählen neben »TruckSimulation 16«, »Galaxy Racers« und die vielfach preisgekrönte »The Great Jitters«-Reihe.

Außerdem setzt das Studio interaktive Exponate und Apps für Kunden aus den Bereichen Industrie und Kultur um.

www.kunst-stoff.de

kunst-stoff
Interactive Media and Games

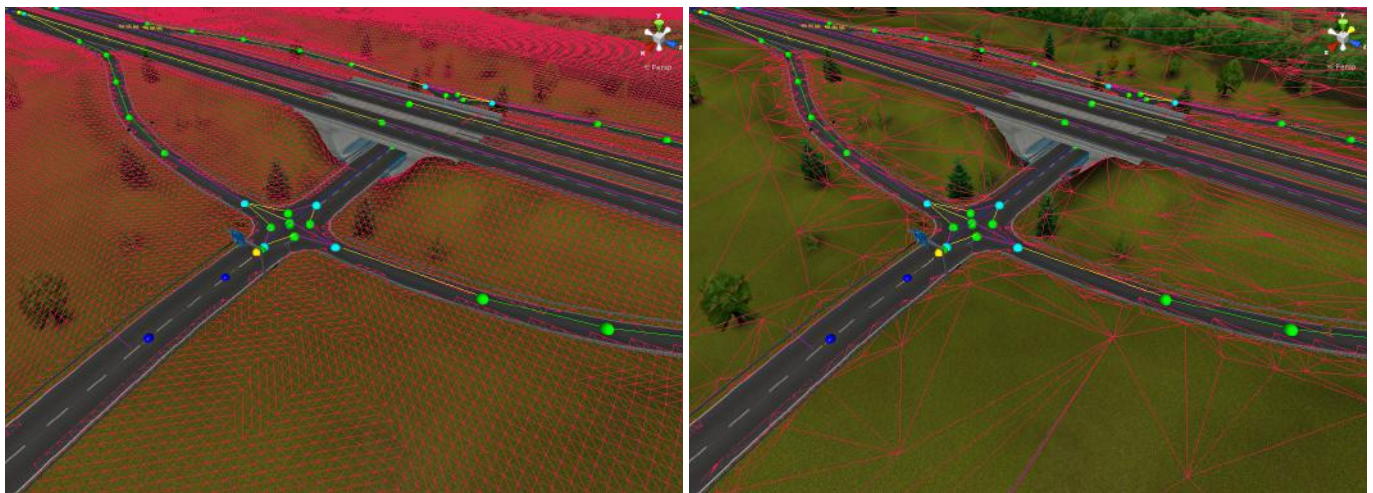


Abbildung 2: Das Unity-Terrain als Höhenfeld kann automatisch in ein polygonreduziertes Mesh konvertiert werden.

reduziert, zur Optimierung wurden die Dreiecke nahe der Straße möglichst nicht angetastet und Dreiecke unter Straßen entfernt. Insgesamt war das Rendern dieses Ansatzes leistungsfähiger, erhöhte aber die komprimierte Größe der App von durchschnittlich 2 MB pro Kachel auf 3 MB. Unkomprimiert entspricht das 5 MB beziehungsweise 7,5 MB. Mit weiterer Entwicklungsarbeit könnte man den Polygonbrei mit Sichtbarkeitsprüfungen gezielter ausdünnen, um dieses Vorgehen zum bevorzugten Verfahren zu machen. Hier bietet die Cruncher-API leider zu wenig Einfluss auf den Prozess. In der Entfernung stellt Unity sein Terrain mit zunehmend weniger Dreiecken dar (dynamic level of detail, ROAM), wodurch es dann Durchdringungsfehler mit der Straße geben kann. Das ist bei einem statischen Mesh nicht der Fall.

Für Tunnelleingänge kann man weder visuelle noch physikalische Löcher in das Unity-Terrain bohren, auch wenn einige Assets aus dem Store das versprechen. Um das Problem zu kompensieren, haben wir das Unity-Terrain im Tunnel unter die Straße gezogen und den Bereich über der Straße mit dem zum Mesh konvertierten Terrain ersetzt. Mit Plugins wie dem »Mesh Toolkit« kann man Meshes in Unity bequem bearbeiten.

Die Straßengeometrie liegt 4 cm über dem Terrain. Es ist der niedrigste Wert, bei dem keine Durchdringungen und kein Z-Buffer-Fighting auf dem Mobilgerät entstehen. Es hat sich herausgestellt, dass die Anpassung des Terrains an die Straße lange dauert und manuell nicht in der Genauigkeit geleistet werden konnte. Aufwendig wurde ein Tool entwickelt, das auch den Straßenrand hübsch anpasst.

Terrains im Allgemeinen brauchen einen Großteil der Rechenleistung auf Mobilgeräten und müssen daher sorgfältig bedacht werden.

Automatisieren

In einer sechsminütigen Fahrmission bei einem Durchschnittstempo von 100 km/h legt der Spieler zehn Kacheln zurück. Als Langzeit-

motivation erstellten wir ein Straßennetz mit 1.300 Straßen auf 180 Kacheln im Aufwand von mehreren Mannjahren! An dieser Stelle möchte ich mich für das tapfere Durchhalten der Level Designer bei dieser fast endlosen Aufgabe bedanken. Eine händische Korrektur, zum Beispiel das Austauschen eines bestimmten Baumtypus, wäre fehleranfällig und würde Tage oder schlimmstenfalls Wochen dauern! Daher sind Skripte zum Abarbeiten aller Kacheln nötig. Ein Beispiel ist die oben angesprochene Umwandlung des Unity-Terrains in statische Geometrie. Eine festgelegte interne Struktur der Kachelszenen ist für Skripte immer hilfreich. Übrigens lassen sich im Editor nur einige Dutzend Szenen gleichzeitig laden, weil die Obergrenze von rund 64.000 Physikobjekten in Unity 4 schnell erreicht wird.

Unity 4 kann leider immer nur ein Terrain bearbeiten. Für das bequeme und gleichzeitige Anzeigen und Arbeiten mit mehreren Kacheln wurde ein Tool geschrieben (Abbildung 1 und 3). Die gemeinsame Grenze zweier Terrains wurde halbautomatisch geglättet (stitching). Beim Speichern von Kacheln werden viele Integritätsprüfungen durchlaufen, etwa ob Straßen korrekt verbunden sind. Es ist wichtig, potenzielle Fehler sofort zu erkennen. Ein komplettes Abfahren und Testen aller Strecken in beide Richtungen dauert mindestens einen Tag. Unity 5 unterstützt inzwischen Multiscene-Editing.

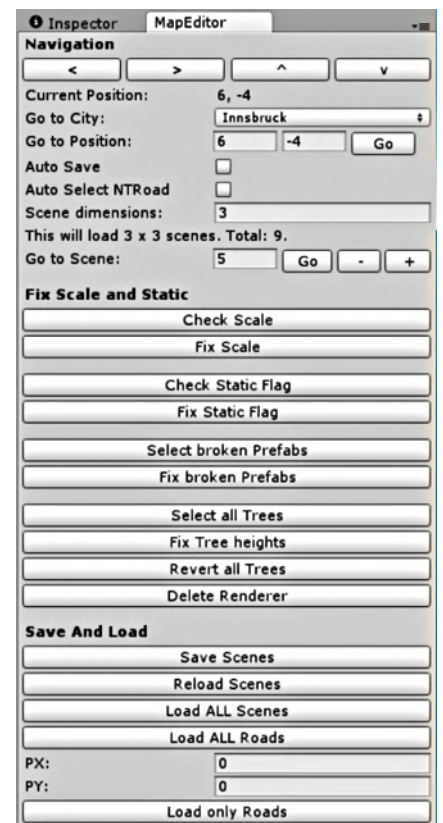
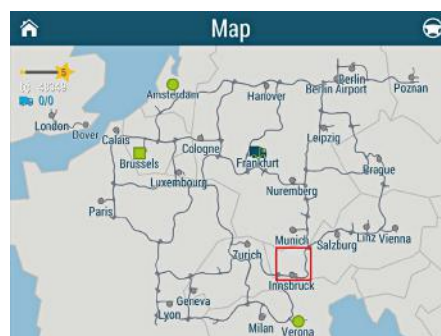


Abbildung 3: Die Landschaftsszenen sind in die vektorisierte Karte des Spiels eingezeichnet (links) und der eigene Tool Map Editor bietet eine bequeme Verwaltung aller Landschaftsszenen (oben).

Licht und Schatten

Ein Echtzeitschatten verlangt Mobilgeräten viel Leistung ab. Bei einer Fahrsimulation fällt außerdem der begrenzte Darstellungsraum der Schatten auf, zum Beispiel wenn in einer Allee nur die nächsten zwei Bäume Schatten werfen, die anderen aber nicht. Lightmaps für die Terrains und alle Objekte sehen toll aus, erfordern jedoch Erstellungsaufwand, Rechenleistung und Speicherplatz. Diese beiden bekannten Schattentechniken verhindern Draw Call Batching (siehe unten). Daher griffen wir in die Trickkiste.

Auf dem Land wurde unter jedem Baum das Terrain dunkler gemacht. Ein Skript oder ein Level Designer erhöhte die Gewichtung für die dunkelste Bodentextur rund um diese Stelle. Diese Maßnahme ist kostenlos, sie braucht also keine Extraleistung. Das Endergebnis sieht ähnlich aus wie die einfachen Projektor-schatten (blobs) unter den Fahrzeugen. Viele Häuser und andere Objekte sind auf dem Land meist weit weg von der Straße und benötigen keinen Schatten.

Für das Szenario »Stadt« ist ein Schatten durch Interpolation der Bodentexturen (blending) oder Multiplikation mit einem Faktor pro Geometrienoten (vertex coloring) des Terrains nicht ausreichend. Die Auflösung des Terrains von einem Meter limitiert die Auflösung des Schattens. Das Terrain wird durch die Straße überdeckt, und Brücken, Häuser und so weiter werfen auch Schatten auf die Straße. Für Städte wurde je eine Lightmap berechnet und auf eine große transparente Ebene mitten in die Stadt gelegt (Abbildung 5). Auch Details wie Laternen werfen hier einen korrekten Schatten auf den Boden. Sie hinterlassen keinen Schatten auf Wänden, diese Situation kommt aber auch nicht vor. Die Lightmap ist wie eine

IGeL-Leistung und kostet extra, etwa einen Frame Leistung auf langsamen Mobilgeräten. Als Einschränkung darf die Sonne nicht zu flach scheinen und nicht wandern, und Städte müssen größtenteils auf einer Ebene gebaut sein. Das alles fällt dem Spieler eigentlich nicht auf. Bei der Terrainvariante mit statischer Geometrie haben wir die Schattierung (nicht Schatten) zur Entwicklerzeit (offline) berechnet und in die Geometrienoten geschrieben, um den eigenen Terrain-Shader zu entlasten.

Eine für spärlich besetzte Landschaften sinnvolle, aber doch nicht verwendete Implementierung von Schatten sind vorberechnete Schattenfahnen. Da Objekte eigentlich beliebig rotiert platziert werden können, kann man den Schatten etwa bei einem Haus in acht Richtungen vorberechnen und auf transparente Geometrie zur Entwicklerzeit projizieren. Oft reicht eine Menge von generischen Schattenformen und mit Drehung und Skalierung der Geometrie kann der Schatten feinjustiert werden. Das Unternehmen Schattenfahnen scheiterte im Vorfeld aber am Aufwand der Erstellung, an Auswahl und Ausrichtung und insbesondere an einer Platzierung am Berg. Insgesamt sind wir mit dem Schatten zufrieden, auch wenn unser Ansatz Spezialfälle wie Tunnel nicht berücksichtigt.

Die Straßen und der Verkehr

Assets und Software für die Simulation von Autoverkehr gab es ein paar, jedoch war keines auf eine riesige Welt und ein dynamisches Nachladen ausgelegt. Die Programmierung eines dynamischen und glaubwürdigen Autoverkehrs hat einige Monate gedauert und sollte nicht unterschätzt werden! Die Implementierung des Straßennetzes und der Auto-KI war Gegenstand unzähliger Diskussionen und Meetings.

Straßen werden durch Kontrollpunkte definiert, die in der richtigen Reihenfolge eine geschwungene Linie oder Raumkurve (spline) und somit die Mittellinie der Straße ergeben. Diese logische Information ist später wichtig für den Autoverkehr, die Karte und die Wegfindung. Für die einzelnen Fahrspuren wird jeweils eine eigene Raumkurve mit einigen Metern Abstand abgeleitet (Abbildung 5).

Die ursprüngliche Idee, während der Fahrt Geometrie und Leitplanken mittels Kontrollpunkten zu erzeugen, hat sich schnell als zu unpraktisch herauskristallisiert. Die Integration des Assets »EasyRoad« zum statischen Kreieren von Straßen und deren Geometrie entpuppte sich erst während der Produktion als Katastrophe. Wir fixten Bugs in schwer editierbarem Quelltext (obfuscated code) und mussten mit Fehlern in den Bibliotheken (dll) leben. Im Nachhinein würden wir die benötigten Funktionen zur Straßengenerierung, zur Anpassung des Terrains, zum Erstellen von Leitplanken, Schallschutzwänden, Laternen und Ähnlichem selbst schreiben. Die Leitplanken erweisen sich aufgrund ihrer Zahl von Polygo-



Abbildung 4: Mit nur einer Textur lassen sich auch ohne Kachelung viele verschiedene Häuser texturieren.

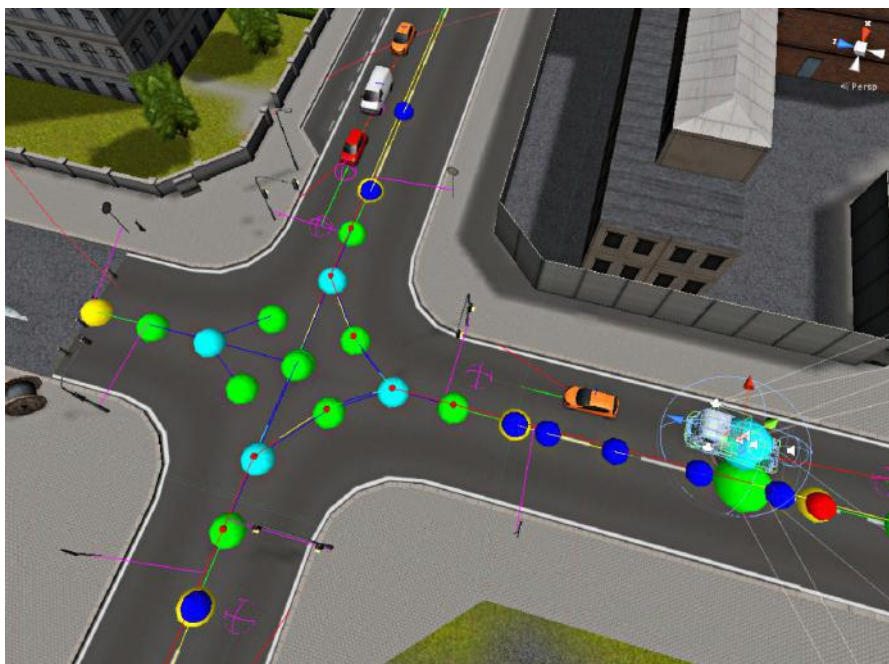


Abbildung 5: Eine Straßenkreuzung mit Ampelanlage und Auto-KI.

nen und Collidern als echter Performance-Killer, besonders in Kurven. Eine auch getestete und verworfene Idee war, diese vielen Collider durch ein unsichtbares Band zu ersetzen, das den LKW zur Straßenmitte zieht. Es kam zu sehr lustigen physikalischen Effekten, die dem Namen »Simulation« aber nicht gerecht wurden.

Kreuzungen und Autobahnabfahrten wurden in einem 3D-Modelliertool gebaut. Um nicht bei jeder Brücke das Terrain so formen zu müssen, dass es mit den Straßen genau zusammenpasst, wurde das Terrain in der Umgebung der Brücke als Geometrie erstellt. Beim Platzieren hat ein Skript das Unity-Terrain wie beim Sandkuchenbacken von unten an diese Form gepresst. Die Form wurde dann entfernt.

Die erste Implementierung der Auto-KI war physikalisch realistisch, überforderte aber den mobilen Prozessor. Ein in den Graben gerutsches Auto schaffte es sogar meist zurück auf die Straße. Bei der zweiten Implementierung werden die Autos an einer gedachten Schnur gezogen. Sie strauchelt an einem kleinen Detail: In jedem Frame für jedes Fahrzeug wird die Position auf einer parametrisierten Raumkurve berechnet (spline interpolation), was die CPU merklich belastet. Wir haben die Zahl der Autos daher auf 25 begrenzen müssen. Eine Verbesserung wäre eine lineare Interpolation zwischen vorberechneten dichten Kontrollpunkten auf der Kurve. Auch das permanente Zuordnen des dem LKW nächstliegenden Straßenabschnitts belastet den Prozessor. Dieses braucht man für das rechtzeitige Abbremsen der Autos und für die Wegfindung. Für die optimale Berechnung der kürzesten Entfernung von einem Punkt zu mehreren Raumkurven wurden Algorithmen aus wissenschaftlichen Artikeln untersucht und programmiert.

Wirklich nachteilig ist die oft geforderte Möglichkeit der Verschachtelung von Prefabs in Unity. Beim Prefab »Kreuzung« würde man gern die Ampelanlage sowie die einzelnen Ampeln als Prefabs anlegen. Meiner Meinung nach ist dies der einzige wirkliche Minuspunkt der Unity-Engine.

Leistung

Eine Kachel hat im Durchschnitt 500 bis 5.000 Objekte. Das Unity-4-eigene System zur Darstellungsgenauigkeit in Abhängigkeit der Kameraentfernung (level of detail) ist für diesen Fall zu langsam. Nahe beieinanderliegende Objekte wurden in einem eigenen System logisch gruppiert und gemeinsam je nach Kameraentfernung in regelmäßigen Zeitabständen ein- oder ausgeschaltet. Ein kleiner Nachteil ist, dass die Ploppdistanz der Objekte leicht variiert. Unity 5 hat das System verbessert.

Besonders in Städten macht sich diese Vielzahl der Game-Objekte in der Leistung negativ bemerkbar. Unity kann Objektverdeckungen für verschiedene Kamerastandpunkte zur Entwicklerzeit vorausberechnen (occlusion culling). Das erhöht die Framerate in Städten fast

auf das Doppelte! Zum großen Bedauern kann diese Funktion nicht bei asynchron geladenen Szenen angewendet werden. Im Asset Store gibt es einige Systeme für Occlusion Culling, die leider nicht unserem Zweck genügten.

Mit Weitsicht wurden die Modelle und Texturen erstellt. Sämtliche Häuser auf dem Land, in Industrievierteln und in Städten, nutzen jeweils nur einen relativ kleinen Texturatlas (**Abbildung 4**). Das ermöglicht schnelles Rendern und hält den Speicherbedarf niedrig. Einige Texturbereiche werden von verschiedenen Objekten verwendet. Je nachdem, in welchem Gebiet der Spieler unterwegs ist, sind idealerweise nur ein bis zwei dieser Atlanten gleichzeitig im Speicher. Dazu kommt ein Atlas für alle Verkehrsschilder. Ein Hochhaus könnte man prima mit kleinen kachelbaren Texturen tapezieren. Solche Texturen wurden nur ausnahmsweise für die Leitplanken und Ähnliches verwendet, um die Zahl der Texturen zu minimieren. Eigene oder aufwendige Shader wurden aus Leistungsgründen nur bei den Fahrzeugen und den Wasserflächen verwendet.

Als Leistungsoptimierung kann Unity die Geometrie mehrerer Objekte in ein großes Objekt (vertex buffer combination) zur Laufzeit zusammenfassen (draw call batching). Dabei gibt es zwei Varianten: die Variante, bei der Objekte zur Laufzeit nicht bewegt oder im Sichtbarkeitsstatus verändert werden (static batching), und die, bei der das nicht zutrifft (dynamic batching). Für die größtenteils unveränderlichen Objekte einer Landschaft bietet sich die erste Variante an. Dafür stellt Unity Bedingungen: Alle Objekte in einem Batch müssen das gleiche Material nutzen und dürfen nicht mehr als 300 Vertices haben. Dieser Schwellenwert hat den Grafikern nicht geschmeckt, ist aber wichtig. Notfalls muss ein aufwendiges Objekt in zwei kleine Objekte unterteilt werden. Transparente Objekte werden praktisch nicht gebatcht und sollten nur sehr sparsam auf Mobilgeräten eingesetzt werden. Weitere Bedingungen stehen in der Dokumentation. Aktiviertes »static batching« erhöht die Framerate signifikant. Einen Schritt weiter kann man mit dem Zusammenfassen der Geometrie zur Entwicklerzeit gehen. Dies würde die Framerate weiter erhöhen, wirkt aber der Auswahl von außerhalb der Kamera liegenden Objekten (frustum culling) entgegen und behindert die Anpassung an Kameraentfernung und Geräteleistung. Zwischen dem langsamsten und schnellsten unterstützten Smartphone liegt ein Leistungsfaktor von 4. Daher bieten wir dem Spieler drei Leistungseinstellungen an.

TruckSimulation 16 besteht nicht nur aus einer Fahrsimulation. Nicht eingegangen wurde auf den Aufbau der eigenen Logistikfirma durch den Bau von Filialen und dem Anstellen von Fahrern.

Abschließend lässt sich sagen, dass offene Spielwelten auf Mobilgeräten immer noch ein Abenteuer sind und die Entwicklungszeit wesentlich verlängern.

Christoph Ender

Truck-Simulation 16

TruckSimulation 16 versetzt den Spieler in die Rolle eines Logistikunternehmers. Dabei kann er selbst Lieferungen umherfahren oder später in einem Management-Part Mitarbeiter einstellen, die ebenfalls Fahrten übernehmen. Das Spiel umfasst sieben Trucks und große Teile von Mittel- und Westeuropa mit mehr als 20 Städten.

- **Plattformen:** iOS und Android
- **Erstveröffentlichung:** 25.11.15
- **Publisher:** astragon Entertainment GmbH
- **Facebook:** /TruckSimulationMobile
- **Website:** www.trucksimulation16.de



BODY PARTS AND MOTION SICKNESS IN VR

MITIGATING MOTION SICKNESS WITH LEGS

A lot of VR users complain about motion sickness and similar discomforts. In a scientific study Dr. Matthew White tried to find out if missing body parts in games might be the reason for those troubles.



Dr. Matthew White
is Data Scientist
at Deep Silver Volition.

Dr. Matthew White is Data Scientist at Deep Silver Volition. A former academic, Dr. White helps Volition embed telemetry in titles, mine player data, perform quantitative analysis of playtests, perform A/B tests, and apply statistics wherever they may be needed. Dr. White has previously chaired content for the IGDA GURsig's yearly summit in 2015, a conference at which he has presented many times. He's a member of the user research SIG, and just recently spoke at #GAMESUR London, an IGDA event. He developed Penn State University's game development program, and was previously an IGDA Scholar (2009). Before Penn State, Dr. White ran two small startups in Canada, and did private usability consulting during his Ph.D., which he obtained from Memorial University of Newfoundland.

In the last few years, we have seen a pretty hefty increase in the number of VR systems, solutions, and peripherals that are available to consumers and even the most junior of developers. Starting with the meteoric popularity of the Oculus Rift, a bevy of followers has emerged. Unfortunately though, for a section of the population, none of the research and development into these cool new toys makes much of a difference.

I'm talking of course about those who are made horribly ill by the phenomenon known as simulator sickness. Simulator sickness, a form of motion sickness, is a sort of no-fun phenomenon in which individuals using a simulation or VR device get sick in a variety of unpleasant and often chunky ways. Some of the symptoms include nausea, vomiting, dizziness, lightheadedness, headaches, fainting, existential ennui, forgetting to call your mother, and more. Needless to say, these kind of hangover-esque phenomena keep people from playing games, using simulations, and so on. For this reason, my Penn State student Dylan Poljak (www.yellowdragon.net) and I were very interested in looking for a computationally inexpensive solution to what has become a pervasive problem. This study and the simple software

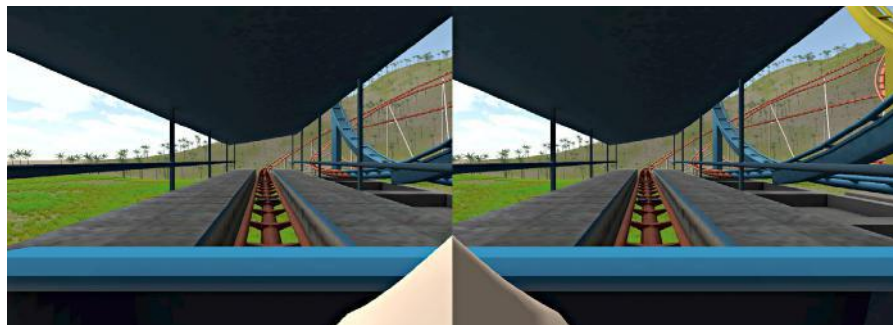
solution it presents were born out of observations of students in my lab, who after putting on the Oculus consistently looked up, then directly down. Our thought process was something like: »What if they're looking for legs?«

Background

There are myriad theories regarding the underlying cause of motion sickness, some of which involve the eyes and a triggering of the vagus nerve. In fact, many of the symptoms of vasovagal syncope (e.g. fainting) are very similar to those felt in simulation sickness – although the syncope symptoms usually do not indicate that precipitous vomiting is imminent – rather, they indicate that you are about to be more horizontal than you may have intended when donning the VR device. A few other studies have made a concerted effort at mitigating the symptoms associated with this illness with varying degrees of success.

One of the studies that inspired our thinking the most was Nasum Virtualis, a study from Purdue University that investigated the role that a virtual nose could play in delaying simulator sickness onset in players¹. The study found that adding a virtual nose to a simulator significantly reduced, or at least delayed simulator sickness in players. The study reported that individuals were able to maintain play without simulation sickness symptoms longer in a situation in which a simple nose was placed in front of the players' faces while they used both the Tuscany Demo packaged with the Oculus Rift, and a simulated roller coaster. This got us thinking – if a nose in the simulator significantly reduced player perceptions of sickness – what might other body parts do?

We started reflecting on our experiences with other games, like »Half-Life«, where for some darned reason, you just don't have legs. One of the real fourth-wall-breakers for me has always been looking down at my feet and just having no lower body to speak of. We made an



One of the studies that inspired the team the most was »Nasum Virtualis«, a study from Purdue University that investigated the role that a virtual nose could play in delaying simulator sickness onset in players.

educated guess (read: guess) that the sickness may be elicited by a vision-perception disconnection. That is to say, motion is seen (e.g. the game visualizes motion), but the game breaks the perceptive immersion by not presenting arms, legs, etc. to the player. Additionally, the visualized motion is not felt by the player in any appreciable way – resulting in a disconnection that has led to sickness in the past².

We decided we'd cook up a study and investigate how the presence of legs affected simulator sickness, if at all, knowing that this was the easier problem to fix. Simulating the feeling of motion is, after all, a bit difficult for two people.

Study and Software Design

We decided to design a traditional random study. During the process, we used Unity to create a 3D environment through which we would expect players to navigate. The objective was to find a ball at the end of an exploratory environment. Sure, not the best game ever, but we were more concerned with players' urge to barf than their enjoyment of the game.

The majority of the environment was created with Asset Store parts, and the assembly took about a week. We also decided that we would employ the same questionnaire used in previous studies, the Simulator Sickness Questionnaire by Kennedy et al³. This instrument has been used extensively and is a fairly sensitive and standard instrument for these types of tests.

Methodology

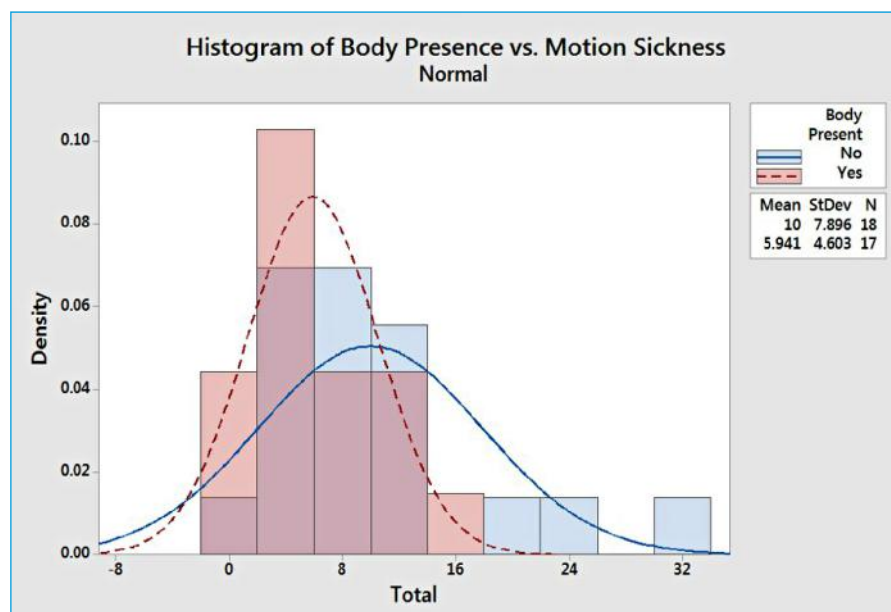
We recruited as many participants as a student and a professor reasonably could in a few weeks, and decided we should randomly split them into two groups – one group would get a body at which they could look (e.g. torso, legs etc.), and one would not. After playing for a bit, we'd give the participants the questionnaire, and review their responses to see if there were any meaningful differences.

Thankfully, the effect size was small enough to require the instrument. That's academic speak for: nobody outright barfed, so we needed a survey to tell if there was any difference or not. It wouldn't require much investigative work if 100 percent of people without legs shot projectile vomit on the wall, now would it?

Findings

As we originally expected, the group with legs had significantly less sickness than the group without legs. We're smart like that. Sometimes. In any case, the effect size was relatively large (about 30 per cent). Like most academic findings, however, we should be careful what we do with this data. The sample size was small, and the sample was collected from individuals who are already familiar with games and technology. As a result, these individuals might be faster technology adopters, and so on.

Nonetheless, this stands to further previous research indicating that the presence of body



As originally expected, the group with legs had significantly less sickness than the group without legs.

parts has a mitigating effect on simulator sickness, significantly reducing number of chunks per hour. This potentially disabling condition makes it nearly impossible for people to use VR, joking aside.

For nerds: the correlation between body presence and sickness was 0.306 using Pearson's r with a p-value of 0.074 (yes, not quite 0.05, but I'm comfortable with that given the preliminary nature of the study, get off my case, ok?). We had 35 participants, and these results seem to be replicable.

Conversation

While the effect size here is considerable, it's worth noting that we had very few people in the study (35). That said, the study was somewhat significant for the size, and we're confident that repeated tests and more sensitive tests that use a wider demographic would continue to validate that point. As with all things academic, further study is needed. In the meantime, the inclusion of a body part in the game seems to further the findings of Nasum's Virtualis: namely, that including elements of a person's real body seems to slow or mitigate the effects of motion sickness. We'd like to run this again using Nasum's methodology to check on time to onset of symptoms when presented with a virtual body.

In the meantime, developers can potentially reduce player churn by including simple fixes. Chances are the player model has been sculpted to some degree for use in cutscenes, cinematics etc., in a three-dimensional game. Where possible, developers should spend the extra resources to provide visible limbs and other body parts in games designed for virtual reality devices, if only to keep them free of sweat, tears, and food particulates.

Dr. Matthew White



The majority of the test level environment was created with Asset Store parts in Unity, and the assembly took about a week.

References

If you want to read more about the references that are mentioned in the text, please see the following publications and links:

1. 'Virtual nose' may reduce simulator sickness in video games by Emil Venere, David Whittinghill
www.makinggames.biz/VirtualNose
2. So, R.H.Y. and Ujike, H. (2010) Visually induced motion sickness, visual stress and photosensitive epileptic seizures: what do they have in common? – Preface to the special issue. *Applied Ergonomics*, 41(4), pp.491-393.
3. Simulator Sickness Questionnaire, Kennedy, Lane, Berbaum, & Lilienthal (1993)
www.makinggames.biz/Simulator-Sickness



FREI SKALIERBARE BACKEND-ARCHITEKTUR

MITHILFE VON ACTOR MODEL UND SERVICE FABRIC

Für sein neues Spiel begab sich das Team hinter dem OnlineFussballManager auf die Suche nach einer geeigneten Backend-Architektur und zeigt hier, welche Technik sich für welche Einsatzgebiete eignet.



Kevin Daniel Setiono
ist Senior Backend Developer
bei der OnlineFussballManager
GmbH.

Kevin arbeitet seit fast zehn Jahren als Full-Stack-Entwickler und Consultant für unterschiedliche Branchen. Als Consultant hat er langjährige Erfahrungen mit der Azure Cloud und der Erstellung von Konzepten für hoch skalierbare Backend-Anwendungen. Seit 2015 ist er bei der OnlineFussballManager GmbH angestellt. In der Spielebranche ist er in dem Projekt »Football Empire« für die Konzeption und Umsetzung der Backend-Systeme verantwortlich.



Andreas Pohl
ist Game Developer und
Scrum Master
bei der OnlineFussballManager
GmbH.

Andreas hat sein Studium an der Hochschule Trier als Bachelor of Science im Bereich Digitale Medien und Spiel mit der Spezialisierung auf Spieleprogrammierung abgeschlossen. Während seines Studiums war er als Projektleiter in der 4Players Liga, der eSport-Plattform der 4Players GmbH tätig. Andreas arbeitet seit 2014 bei der OnlineFussballManager GmbH als Backend-Entwickler am OFM. Mit dem Projektstart von »Football Empire« hat er die Planung und Organisation des Backends übernommen.

Mitte des letzten Jahres haben wir, die OnlineFussballManager GmbH, mit der Produktion unseres neuen Spiels begonnen. Das Ziel ist es, einen Fußball-Manager zu entwickeln, der einen klassischen Manager mit dem Thema Aufbastrategie verbindet. Im Gegensatz zu unserem Bestandsprodukt, dem OnlineFussballManager (OFM), der vor über zwölf Jahren als Browsergame gestartet ist, fokussieren wir uns bei unserem neuen Projekt unter dem Titel »Football Empire« auf Android, iOS und Windows Phone (ggf. Windows 10) mit Unity.

Aufgrund der langjährigen Weiterentwicklung des Browsergames OFM besitzen wir Kompetenzen in der Konzeptionierung und Umsetzung von Backend-Architekturen. Diese Erfahrungen haben uns geholfen, eine solide Lösung für die Backend-Architektur von Football Empire zu finden.

In diesem Artikel möchten wir die Erfahrungen und Erkenntnisse aus unserer intensiven Recherche teilen und die von uns gewählte Lösung vorstellen.

Problemstellung

Bei der Entwicklung einer neuen Backend-Architektur und Infrastruktur sind viele Dinge zu beachten. Vor allem Informationen zu den potenziellen Nutzern und die zu erwartende Last sind ausschlaggebend für die Architektur

des Backends. Wer möchte schon, dass das eigene Spiel nicht mehr erreichbar ist, wenn gerade ein TV-Werbespot auf einem großen Sender gelaufen ist und auf einmal viele neue Nutzer in das Spiel kommen? Natürlich passieren solche Events nicht zufällig und man kann sich als Betreiber darauf vorbereiten. Dennoch muss das Backend skalierbar genug sein, um auch solche vorübergehenden Lastspitzen bewältigen zu können. Darüber hinaus ist in einer agilen Umgebung davon auszugehen, dass sich die Anforderungen im Laufe des Projektes ändern werden, was es sehr schwer macht, einen Kompromiss zwischen Kosten, Komplexität und Verfügbarkeit zu finden. Aus diesem Grund und zugleich wegen fehlender Flexibilität werden neue Backend-Systeme häufig auf die erwarteten Lastspitzen ausgelegt und optimiert. Dies ist in der Mehrzahl der Fälle nicht kosteneffizient, weil so immer für die Lastspitzen und nicht für die aktuelle Last gezahlt wird. Das andere Extrem sind Luxusprobleme, wie zum Beispiel »Rocket League«, bei dem die Entwickler den massiven Erfolg nicht erwartet hatten und das Backend dem enormen Nutzeransturm einfach nicht gewachsen war. Doch auch wenn man theoretisch in der Lage ist zu skalieren, muss dies meist manuell durch eine Person im Unternehmen übernommen werden. Dabei sprechen wir für gewöhnlich von DevOps- oder System-Administratoren. Und weil wir keinen dedizierten DevOps Engineer im Team haben, musste unsere Lösung dies berücksichtigen.

Ziel

Mit Football Empire sind wir in der seltenen Lage, das Backend komplett neu zu entwickeln und so die bekannten Problemstellungen mithilfe einer modernen Umgebung zu lösen. Bei der Entwicklung sind uns folgende Akzeptanzkriterien besonders wichtig:

- Skalierbarkeit der Infrastruktur (100.000 – 10.000.000)
- Kostenminimierung der Infrastruktur
- Minimieren des Wartungsaufwandes aufgrund begrenzter DevOps-Ressourcen
- Sicherstellen einer schnellen Entwicklung
- Spieler sollen zur Laufzeit Änderungen an ihren Teams vornehmen können (etwa das Auswechseln eines Spielers während eines Spiels)

Weil wir begrenzte Ressourcen haben, ist es uns wichtig, dass die Entwicklung in einem überschaubaren Zeitrahmen und von wenigen Personen erfolgen kann. Um dies sicherzustellen, müssen wir die Aufgaben, mit denen sich das Team zu beschäftigen hat, so klein wie möglich halten. Das widerspricht ein wenig der Anforderung einer hoch skalierbaren Infrastruktur, weil diese meist sehr komplex und wartungsintensiv ist.

Infrastruktur-Übersicht

Am Anfang der Entwicklung haben wir zunächst mit der Evaluierung von klassischen Architektur-Lösungen begonnen. So könnten wir einen Monolith entwickeln und diesen horizontal einfach durch mehr Hardware skalieren. Einen ähnlichen Ansatz haben wir im OFM verfolgt und wissen daher, dass unsere Anforderungen kaum oder nur sehr schwer zu halten sein werden. Der Grund dafür ist die fehlende Fähigkeit, einzelne Teile des Monoliths zu skalieren. Wenn, dann muss immer der gesamte Monolith skaliert werden, was in der Regel zu einem Bottleneck in der Persistenz-Schicht führt. Somit kommt ein Monolith bei unseren Anforderungen nicht infrage.

Eine Trennung der Architektur in verschiedene Schichten wie zum Beispiel Queues, Worker, Cache und Persistenz würde das System flexibler halten und wäre als Service darstellbar. Dabei dienen die Queues dem Entkoppeln der Worker und der Persistenz. Bei dieser Lösung wäre man in der Lage, nur bestimmte Worker oder Queues zu skalieren, was zu einer höheren Flexibilität führt. Allerdings sprechen wir hier im schlimmsten Fall von einer Vielzahl von verschiedenen Systemen, die es zu verstehen und zu warten gilt. Aufgrund der Erfahrungen, die wir mit dem Browsergame OFM sammeln konnten, wissen wir, dass eine solche Architektur einen nicht zu vernachlässigenden Wartungsaufwand mit sich bringt. Um einen stabilen Service zu gewährleisten, müssten wir unser Team um einen dedizierten DevOps Engineer erweitern, womit auch diese Lösung nicht optimal für uns ist.

Moderne Infrastruktur

Die erste Frage, die sich einem stellt, ist: Sollte man eine bewährte Lösung anzweifeln? Nun ja, gerade in der Informatik befinden wir uns in einem Bereich, der sich kontinuierlich und schnell weiterentwickelt. Auch wenn neue Lösungen nicht immer besser sind als etablierte Konzepte, so haben wir uns gefragt, ob es nicht effizientere Technologien gibt, um unser Backend zu entwickeln.

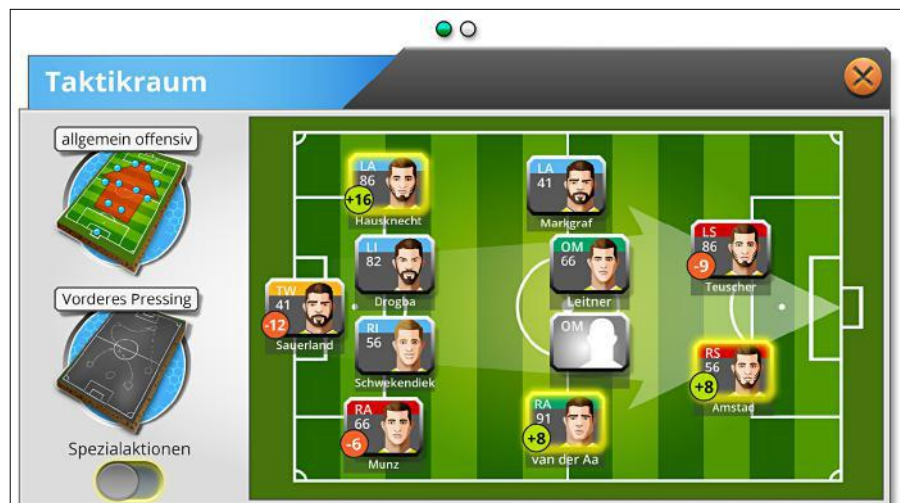
Wenn man heutzutage über die Entwicklung von verteilten Systemen oder hoch skalierbaren Systemen spricht, dann kommt man nicht mehr um das Thema Microservices herum. Besonders durch die Erfolgsgeschichten, die man immer wieder über Spotify, Netflix und Co liest, scheint der Einsatz von Microservices der neue, moderne Heilige Gral zu sein.

Microservices

Zu Beginn unserer Entwicklung haben wir uns gefragt, ob Microservices wirklich das Allheilmittel unserer Herausforderungen sein werden und in welchen Punkten sie für unseren Anwendungsfall relevante Vorteile bringen würden.

Im einfachsten Fall verbindet ein Microservice Logik und Daten, um eine Schnittstelle nach außen bereitzustellen. So kann jeder Microservice seine eigene Datenbank samt den Daten haben, mit denen er arbeitet. Dadurch kann eine hohe Skalierbarkeit erreicht werden.

Wenn wir uns Microservices vom Blickpunkt der Software-Architektur nähern, dann würde man sich im optimalen Fall starke Module Boundaries wünschen. Eine Microservice-Architektur unterstützt diesen Grundsatz, indem ein Service keinen Zugriff auf die Daten eines anderen Services hat, außer über die Service-Schnittstelle selbst. Der Vorteil durch diese klaren Grenzen zeigt sich vor allem in großen Teams. Damit erzwingen Microservices von Natur aus eine saubere Architektur. Das ist auch ohne Weiteres in einem Monolith möglich, erfordert aber eine wesentlich höhere Disziplin des Teams. Im Gegensatz zu einem



Das Team hinter dem OnlineFußballManager arbeitet an einer Mischung aus Fußball-Manager und Aufbaustrategie. Der Football Empire getaufte Titel erscheint auf Mobilien Geräten.

Monolith erhöht der Microservice-Ansatz allerdings die Wahrscheinlichkeit auf eine höhere Modularität der Applikation.

Weil Microservices in der Regel relativ klein sind und immer nur eine konkrete Aufgabe sowie ihre Daten selbst verwalten müssen, sind sie schnell und unabhängig voneinander veröffentlicht, was dem Continuous-Integration-/Deployment-Gedanken zugutekommt. Zudem kommunizieren Services nur über ihre Schnittstellen miteinander, weshalb jeder Service in einer anderen Programmiersprache entwickelt werden kann, was zu einer höheren Flexibilität führt.

Auch wenn Microservices einige Vorteile bieten, so sind sie immer noch ein verteiltes System – und ein solches bringt von Natur aus eine sehr hohe Komplexität mit sich. In einem auf Microservices basierendem System kommunizieren Services durch Remote Calls miteinander, die immer mit Kosten und dem Risiko des Fehlschlags verbunden sind.

Sollte ein Service also nicht mehr erreichbar sein, so muss sich das System selbst wiederherstellen und wie gewohnt funktionieren. In der Regel ist dafür der Entwickler verantwortlich.

Als Ausgleich für die entstehende Latenz durch Remote Calls wird in den meisten verteilten Systemen sehr viel Wert auf asynchrones Abarbeiten gelegt, wodurch die Fehleranalyse deutlich erschwert wird.

In einem Microservice-basierten System müssen wir aufgrund der Trennung von Daten auch mit Eventual Consistency leben, weil es kaum möglich ist, eine starke Konsistenz in einem verteilten System zu erzeugen. Das kann unter Umständen durchaus zu einem Problem werden, denn die Replikation von Daten in verteilten Systemen benötigt immer eine gewisse Zeit.

Service oder Microservice

Bisher gibt es noch keine einheitliche Definition, was eigentlich ein Microservice genau ist oder wie er auszusehen hat. Wir würden das Konzept von Microservices als Teilmenge von Service-oriented Architecture (SOA) einordnen, um eine Definition zu erhalten. Demnach

sollten Microservices mindestens folgende Eigenschaften erfüllen:

- Eindeutige Application Boundaries
- Services sind autonom
- Services teilen Schema und Contracts, keine Klassen
- Service compatibility is based on policy

Nach Martin Fowler unterscheiden sich Microservices von SOA mitunter dadurch, dass sie unabhängig von anderen Services updatierbar sind und jede Business-Anforderung in einem eigenen Service abgebildet ist.

In der Praxis sieht das so aus: Nehmen wir an, dass ein Team-Service existiert, der dafür zuständig ist, ein Team aus Fußballspielern zu verwalten. Nehmen wir weiterhin an, dass ein Kicker niemals ohne Team existieren kann. Wenn wir den SOA-Weg gehen, dann kann man das wie in **Abbildung 1** visualisieren.

In **Abbildung 2** bilden wir jeden Bereich unserer Logik in einem separaten Service ab. Somit ist der Kicker-Service selbst dafür verantwortlich, einen neuen Spieler zu generieren, und stellt nur die entsprechenden Schnittstellen bereit.

Wann sollte man Microservices nutzen?

Zusammenfassend kann man sagen, dass man sich die Skalierbarkeit durch eine erhöhte Komplexität eines verteilten Systems erkaufen kann. Wir mussten uns also entscheiden, ob wir das Risiko einer erhöhten Komplexität des Systems in Kauf nehmen wollen. Denn einen wirklichen Vorteil bietet diese Lösung nur, wenn die Komplexität eines Monoliths sehr hoch ist. Eine Trennung in Microservices lohnt sich für einen trivialen Monolith nicht, weil der Komplexitäts-Overhead zu groß wäre. Außerdem muss von vornherein klar sein, welche Lastanforderungen es gibt, um effiziente Microservices zu entwickeln.

Wenn man sich diese Fragen stellt, dann ist das in der Regel ein Indikator dafür, dass eine Trennung auf Microservices erst mal nicht sinnvoll ist.

Die Lösung?

Wir wissen also, dass wir die erhöhte Komplexität von verteilten Systemen nicht unbedingt in Kauf nehmen möchten, wollen aber gleichzeitig unseren Anforderungen der Skalierbarkeit entsprechen. Ebenso wissen wir, dass eine hohe Skalierbarkeit mit den etablierten Ansätzen zwar möglich, jedoch nicht trivial ist. Unser Lead Game Designer und Product Owner Guido Schmidt hat es passend in folgende Worte gepackt: »Wir haben die Kompetenz, aber nicht die Erfahrung in solchen Systemen.«

Wir brauchen folglich ein System, das uns die Vorteile von verteilten Systemen bietet und gleichzeitig die Komplexität solcher Systeme verringert, indem sinnvolle Standardverhalten angewendet werden.

Schließlich sind wir durch ausführliche Analysen auf das »Actor Model« gestoßen, das unsere Probleme vielleicht lösen könnte.

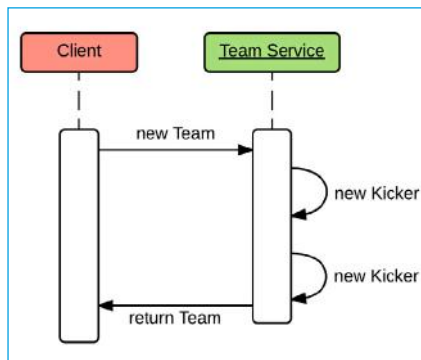


Abbildung 1: Beispiel eines Team-Services als SOA.

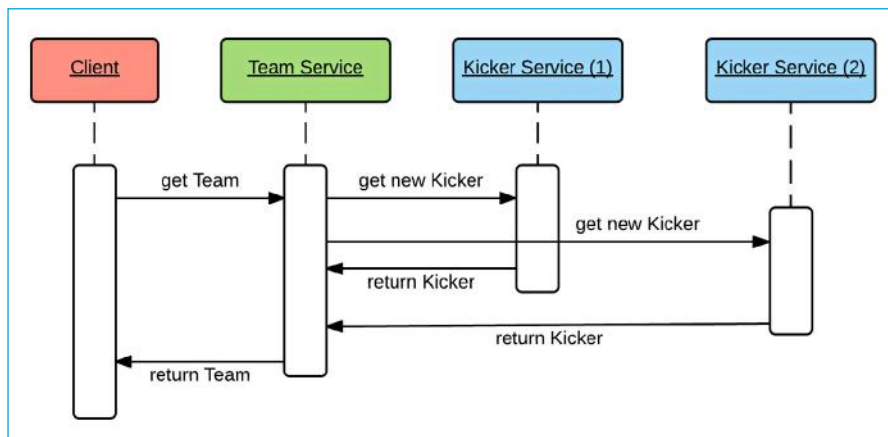


Abbildung 2: Beispiel eines Team-Services als Microservice.

Actor Model

Das Actor Model ist ein Modell für die Entwicklung von nebenläufiger Software. Die Nebenläufigkeit beziehungsweise Parallelität einer Software wird immer wichtiger, um eine schnelle Anwendung zu gewährleisten. Die Ausführung von Software wurde immer schneller, weil die Hardware in den letzten Jahren immer schneller wurde. Zurzeit ist jedoch zum Beispiel bei der CPU eine physikalische Grenze erreicht, in der es nicht mehr möglich ist, die Taktrate zu erhöhen. Um die Software zu beschleunigen, werden die Prozesse mit Multithreading parallelisiert, die simultan mithilfe mehrerer Kerne in der CPU abgearbeitet werden können. Um Daten zwischen den Threads auszutauschen, wird ein geteilter Zustand verwendet. Die Herausforderung dabei ist folgende: Wenn ein Thread auf diesen State »schreibend« zugreift, muss ein anderer Thread warten, bis ersterer wieder freigegeben ist.

Dieser geteilte Zustand wird in der Informatik »Mutable Shared State« (veränderbarer geteilter Zustand) genannt. Das Warten auf die Zugriffsrechte vom Mutable Shared State wird mithilfe von Lock und einer Critical Zone erreicht. In einer Critical Zone darf immer nur ein Prozess oder Thread sein. Dies lässt sich im Quellcode (zum Beispiel C#) durch einen Lock erzielen. In **Abbildung 3** vergleichen wir eine Klasse »Counter«, die »thread-safe« ist (es können mehrere Threads auf die Klasse zugreifen), mit derselben Funktionalität in **Abbildung 4** als Actor.

Was ist ein Actor?

Es gilt die Philosophie »Everything is an actor«, die analog zu »Everything is an object« in der objektorientierten Programmierung (OOP) ist. Dies ermöglicht Softwareentwicklern einen einfachen Einstieg in die Entwicklung von Actor-basierten Systemen.

Ein Actor ist eine Einheit für Berechnungen, die den eigenen State verwaltet. Um Daten

```
namespace Ofm.Example
{
    /// <summary>
    /// Counter class
    /// </summary>
    public class Counter
    {
        /// <summary>
        /// The lock object
        /// </summary>
        public object lockObject = new object();

        /// <summary>
        /// The counter
        /// </summary>
        private int counter;

        /// <summary>
        /// Adds one to the counter.
        /// </summary>
        public void AddCounter()
        {
            lock (this.lockObject)
            {
                this.counter++;
            }
        }

        /// <summary>
        /// Gets the counter.
        /// </summary>
        public int GetCounter()
        {
            lock (this.lockObject)
            {
                return this.counter;
            }
        }
    }
}
```

Abbildung 3: Eine Klasse für das Verwalten eines Counters (thread-safe).

```
namespace Ofm.Backend.ActorBackend.actors
{
    /// <summary>
    /// Counter Actor
    /// </summary>
    public class CounterActor : StatefulActor<CounterState>, ICounterActor
    {
        /// <summary>
        /// Adds one to the counter.
        /// </summary>
        public Task AddCounter()
        {
            this.State.Counter++;

            return Task.FromResult(true);
        }

        /// <summary>
        /// Gets the counter.
        /// </summary>
        [ReadOnly]
        public Task<int> GetCounter()
        {
            return Task.FromResult(this.State.Counter);
        }

        /// <summary>
        /// Called when [activate async] is called, initializes the State.
        /// </summary>
        /// <returns></returns>
        protected override Task OnActivateAsync()
        {
            if (this.State == null)
            {
                this.State = new CounterState();
            }

            return base.OnActivateAsync();
        }
    }
}
```

Abbildung 4: Ein Actor für das Verwalten eines Counters (thread-safe).

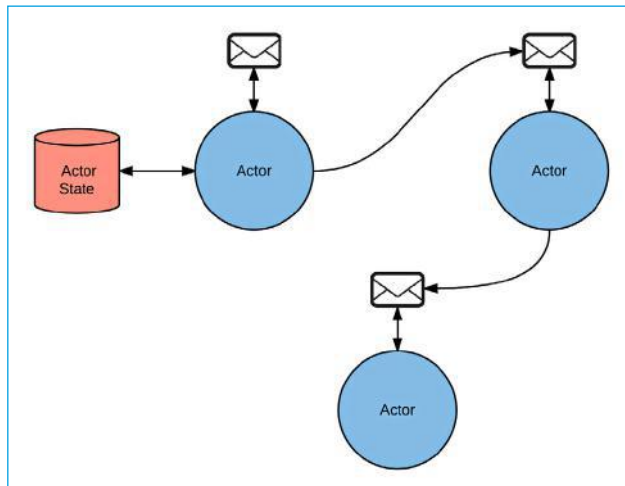


Abbildung 5: Eine Visualisierung, wie Actor-Systeme aussehen.

zwischen verschiedenen Actors auszutauschen, werden immutable (unveränderbare) Nachrichten versendet. Die Nachrichten werden asynchron versendet und in einer Mailbox zwischengespeichert. Die Mailbox dient als Warteschlange für Anfragen an den Actor. Eine Nachricht in der Mailbox ist analog zu einem Function Call in der Programmierung und lässt sich auch ähnlich programmieren. Ein Actor kann nach dem Empfangen einer Nachricht folgende Aktionen ausführen:

- eine endliche Anzahl von Nachrichten an andere Actors versenden
- eine endliche Anzahl von neuen Actors erstellen
- seinen eigenen State verwalten

Frameworks für das Actor Model

Es gibt viele Frameworks für das Entwickeln eines verteilten Systems nach dem Actor Model. Die bekanntesten sind Akka, Orleans, Service Fabric und Orbit. Wir haben verschiedene Frameworks für unser verteiltes System evaluiert, wobei nur ein paar davon vorgestellt werden.

Akka

Akka (<http://akka.io>) ist ein Open-Source-Projekt unter der Apache-2-Lizenz. Es dient dem Erstellen von nebenläufiger Software in einem verteilten System. Akka ist für die Java Virtual Machine, obgleich es auch eine .NET-Portierung gibt. Der Hauptfokus von Akka ist es, ein Framework zu liefern, bei dem der Softwareentwickler alles im System verändern kann. Sprich, das Framework versteckt die Komplexität von verteilten Systemen nicht, um dem Softwareentwickler alle Möglichkeiten zu bieten. Der DevOps-Aufwand eines Akka-Systems ist laut unserer Analyse hoch.

- **Actor Creation & Load Balancing:** Die Actors werden von ihrem Elternobjekt erstellt. Das heißt, alle Actors müssen explizit gestartet und gestoppt werden. Der Ersteller hat somit auch die Möglichkeit, dem Actor – bei der Erstellung – Parameter mitzugeben. Es

muss explizit angegeben werden, wo sich der neu erstellte Actor befindet, wobei dieses Verhalten auch über die Konfiguration verändert werden kann. Das hat zur Folge, dass der Softwareentwickler auf das Load Balancing achten muss.

- **Persistence:** Der State eines Actors wird mittels Event Sourcing persistiert. Beim Event Sourcing werden die Nachrichten gespeichert, die der Actor erhalten hat. Sollte der Actor seinen State wiederherstellen müssen, werden die Nachrichten nochmals eingespielt. Aus Performance-Gründen gibt es die Möglichkeit, einen Snapshot zu verwenden. Bei einem Snapshot werden nur die Nachrichten nach diesem Snapshot eingespielt. Es gibt Plugins für Akka, die das Verhalten vereinfachen.
- **Timer and Reminder:** Akka unterstützt Timer für einen Actor nur, solange der Actor aktiv ist. Wenn der Actor momentan nicht verwendet wird, dann wird der Timer auch nicht weiter ausgeführt. Das Gegenstück zum Timer wären Reminder. Diese werden ausgeführt, auch wenn der Actor nicht mehr aktiv ist, und aktivieren den Actor zu gewissen Zeitpunkten. Reminder werden zurzeit nicht von Akka unterstützt.

Orleans

Orleans (<https://github.com/dotnet/orleans>) ist ein Open-Source-Projekt, das von der Community weiterentwickelt wird. Es bietet Nichtexperten die Möglichkeit, effiziente, skalierbare und hoch verfügbare, verteilte Systeme zu entwickeln. Orleans implementiert ein Virtual Actor Model und ist angedacht, in der Cloud verwendet zu werden. Das Virtual Actor Model geht davon aus, dass ein Actor immer existiert. Sollte ein Actor nicht existieren, dann wird dieser automatisch erstellt. Im Gegensatz zu Akka wird einem bei Orleans viel Arbeit vom System abgenommen.

- **Actor Creation & Load Balancing:** Ein Actor ist virtuell in Orleans. Das heißt, das System sorgt dafür, dass ein benötigter Actor immer existiert. Die Actors werden automatisiert erstellt und über das Cluster verteilt. Von Haus aus bietet das System somit eine gute Verteilung über den Cluster, selbst wenn der Softwareentwickler in Themen wie Load Balancing unerfahren ist.
- **Persistence:** Der State eines Actors wird in Orleans mithilfe von Snapshots persistiert. Der Softwareentwickler muss genau angeben, wann der State persistiert werden soll. Der State wird dann als Snapshot abgelegt und wieder geladen.
- **Timer and Reminder:** Orleans unterstützt Timer und Reminder. Timer sind nur aktiv, solange der Actor aktiv ist; wobei Reminder einen Actor aufwecken, auch wenn dieser vorher aus dem Speicher entfernt wurde. Orleans hat hier im Vergleich zu Akka den Vorteil von Remindern.

Service Fabric

Microsoft hat vor kurzem Service Fabric (<https://azure.microsoft.com/en-us/services/service-fabric/>) als »Public Preview« freigegeben. Service Fabric hat viele Parallelen zu Orleans. Zum Beispiel ist Service Fabric ebenfalls ein Virtual Actor Model. Service Fabric bietet neben Actor-Entwicklung auch die Entwicklung von Reliable Services. Die Services sind effizient im Verwalten von Listen oder können für Konfigurationen verwendet werden.

Service Fabric ist im Gegensatz zu Orleans und Akka nicht Open Source, dafür ist die Integration von Service Fabric in die Azure Cloud oder auf dem Windows Server (Cluster) intuitiver.

- **Actor Creation & Load Balancing:** Service Fabric verwendet ebenfalls ein Virtual Actor Model und ähnelt in dem Verhalten der Actor Creation und des Load Balancings Orleans. Es können aber für die Orchestration auch Custom-Metriken festgelegt werden, damit zum Beispiel rechenintensive Actors nur auf der leistungsfähigeren Node erstellt werden. Ansonsten werden Actors immer als reliable und vorhanden angesehen.
- **Persistence:** Der State eines Actors wird in Service Fabric nach fast jeder Nachricht persistiert. Es gibt die Möglichkeit, eine Nachricht als »read-only« zu deklarieren, wodurch der State nicht neu persistiert wird. Es ist empfehlenswert, Read-only-Nachrichten zu verwenden, um die Performance zu steigern. Der Softwareentwickler kann zwischen einem »Volatile State Provider« und einem »Key-value Store State Provider« wählen. Es besteht auch die Möglichkeit, einen eigenen State Provider zu erstellen, wobei es hier leider noch keine brauchbaren Beispiele gibt. Der »Volatile State Provider« hält den State eines Actors nur im Speicher und verliert diesen, wenn der Actor inaktiv wird. Im Gegensatz dazu persistiert der »Key-value Store State Provider« den State auf der Festplatte und behält den State, auch nachdem ein Actor inaktiv wurde. Der State wird nach dem Abarbeiten einer Nachricht (Function Call) ausgeführt, wobei der Softwareentwickler wieder die Möglichkeit hat, eine Nachricht als »read-only« zu deklarieren.
- **Timer and Reminder:** Service Fabric unterstützt Timer und Reminder analog zu Orleans. Timer sind nur aktiv, solange der Actor aktiv ist; wobei Reminder einen Actor aufwecken, auch wenn dieser vorher aus dem Speicher entfernt wurde.

Welches Framework verwenden wir?

Wir haben in unserem Team Kompetenzen im DevOps-Bereich, wobei die Erfahrungen in diesem Bereich nicht ideal sind. Akka hat im Vergleich zu Orleans und Service Fabric die meisten Möglichkeiten, aber auch die meiste Verantwortung, diese auszuschöpfen. Die Portierung von Akka mit Akka.NET war noch nicht stabil, sodass wir uns gegen Akka entschieden haben.

Orleans ist ein sehr stabiles und »battle-tested« Framework, das weniger DevOps-Hürden als Akka hat. Wir haben uns dennoch eher für das »neuere« Framework Service Fabric entschieden. Wir haben sehr gute Erfahrungen mit dem Support von Microsoft in Richtung Service Fabric und Azure Cloud bekommen. Christian Geuer-Pollmann, Thorsten Rendel und Damir Dobric haben uns bei der Entscheidung für Service Fabric begleitet. Außerdem bietet Service Fabric sinnvolles Standardverhalten für die meisten Fälle, wodurch uns viele Sorgen abgenommen werden. Die Upgrade/Downgrade-Mechanismen von Software im Service-Fabric-Cluster sind über die Deployment-Tools einfach durchführbar. Unterm Strich kann gesagt werden: Wir halten Service Fabric für die beste Komplettlösung.

Wir lassen unseren Service-Fabric-Cluster in der Azure Cloud hosten. Die Azure Cloud in Verbindung mit Service Fabric ermöglicht es uns, dass die Nodes automatisch skalieren. Zur-

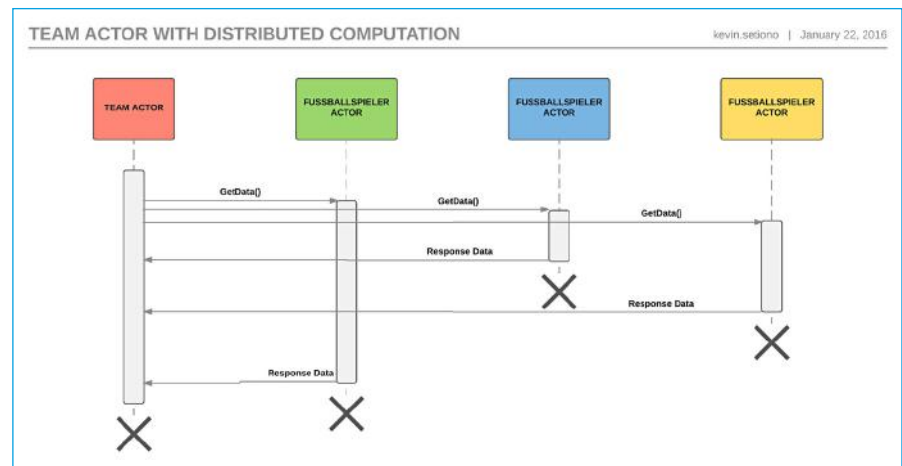


Abbildung 6: Ein Team-Actor mit Distributed Computation.

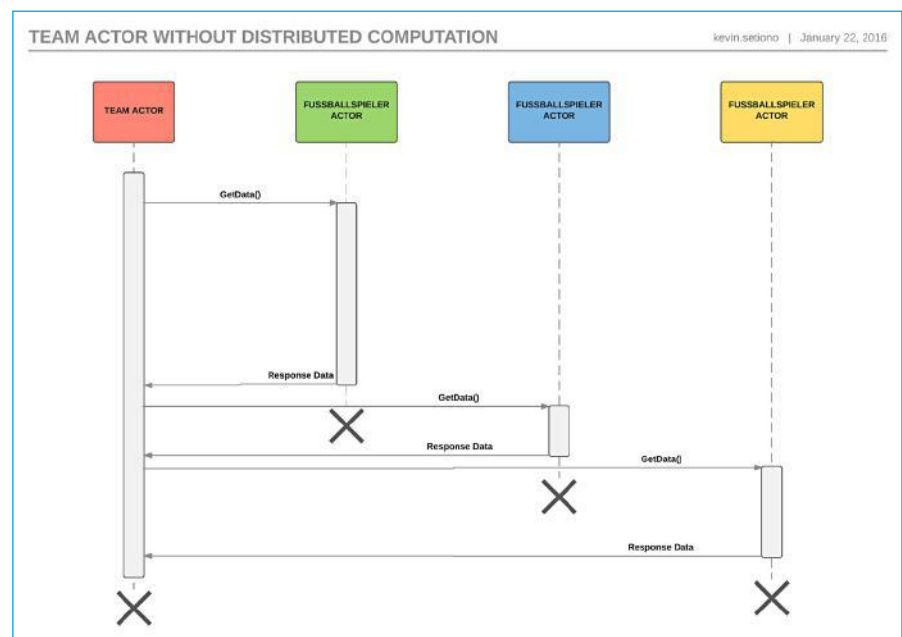


Abbildung 7: Ein Team-Actor ohne Distributed Computation.

zeit funktioniert das automatische Hinzufügen und Entfernen von Nodes im Cluster noch nicht. Wir zeigen im Folgenden unser tägliches Arbeiten mit Service Fabric.

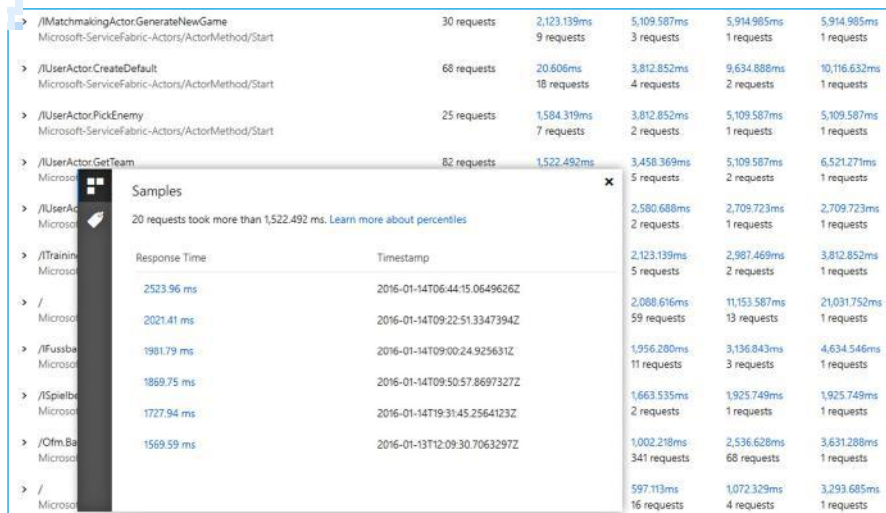


Abbildung 8: Service Profiler Summary Page.

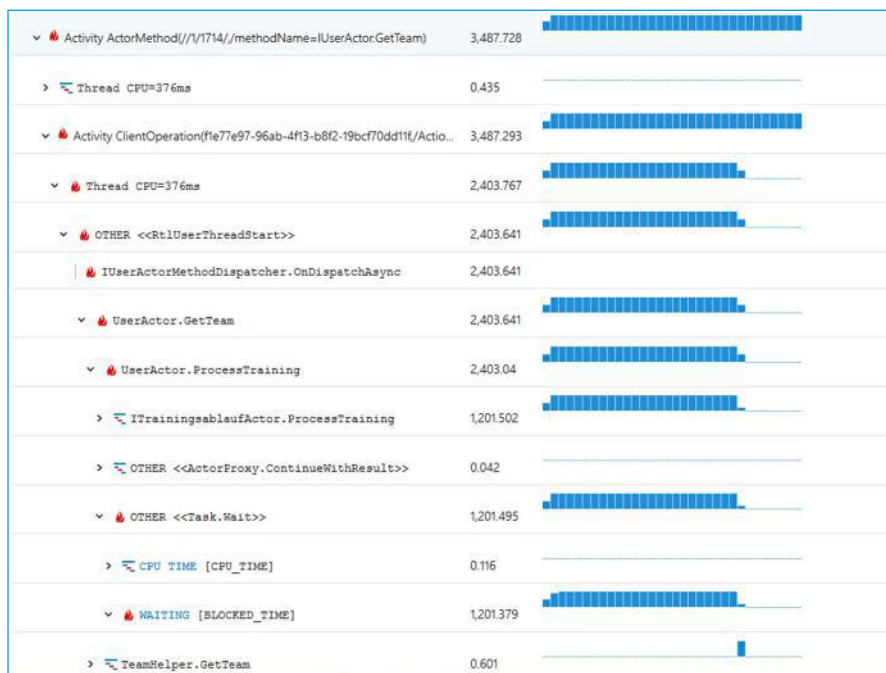


Abbildung 9: Service Profiler Detail Get Team Page.

```
<!-- Code package is your service executable. -->
<CodePackage Name="Code" Version="1.0.0">
  <EntryPoint>
    <ExeHost>
      <Program>ServiceBackend.exe</Program>
    </ExeHost>
  </EntryPoint>
</CodePackage>

<!-- Config is the contents of the Config directory under PackageRoot that contains an inde-
pendently-updateable and versioned set of custom configuration settings for your service. -->
<ConfigPackage Name="Config" Version="1.0.0" />
<DataPackage Name="Data" Version="1.0.0.0" />
```

Abbildung 10: Ausschnitt der XML-Datei Service Manifest.

Arbeiten mit Service Fabric

Service Fabric ist ein einfaches System, mit dem jeder Softwareentwickler aus der objekt-orientierten Programmierung zurechtkommt. Wir haben mit Service Fabric angefangen, als die Beispiele und die Dokumentation noch nicht ganz ausgereift waren. Wir stellen nicht-triviale Herausforderungen mit dem System vor und zeigen die eleganten Lösungen mit Service Fabric. Hinter Service Fabric versteckt sich die Komplexität eines verteilten Systems. Um das Framework komplett auszuschöpfen, muss auf eine gute Lastverteilung und Reduzierung der Wartezeiten Wert gelegt werden.

Lastverteilung und Wartezeiten

Ein verteiltes System sollte Wartezeiten reduzieren und die Last ideal verteilen. Der Softwareentwickler muss das Framework unterstützen, um die Wartezeiten zu verkürzen. Wir haben zum Beispiel in unserem Spiel für jeden Fußballspieler einen eigenen Actor. Dieser Actor verwaltet Trainingspunkte, Erfahrungspunkte und so weiter. Der Client will für die Auswahl eines Fußballspielers eine Liste aller Fußballspieler im Team anzeigen. Um die Wartezeit gering zu halten, verteilen wir die Anfragen an die Fußballspieler-Actors parallel und warten auf alle Fußballspieler-Actors, anstatt diese nacheinander abzuarbeiten. Mit anderen Worten: Wir füllen die Mailbox aller Fußballspieler-Actors in unserem Team und warten auf die Antwort von allen (siehe [Abbildung 6](#)). Das Software Pattern trägt den Namen »Distributed Computation«. In einem Sequenzdiagramm sieht man den theoretischen Gewinn. Wenn wir immer einen Actor nach dem anderen abarbeiten, dann benötigen wir mehr Zeit für dieselbe Aufgabe (siehe [Abbildung 7](#)).

Service Fabric bietet alles, was benötigt wird, um effiziente Software zu entwickeln. Aber wie so oft ist es »easy to use, but hard to master«. Um Bottlenecks in der Anwendung zu erkennen, zum Beispiel durch eine nicht ideale Lastverteilung wegen zu hoher Wartezeiten, bietet Service Fabric das nötige Tooling an – den Service Profiler.

Service Profiler

Der Service Profiler zeigt die Performance von Anfragen und Berechnungen innerhalb des Systems. Er kann zurzeit nur verwendet werden, wenn man eingeladen wurde (»invite-only preview«). Es ist schwer, im Livebetrieb Performance-Probleme nachzustellen, weil sie nur in seltenen Konstellationen auftreten. Der Overhead auf das Gesamtsystem lässt sich im Service Profiler konfigurieren. Es wird lediglich ein kleiner Agent verwendet, der bestimmte Events analysiert. In der Anwendung muss der Entwickler keine Änderungen vornehmen, um ein solides Basis-Tracking der Performance zu erhalten. Der Service Profiler listet alle Anfragen auf, die an das System gestellt wurden (siehe [Abbildung 8](#)). Es liegen nicht zu allen Anfragen detaillierte Informationen

bereit, weil der Service Profiler versucht, nur relevante Informationen darzustellen. Wenn alle Informationen immer gespeichert werden würden, dann würde der Service Profiler zu viel Last auf das System ausüben. Wir können uns detaillierte Informationen über Anfragen anzeigen lassen. In **Abbildung 9** sehen wir eine Anfrage, in der wir alle Fußballspieler von einem Team holen. Unser System führt das Training des Teams aus und gibt dann die aktualisierten Daten der Fußballspieler zurück. Die Daten im Service Profiler zeigen, dass das Training die Hälfte der Zeit einnimmt. In der Anzeige »WAITING [AWAIT_TIME]« (siehe **Abbildung 9**) verbirgt sich zum Großteil der Kommunikations-Overhead. Wenn ein Actor die Antwort von einem anderen Actor oder Service benötigt, dann wartet dieser. Hier könnten wir gegebenenfalls optimieren.

Konfiguration der Anwendung

In unserem Spiel und in den meisten anderen Anwendungen gibt es Konfigurationen. Eine gute Lastverteilung und geringe Wartezeiten sind auch bei Zugriffen auf Konfigurationen sinnvoll. Service Fabric ermöglicht Updates für Daten (Konfigurationen für die Anwendung), ohne einen neuen Code zu verteilen. Hierfür muss lediglich ein Ordner erstellt und in der Konfiguration angegeben werden (siehe **Abbildung 10**). Der Ordner muss an einer speziellen Stelle angelegt werden (ServiceBackend/PackageRoot/Data). Ein Service – kein Actor in diesem Fall – kann bei der Initialisierung auf die Konfiguration zugreifen. Wird die Konfiguration erneuert, dann erhält der Service ein Event und kann seinen internen Cache erneuern. Der Service ist stateless, weil er seinen eigenen Cache nur im Speicher hält, und kann deshalb uneingeschränkt die Last auf den Cluster verteilen. Ein Actor kann über den Service Werte aus der Konfiguration auslesen. Für genauere Informationen über das Data-Package-Handling in Service Fabric haben wir einen Quellcode-Ausschnitt in **Abbildung 11** bereitgestellt.

Matchmaking mit dem Actor Model

Das Actor Model ist nicht ideal für Aufgaben, welche die Daten von vielen Actors brauchen, zum Beispiel die Teamstärke von allen Team-Actors. Um große Listen zu vermeiden, haben wir uns für einen intelligenten Router entschieden (siehe **Abbildung 12**).

Der Matchmaking-Routing-Service hat keinen eigenen State (stateless), dadurch skaliert er uneingeschränkt. Der Router verwaltet seinen eigenen Cache mithilfe von Buckets. In jedem Bucket-Actor stehen die Teams mit der jeweiligen Stärke. Zum Beispiel sind in Teamstärke 5 alle Teams mit der Teamstärke 5 aufgelistet. Ein Bucket-Actor ist immer limitiert auf die letzten 10.000 Teams. Diese Limitierung ermöglicht dem System, schnell einen Gegner zu finden. Im Spiel verändert sich die Teamstärke mit der Zeit, sodass die Limitierung für den Spieler nicht spürbar ist.

```
/// <summary>
/// Gets the Data Package
/// </summary>
/// <returns>Data Package Version</returns>
public Task<int> GetVersion()
{
    if(!this.isReady || this.Data == null)
    {
        throw new ServiceNotReadyException(ActorConfiguration.DataServiceTypeName);
    }

    return Task.FromResult(this.Data.Version);
}

/// <summary>
/// Initializes Data Packages Changes
/// </summary>
/// <param name="cancellationToken">Cancellation Token</param>
/// <returns>Task for Service Fabric</returns>
protected override Task RunAsync(Cancellation token cancellationToken)
{
    IList<string> packageName = this.ServiceInitializationParameters.CodePackageActivationContext.GetDataPackageNames();

    foreach (string name in packageName)
    {
        System.Fabric.DataPackage dataPackage = this.ServiceInitializationParameters.CodePackageActivationContext.GetDataPackageObject(name);
        this.ProcessDataPackage(dataPackage);
    }

    this.ServiceInitializationParameters.CodePackageActivationContext.DataPackageModifiedEvent += (sender, e) =>
    {
        this.ProcessDataPackage(e.NewPackage);
    };

    return base.RunAsync(cancellationToken);
}

/// <summary>
/// Process Data Package
/// </summary>
/// <param name="dataPackage">New Data Package</param>
private void ProcessDataPackage(System.Fabric.DataPackage dataPackage)
{
    string filename = Path.Combine(dataPackage.Path, "DataPackage.json");

    if (!File.Exists(filename))
    {
        return;
    }

    this.isReady = false;
    try
    {
        using (StreamReader reader = new StreamReader(filename))
        {
            this.DataJson = reader.ReadToEnd();
            this.Data = JsonConvert.DeserializeObject<Shared.Entities.DataPackage>(this.DataJson);
        }

        this.isReady = true;
    }
    catch (Exception e)
    {
        ServiceEventSource.Current.ServiceMessage(this, e.ToString());
    }
}
```

Abbildung 11: Quellcode-Ausschnitt eines Data Service für das Data-Package-Handling in Service Fabric.

- ## Fazit

Während unserer Evaluierung haben wir uns mit einigen Anbietern und Kollegen aus der Branche unterhalten und über den aktuellen Stand der Technik gesprochen. Das Feedback, das wir aus diesen Gesprächen erhalten haben, hat uns dazu gebracht, auch über noch nicht etablierte Systeme nachzudenken.

Als Beispiel sei hier das Actor Model zu nennen, das auf viele unterschiedliche Arten implementiert und betrieben werden kann. Es wäre auch möglich gewesen, zum Beispiel Orleans bei Google oder Softlayer zu hosten. Allerdings würde da die Verantwortung für die Infrastruktur bei uns im Team liegen, was aufgrund von fehlenden DevOps-Ressourcen problematisch ist.

Wir haben einen straffen Zeitplan für unser Projekt und wollen somit so effizient wie möglich agieren und uns auf das Programmieren des Spiels konzentrieren und nicht mit der Verwaltung von Infrastruktur beschäftigen. Genau hier unterstützt uns die Service Fabric mit der Azure Cloud, denn sie offeriert eine wartungsarme Laufzeitumgebung für das Actor Model.

Auch wenn das Actor Model nicht alle Probleme lösen kann, bietet es eine sinnvolle Abstraktion eines verteilten Systems. Diese Abstraktion ermöglicht es einem Softwareentwickler, die Herausforderungen im bekannten Paradigma der Objekt-Orientierung zu lösen.

Aufgrund dieser Entscheidung haben wir alle unsere Ziele erreicht. Die Entwicklungszeit der Backend-Logik war kürzer als erwartet, wodurch wir mehr Zeit für die Qualitätssicherung hatten.

Andreas Pohl, Kevin Daniel Setiono

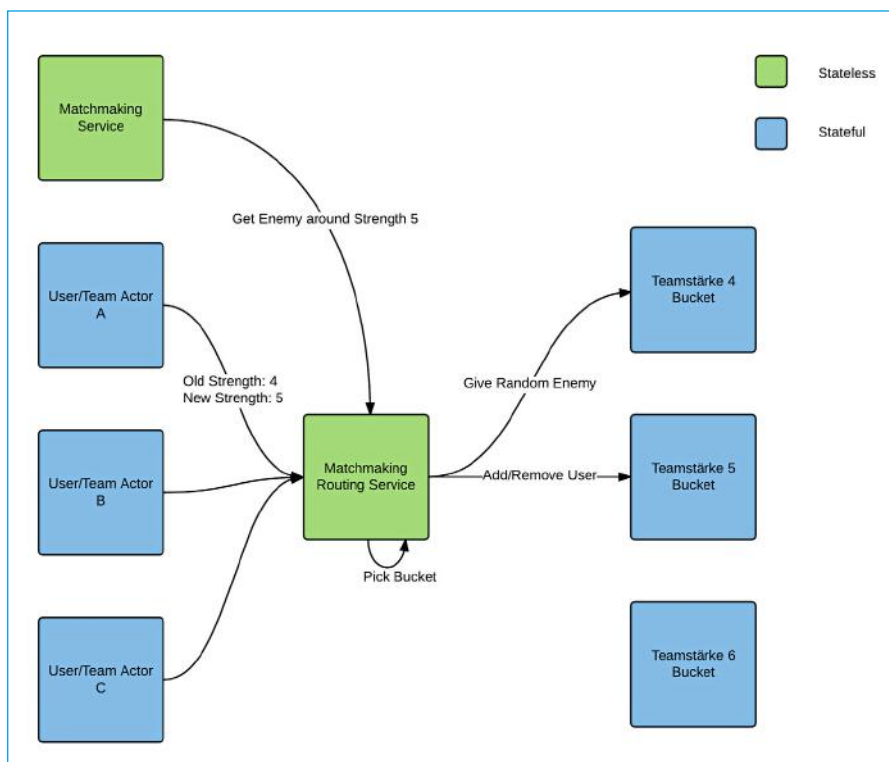


Abbildung 12: Architektur unseres Matchmaking-Systems in Football Empire nach dem Actor Model



YOUR KEY TO THE GERMAN GAMES INDUSTRY

MAKING GAMES MAGAZINE: 10,000 READERS
CENTRAL EUROPE'S MOST RELEVANT MAGAZINE FOR GAME DEVELOPERS

MAKINGGAMES.BIZ: 12,000 UNIQUE VISITORS
GERMANY'S BIGGEST WEBSITE ABOUT GAME DEVELOPMENT

FACEBOOK.COM/MAKINGGAMES: 17,400 FANS
EUROPE'S LARGEST GAME DEVELOPER COMMUNITY ON FACEBOOK

MAKING GAMES MAIL: EMAIL DATABASE
WITH MORE THAN 9,000 B2B-CONTACTS

MAKING GAMES PROFESSIONALS: LEAD DATABASE
WITH MORE THAN 700 FULLY QUALIFIED GAMES PROFESSIONALS

MAKING GAMES TALENTS: GERMANY'S MOST SUCCESSFUL
RECRUITING EVENT FOR THE GAMES INDUSTRY

KEY PLAYERS: THE WORLD'S BIGGEST GAMES INDUSTRY
COMPENDIUM WITH MORE THAN 60 COMPANY PORTRAITS

TALK TO US!

PROJECTS@MAKINGGAMES.DE

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
Bad Monkee		Grindelallee 100	20146 Hamburg		info@badmonkee.de
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	key player	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 Willua Games		Schellingstraße 35	80799 München	089 / 4167744	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	key player	Adlerstraße 74	40211 Düsseldorf	0211 / 338000	info@bluebyte.de
Blue Byte Mainz	key player	Römerpassage 1	55116 Mainz	06131 / 554470	info@related-designs.de
Bongfish		Schönhaugasse 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	key player	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 Rohrendamm 8	20539 Hamburg	0173 / 4361509	team@crazybunch.biz
createale		Mörkestraße 28/2	71636 Ludwigsburg	07141 / 2399563	contact@createale.de
CreTeam 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		Papenreye 51	22453 Hamburg	040 / 43261270	info@daedalic.de
Decane		Hallerstraße 5e	20146 Hamburg	040 / 42910947	info@decane.net
Deck13 Interactive		Gutleustraße 82	60329 Frankfurt am Main	069 / 71671660	info@deck13.com
Deep Silver Fishlabs		Großer Burstah 50-52	20457 Hamburg	040 / 8888000	business@fishlabs.net
Destrax Entertainment		Merkurstraße 10	33739 Bielefeld		info@destrax.de
Digitalmindsoft	key player	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	key player	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
Fancy Bytes		Niederstraße 5	26931 Huntorf	02151 / 3259912	biz@fancy-bytes.de

DEVELOPMENT					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Far Away Illustration & Design		Im Goldberg 42	63571 Gelnhausen		info@faraway-illustration.com
FDG Entertainment		Agnes-Pockels-Bogen 1	80992 München	089 / 18941370	info@fdgenterertainment.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
Fried Bytes		Papestraße 12	66119 Saarbrücken	0681 / 68619141	info@friedbytes.de
Funatics Software	key play	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		Rosenau 8	86551 Aichach	0700 / 38746600	info@fusionsphere.com
GameArt Studio		Bismarckstraße 107	10625 Berlin		kontakt@GameArtStudio.de
GameDuell	key play	Taubenstraße 24-25	10117 Berlin	030 / 288768210	impressum@gameduell.de
Gameforge		Albert-Nestler-Straße 8	76131 Karlsruhe	0721 / 3548080	info@gameforge.de
gameus		Kufsteinerstraße 40	83064 Raubling	08035 / 6689	info@gameus.de
Gaming Minds Studios		Georgstraße 6	33330 Gütersloh		contact@gamingmindsstudios.com
Gbanga		Limmatstraße 73	CH-8005 Zürich	+41 / 43 / 5366701	matthias@gbanga.com
GIANTS Software GmbH		Rütistrasse 18	8952 Zürich-Schliren	+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	22175 Hamburg	0160 / 7002927	alexander@ilogos.biz
Independent Arts Software		Münsterstraße 5 / HAMTEC Haus 2b	59065 Hamm	02381 / 488818	info@independent-arts-software.de
InnoGames	key play	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		Erkelenzdamm 59-61	10999 Berlin	030 / 57709936	info@k1010.de
keen games		Dreieichstraße 59	60594 Frankfurt am Main	069 / 66058838	info@keengames.com
King Art	key play	Tiefer 5	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 / 7809460	
MaDe Games GbR Neetix - Wycislik		Hochstraße 34	41334 Nettetal		info@madegames.com
McPeppergames		Nikolaus-Molitor-Straße 11a	97702 Münsterstadt	09733 / 781006	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 / 507896	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	key play	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
Phobator		Wiesentalstraße 5	90419 Nürnberg	0911 / 9389292	wendt@phobator.de
Piranha Bytes		Ruhrallee 63	45138 Essen	0201 / 806720	info.nospam@piranha-bytes.com
pixelamer.net		Clausewitzstraße 6	10629 Berlin	030 / 34347690	welcome@pixelamer.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 / 23189680	info@quadrige-games.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
Realmforge Studios		Hofer Straße 15	81737 München	089 / 55069134	info@realmforgestudios.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	key play	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		Hörvelsinger 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
Sharkbombs 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
Snapjaw Games		Dieselstrasse 7	50259 Pulheim	02238 / 4785813	info@snapjawgames.com
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@go4spirit.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 playen	Fernkornegasse 10	A-1100 Wien	+43 / 1 / 60430280	office@sproing.com
stillalive studios		Rosengasse 1	A-6020 Innsbruck	+43 / 650 / 3468942	info@stillalive-studios.com
Stratosphere Games	key playen	Gormannstraße 14	10119 Berlin	0177 / 7887699	info@stratosphere-games.com
Studio Fizbin		Hofstraße 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 playen	Hafenstraße 13	96052 Bamberg	0951 / 510908100	mail@upjers.com
VisionaryX		Friedrichstraße 12	71011 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		Friedensallee 290	22763 Hamburg	0 / 4035730010	info@xyrality.com
YAGER Development	key playen	Phuelstraße 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@kuechmeister.de
Eric Jannot (Game Design, Game Writing)		Sonnenallee 70	12045 Berlin	0178 / 5223697	
Filippo Beck Peccoz		Franz-Joseph-Straße 19	80801 München	089 / 80033204	mail@fbpsound.com
Jochen Hamma (Producing)		Wehinger Weg 9	78583 Boettingen	07429 / 910793	jochen.hammer@fantastic-realms.com
Jörg Beilschmidt (Game Design)		Berliner Ring 20	21406 Melbeck	0172 / 431243	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
Sebastian Zimmermann (3D Character Creation)	key playen			0176 / 23844655	contact@szart.info
Thomas Obermaier (Programmierung)		Kapellenstraße 1	86869 Gutenberg	0170 / 3815144	contact@polyforming.com
Wolfgang Walk (Full Scale Producing)		Karl-Leopold-Straße 6	76229 Karlsruhe	0721 / 4839963	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 playen	Rosenstraße 17	10178 Berlin	030 / 240888221	info@ad2games.com
Aeria Games Europe	key playen	Schlesische Straße 27, Aufgang C	10997 Berlin		info@aeriagames.com
Aerosoft		Lindberghring 12	33142 Büren	02955 / 760310	info@aerosoft.de
Application Systems Heidelberg		Pleikartsforsterhof 4/1	69124 Heidelberg	06221 / 300002	kontakt@application-systems.de
AppLift		Rosentraße 17	10178 Berlin	030 / 240888208	info@applift.com
astragon Software		Limitenstrasse 64-78	41236 Mönchengladbach	02166 / 146450	info@astragon.de
Bandai Namco Entertainment Germany		Carl-Benz-Straße 21	60386 Frankfurt am Main	069 / 24449100	de.info@bandainamcoent.eu
BHV		Novesiastraße 60	41564 Kaarst	02131 / 76501	info@bhv.de
Bigpoint		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 playen	Schwedter Straße 263	10119 Berlin		info@crobo.com
Crimson Cow		Tarpen 40, Haus 5	22419 Hamburg	040 / 59355237	info@crimsoncow.de
Crytek	key playen	Grüneburgweg 16-18	60322 Frankfurt am Main	069 / 21977660	cryengine@crytek.com
Daedalic Entertainment	key playen	Papenreye 53-Workport Unit 2	22453 Hamburg	040 / 43261270	info@daedalic.de
Deck13 Interactive		Guteleustraße 82	60329 Frankfurt am Main	069 / 71671660	info@deck13.com
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	key playen	Wiener Platz 7	81667 München		contact@gamesgroup.eu
EuroVideo Medien		Bavariafilmpfad 7 / Gebäude 33	82031 Grünwald	089 / 96244442	info@eurovideo.de
flashpoint	key playen	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 playen	Albert-Nestler-Straße 8	76131 Karlsruhe	0721 / 3548080	info@gameforge.de
GameGenetics	key playen	Alte Jakobsstraße 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	1060 Wien	+43 / 1494 / 5056	3rdparty@gametwist.com
GamesInFlames		Karlstraße 68	80335 München	089 / 210205715	info@gamesinflammes.com
gamigo	key playen	Behringstraße 16b	22765 Hamburg	040 / 4118850	info@gamigo.com

PUBLISHING/DISTRIBUTION					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Halycon Media		An der Autobahn 24	23858 Reinfeld	04533 / 61090	info@mail@halycon.de
Headup Games		Nordstraße 102	52353 Düren	02421 / 4868700	info@headupgames.com
HitFox Group		Rosenstraße 17	10778 Berlin	030 / 240888200	info@hitfoxgroup.com
Infernum Productions		Uhländstraße 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 play	Wilhelm-Leuschner-Straße 11-13	67547 Worms	06241 / 5061900	info@kalypsomedia.com
Koch Media		Lochamer 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		Kummelenstrasse 20	CH-4104 Oberwil/BL	+41 / 61 / 7220712	werner.sala@millform.ch
mixtvision Digital	key play	Leopoldstraße 25	80802 München	089 / 383770925	kontakt@mixtvision.de
Morphicon Limited	key play	Lindwurmstraße 88	80337 München	089 / 45235460	info@morphicon.de
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 play	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		Kaferstraße 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		Limitenstraße 64-78	41236 Mönchengladbach	02166 / 618660	info@rondomedia.de
RTL Games		Achenerstraße 1036	50858 Köln	0221 / 4560	entwickler@rtl-interactive.de
Rumble Media		Kriegsstraße 5	76137 Karlsruhe	0721 / 2018020	info@fettspielen.de
S.A.D.		Rötelsbachstraße 91	89079 Ulm	07305 / 96290	kontakt@s-a-d.de
Sega		Rablistraße 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	key play	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@t24.de
Tivola Publishing		Oeverseestraße 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@traffcaptain.com
Travian Games		Wilhelm-Wagenfeld-Straße 22	80807 München	089 / 3249150	mail@traviangames.com
Ubisoft	key play	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 play	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 play	Charlottenstraße 62	10179 Berlin	030 / 24087790	info@biu-online.de
BIU.Dev/BIU.Net	key play	Charlottenstraße 62	10177 Berlin	030 / 24087790	info@biu-online.de
FilmFernsehFonds Bayern	key play	Sonnenstraße 21	80331 München	089 / 5446020	filmfoerderung@fff-bayern.de
Film- und Medienstiftung NRW	key play	Kaistrasse 14	40221 Düsseldorf	0211 / 9305000	info@filmstiftung.de
G.A.M.E.	key play	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 play	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 berlin/brandenburg	key play	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 / 7744881	info@hessen-it.de
mediaboard berlin-brandenburg	key play	August-Bebel-Straße 26-53	14482 Potsdam-Babelsberg	0331 / 743870	info@mediaboard.de
MFG Medien- und Filmgesellschaft Baden-Württemberg		Breitscheidstraße 4	70174 Stuttgart	0711 / 90715-300	info@mfg.de
Mitteldeutsche Medienförderung	key play	Hainstraße 17-19	4109 Leipzig	0341 / 269870	info@mdm-online.de
nordmedia	key play	Expo-Plaza 1	30539 Hannover	0511 / 1234560	info@nordmedia.de
Northstar Developers		Bödekerstr 84	30161 Hannover	0160 / 97880272	d.brueinig@collision-studios.com
OVUS		Aichhörnstraße 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 play	Torstraße 6	10119 Berlin	030 / 24088660	kontakt@usk.de
WERK1 München	key play	Grafinger Straße 6	81671 München	089 / 9952990	info@werk1muenchen.de
EDUCATION					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
Acagamics e.V.		Universitätsplatz 2, 629	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 play	Rungestrasse 20	10179 Berlin	030 / 29779120	info@games-academy.de
Games Academy Frankfurt	key play	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

EDUCATION					
Firma		Straße/Hausnummer	Postleitzahl/Ort	Telefon	E-Mail
HTW Berlin		Wilhelmshofstraße 75a	12459 Berlin	030 / 50190	
Macromedia Akademie		Eisenheimerstraße 63	80687 München	089 / 96160800	ausbildung@macromedia.de
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 player	Lindenstraße 20-25	10969 Berlin	030 / 3992660	info-ber@mediadesign-fh.de
Mediadesign Hochschule	key player	Werdener Straße 4	40227 Düsseldorf	0211 / 1793930	info-dus@mediadesign-fh.de
Mediadesign Hochschule	key player	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 player	Soltauer Straße 18-22	13509 Berlin	030 / 43094470	berlin@sae.edu
SAE Institute Bochum	key player	Metzstraße 23	44793 Bochum	0234 / 93451310	bochum@sae.edu
SAE Institute Frankfurt	key player	Homburger Landstraße 182	60435 Frankfurt am Main	069 / 543262	frankfurt@sae.edu
SAE Institute Hamburg	key player	Feldstraße 66	20359 Hamburg	040 / 23688080	hamburg@sae.edu
SAE Institute Köln	key player	Medienzentr. Ost, Carlswerkstr. 11c	51063 Köln	0221 / 9541220	koeln@sae.edu
SAE Institute Leipzig	key player	Dittrichring 10	40109 Leipzig	0341 / 3085160	leipzig@sae.edu
SAE Institute München	key player	Bayerwaldstraße 43	81737 München	089 / 89068771	muenchen.sae.edu
SAE Institute Stuttgart	key player	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	key player	Ludwig-Guttman-Straße 6	69123 Heidelberg	06221 / 882790	Daniel.Goerlich@Hochschule-Heidelberg.de
Zürcher Universität der Künste	key player	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@gametaskforce.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 / 659998000	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 / 41719113	info@freaks4u.de
Friedmann Kommunikation		Schanzenstraße 36	51063 Köln	0221 / 5348710	info@fr-k.de
Gärtner PR		Destouchesstraße 68	80796 München	089 / 30766854	info@gartner-pr.de
Game Developers Conference Europe	key player	Kaiser-Wilhelm-Straße 30	12247 Berlin	030 / 34622644	carsten.kohlenbeck@ubm.com
gamescom	key player	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
Globaloc	key player	Storkower Straße 158	10407 Berlin	030 / 428075780	info@globaloc.de
GRAEF Rechtsanwälte		Jungfrauenal 8	20149 Hamburg	040 / 80600090	www.graef.eu
HoneyTracks (Game Analytics)		Schelling Straße 35	80799 München	089 / 28723995	info@honeytracks.com
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
LOVOO	key player	Prager Straße 10	01069 Dresden	0351 / 41889939	boost@lovoo.net
Marchsreiter Communications		Guldeinstraße 41a	80339 München	089 / 5191942	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 / 5659777	info@motionarea.de
Osborne Clarke (Anwaltskanzlei)		Innere Kanalstraße 15	50823 Köln	0221 / 51084000	info@osborneclarke.de
Osborne Clarke (Anwaltskanzlei)		Leipziger Straße 28	40822 Mettmann	02104 / 172660	info@partnertrans.com
paysafecard.com Wertmarken		Am Europplatz 2	1120 Wien	+43 / 1 / 72083800	sales@paysafecard.com
Pixelworkshop / IG Computergrafik		Hirschengasse 1/26	1060 Wien	+43 / 1 / 588018698	info@pixelworkshop.at
Quinke Networks (PR, Marketing)		Bei den Mühren 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)		Henriettenweg 4	20259 Hamburg	040 / 43270432	info@rode-mathe.de
ROESSLER PR (Marketing, PR, Consulting)		Walter-Leiske-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	67146 Deidesheim	06326 / 6010	info@schanzgames.com
Schulte Riesenkampf (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		Gerresheimer Straße 9	40721 Hilden	02103 / 9411914	info@two-pi-team.de
Visibility Communications (PR, Text)		Wichertstraße 16/17	10439 Berlin	030 / 58859941	hi@visi.bi
wazal		Sonnenallee 70	12045 Berlin	0178 / 5223697	jannot@wzaservices.de
Webedia Gaming	key player	Ridlerstraße 55	80339 München	089 / 244136600	david.bhulaputra@webedia-group.com

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
Prof. Dr. Malte Behrmann	Ganghoferstraße 4	12163 Berlin	mb@malte-behrmann.de	030 / 65214472
Marc Huppke (Projektmanagement)	Frankfurter Straße 5	64521 Gross-Gerau	06152 / 9614432	marc@make-projects.com
Marc Oberhäuser (Consulting)	Berzbuier Strasse 98	52355 Düren	02421 / 51547	marc.oberhauser@freelancer-games.com
Stefan Köhler (Lektor / Autor)	Am Salgenteich 34	38259 Salzgitter	0176 / 22512273	stefanclemenskoeherl@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. OG	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
DICO Deutschland	Freiburger Straße 5	51107 Köln	0221 / 20430504	info@dico4u.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
fattoogoo	Mariahilferstraße 50	A-1070 Wien	+43 / 1 / 23622970	office@fattoogoo.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 / 808090	info@keuthen.net
Konsole Labs	Gritznerstraße 42	12163 Berlin	030 / 23634838	info@konsole-labs.com
Lingoon (Lokalisations-Middleware)	Hildastraße 38	68723 Plankstadt	06202 / 4095030	info@lingoon.com
Living Liquid Software	Langer Weg 15	A-6020 Innsbruck	+43 / 0 / 512263535	info@livingliquid.com
metricminds (Grafik)	Russelsheimer 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
PIXABLE STUDIOS	Buchenstraße 16b	01097 Dresden	0351 / 56341360	info@pixable.de
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)	Gneisenastraße 10/11	97074 Würzburg	0931 / 8049000	info@virgin-lands.com
Z-Ground Illustration (Grafik)	Oeltzenstraße 17	30169 Hannover	0511 / 1696959	jz@z-ground.com
zuukal (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@knightsoundtrack.com
Johan Weigel (Musik / Ton / Sound)	Christburger Straße 28	10405 Berlin	0176 / 41446494	johan@sonic-gallery.com
Max Schulz				info@xocol.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
Sound42 / Lukas Hasitschka	Kulmgasse 38/8	1170 Wien	+43 / 6641059157	lukas@sound42.com

Ein Tag bei ... WARGAMING CHICAGO-BALTIMORE



Jana Verden vom QA Testing Department schaut ganz genau hin, wenn es um die letzten Bugs vor dem Release geht ...



Die Herrin der Zahlen – Business-Intelligence-Analystin Erica Absetz verliert selbst bei noch so komplizierten Zahlenkolonnen ihre gute Laune nicht.



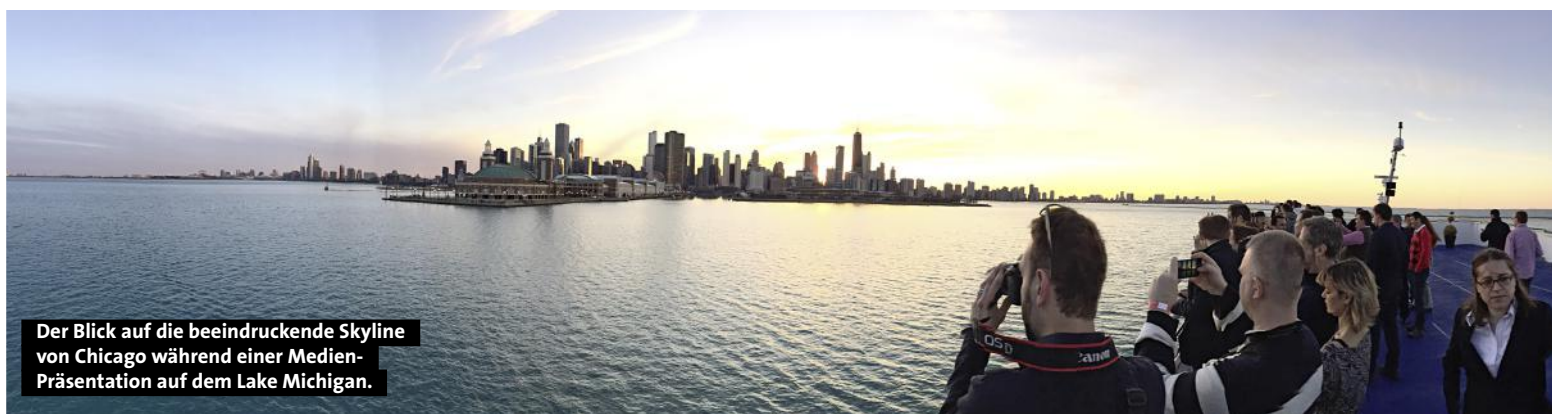
Als Senior Sound Designer verantwortet Brendan Blewett nicht nur die Musik in »World of Tanks«, sondern auch die krachenden Panzergeräusche von Ketten und Schüssen.



Lead Game Designer Jeff Gregg (in der Mitte mit Basecap) beim täglichen Jour Fix mit dem Team in lockerer Atmosphäre.



Associate Producer Jonathan Harper freut sich über den reibungslosen Start von »World of Tanks« auf der PlayStation 4 am 19. Januar 2016.



Der Blick auf die beeindruckende Skyline von Chicago während einer Medien-Präsentation auf dem Lake Michigan.



Auch europäische Gäste sind gern gesehen beim Entwicklerteam in Chicago – hier der Senior Communication Coordinator EU East Ingo Horn und der PR Manager Poland Konrad Rawinski im Rahmen eines Pressevents.



Das Tor, das Gäste von Wargaming Chicago-Baltimore in die Welt der Konsolenversionen von »World of Tanks« führt, ist ein »unscheinbarer« Fahrstuhl.



Associate Producer Jonathan Harper prüft in einer frühen Version die DualShock-Controller-Unterstützung.

Das Wargaming-Studio in Chicago ist hauptsächlich dafür verantwortlich, die erfolgreichen Free2Play-Titel des weißrussischen Entwicklers auf den Konsolen an den Start zu bringen – so wie zuletzt im Januar »World of Tanks« für die PlayStation 4. Hier gewährt das Team einen Blick in die tägliche Arbeit.

Hat dank unkomplizierten Abläufen bei den Greenlight-Meetings gut lachen: Live Publishing Producer Darold Higa.



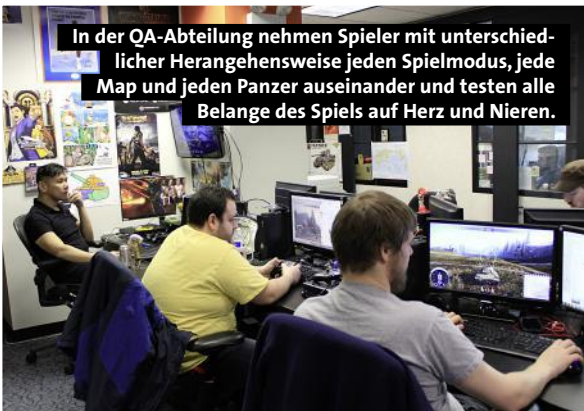
Das jährliche »Chilli Cook Off« fordert jedes Jahr seine Opfer, die sich beim Schärfegrad etwas überschätzen ...



Die mehr als 50.000 Polygone pro Panzer benötigen entsprechende Texturierungen, wozu das Grafikdepartment vollen Einsatz zeigt.



In der QA-Abteilung nehmen Spieler mit unterschiedlicher Herangehensweise jeden Spielmodus, jede Map und jeden Panzer auseinander und testen alle Belange des Spiels auf Herz und Nieren.



QA Shift Lead Rachel Walaski jongliert mit den Einsatzzeiten in der Quality Assurance, um alle Teststationen besetzt zu halten.



Producer Matt Monforte (im blauen T-Shirt) und UX QA Lead Jared Pace bei teilweise hitzigen Diskussionen um das User Interface – deshalb kühlt ein Ventilator (auch) die Gemüter.



Dank regelmäßiger Meetings der Grafiker kann gewährleistet werden, dass auch die PS4-Version von »World of Tanks« bald über die volle Anzahl von über 350 Panzern verfügt.



In dem umfangreichen Fuhrpark des Spiels steckt jede Menge Arbeit, damit die Modelle detailliert und realistisch auf dem Bildschirm wirken.



Teamplay wird großgeschrieben – hier einige der weniger kamerascheuen Kreativen, die hinter der frisch veröffentlichten PlayStation-4-Version von »World of Tanks« stehen.



MAKING GAMES VORSCHAU

03/2016
MAKING GAMES
ERSCHEINT AM 22. APRIL 2016

Culture Shock

Steam, verschiedene App Stores und PlayStation Network sowie Xbox Live machen es Entwicklern besonders einfach, ihre Spiele zu veröffentlichen. Noch dazu eröffnen sie Märkte wie China, Russland oder Brasilien, die früher nur mit viel Aufwand oder Publishing- und Distributionspartnern überhaupt bedienbar waren. Diese neue Freiheit bringt aber auch Probleme mit sich. Daher geben unsere Experten Tipps, welche Länder welche Genres und Plattformen am liebsten mögen. Sie erklären, wie Spiele für einzelne Territorien angepasst werden müssen oder welche gesetzlichen Einschränkungen es geben mag und wie man den richtigen Partner in den neuen Märkten findet.

Weitere Themen

Traversal Level Design

Unser Experte veranschaulicht, wie Levels und Rätsel in Titeln wie »Tomb Raider« und »Assassin's Creed« entworfen werden.

Post Mortem zu Bastion für PS Vita

Das Team von BlitWorks blickt auf die Portierungsarbeiten zurück und erklärt, wie es das C#-basierte Spiel mit viel Aufwand auf C++ umstellte.

Musik und Sound-Effekte in Horrorspielen

Unser Experte erklärt, warum Geräusche und Hintergrundmusik enorm wichtig für die Atmosphäre von Spielen sind und wie man gute produziert.

IMPRESSUM

Webedia Gaming GmbH
Redaktion Making Games
Ridlerstraße 55
80339 München
Tel.: 089 / 244 136-0, Fax: 089 / 244 136-118
Geschäftsführer: Nicolas John
Chefredakteur: Heiko Klinge (v. i. S. d. § 55 Abs. 2 RStV) - Anschrift des Verlags
Veröffentlichung gemäß § 8, Absatz 3 des Gesetzes über die Presse vom 08.10.1949: Die GameStar GmbH ist eine 100%ige Tochtergesellschaft der Webedia Gaming GmbH
Ridlerstraße 55, 80339 München
Tel.: +49 (89) 244 136-0, Fax: -118
Geschäftsführer: Nicolas John
Registergericht: Amtsgericht München
Handelsregisternummer: HR B 218 859
USt-ID Nr.: DE 300 33 09 61
Gesellschafter der Webedia Gaming GmbH sind die Mehrheitsgesellschafter Webedia GmbH (Amtsgericht Charlottenburg, HR B 114531 B), sowie PietS-miet Holding GmbH (Amtsgericht Köln, HR B 84294) und die Connected 4 Media GmbH (Amtsgericht München, HR B 218234)

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

REDAKTION

Chefredakteur	Layout & Design
Heiko Klinge	Sigrun Rüb (ltd.)
	Manfred Aumaier
Managing Editor	Alexander Wagner
Sebastian Weber	Eva Zechmeister
	Anita Blockinger
Redaktion	(Freelancer)
Yassin Chakhchoukh	
Patricia Geiger	Lektorat (Freelancer)
Markus Schwerdtel	Marion Schneider

Freie Mitarbeiter in dieser Ausgabe (Redaktion)

Daniel Dumont, Christoph Ender, Mark Estdale, Balázs Farkas,
Alexander Karl, Jan Klose, Yowan Langlais, Carlos Llevano,
Andreas Pohl, Christian Schlütter, Kevin Daniel Setiono,
Mischa Strecker, Jan Theysen, Arne von Öhsen, Timm Walter,
Dr. Matthew White

Wir danken dem Interview-Partner:

Stefan Marcinek

Assistenz: Isa Stamp, Anita Thiel, Annie Weissenberger

KONTAKT

Anzeigenanfragen:
Ralf Sattelberger 089 / 244 136-730
ralf.sattelberger@webedia-group.com
Director Sales:
Ralf Sattelberger 089 / 244 136-730
Artikel- & Themenvorschläge:
editor@makinggames.biz
Recruiting- & Projektanfragen:
projects@makinggames.biz
Adressänderungen Abo:
kundenservice@makinggames.de
Änderungen für Firmenregister:
info@makinggames.biz
Betreff: Firmenregister
Digitale Anzeigenannahme Print:
Manfred Aumaier, 089 / 244 136-668
anzeigendispoprint@gamestar.de
Digitale Anzeigenannahme Online:
ads@webedia-group.de
Anzeigenpreise:
www.makinggames.biz/mediadaten
Druck und Beilagen:
SDV Direct World GmbH
01159 Dresden

HAFTUNG & CO.

© Copyright Webedia Gaming 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:
Webedia Gaming GmbH
Redaktion Making Games
Ridlerstraße 55
80339 München
Tel.: 089 / 244 136-0, Fax: 089 / 244 136-118

Want to be a KEYPLAYER?

8 gute Gründe für einen Eintrag in Key Players 2016

- 1 12.500 Exemplare – das weltweit auflagenstärkste Verzeichnis der Games-Industrie
- 2 das einzige Branchenverzeichnis, das offizieller Medienpartner der GDC und der GDC Europe ist und dort kostenlos an Branchenentscheider verteilt wird
- 3 kostenloser Versand an alle Making Games Abonnenten, dadurch optimale Branchenabdeckung im deutschsprachigen Raum
- 4 weltweiter Vertrieb als kostenlose iPad-App und als ePaper
- 5 beste Aufmerksamkeit für Ihr Firmenprofil dank höchster Produktqualität und durchdachter Leserführung
- 6 kompetente Beratung bei der Eintragserstellung durch das erfahrene Redaktionsteam von Making Games
- 7 nachweislich hohe Relevanz für Presse, Politik und Geschäftspartner: über 60 Unternehmen aus allen Bereichen im ausführlichen Portrait
- 8 fair und transparent: egal ob Start-up oder Weltmarktführer – jeder zahlt das Gleiche, jeder bekommt das Gleiche



Sichern Sie sich Ihren Eintrag noch bis zum 07. März 2016 zum Early-Bird-Tarif mit 20 Prozent Rabatt!

- Unternehmensprofil (2 Seiten).....990 Euro
- Freelancer-Profil (0,25 Seiten).....195 Euro

iPad App, ePaper & Buchungsformular → keyplayers.de

**Beratung
und weitere
Informationen**



Kundenbetreuung
Susanne Schreiner
Sales

Telefon: +49 (0)89/244 136-733
E-Mail: susanne.schreiner@webedia-group.com



Projektleitung
Sebastian Weber

Telefon: +49 (0)89/244 136-650
E-Mail: sebastian.weber@webedia-group.com



SPIELEND KARRIERE MACHEN!

UNSER STANDORT: KARLSRUHE
UNSER SPIELFELD: DIE WELT
WIR MACHEN ERSTKLASSIGE ONLINE-SPIELE.

JOB-LEVEL-UP GEFÄLLIG?
DANN KOMM ZU GAMEFORGE!

<http://corporate.gameforge.com/jobs/>

Adresse:

Recruiting Office
Albert-Nestler-Str. 8
76131 Karlsruhe
Tel: +49 721 354 808 244

