

making games

DESIGN | BUSINESS | ART | TECHNOLOGY

Mainstream

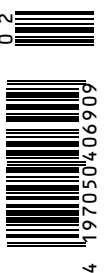
World of Tanks
Shovel Knight
This War of Mine
The Binding of Isaac

F*#K MAINSTREAM.

GAME DESIGN ANALYSIS
THE MAKER OF AMNESIA DISSECTS
THE HORROR IN ALIEN: ISOLATION

BLIZZARD-INTERVIEW
HOW THE EXPERIMENT HEARTH-
STONE BECAME A GLOBAL SUCCESS

KI-PROGRAMMIERUNG
HOW A STRATEGY GAME FROM 1990
MADE SHADOWRUN MORE INTELLIGENT



GDC RETURNS MARCH 2-6, 2015

SAVE UP TO \$300 ON ONSITE PRICING BEFORE FEBRUARY 25, 2015

**making
games**

READERS, USE CODE

GDC15MG

TO SAVE EVEN MORE ON
ALL ACCESS & MAIN
CONFERENCES PASSES!

500+ SESSIONS

DIVERSE EXPO FLOOR

BUSINESS & CAREER
DEVELOPMENT

PARTIES, EVENTS &
SPECIAL EXHIBITS

INDEPENDENT
GAMES FESTIVAL

GAME DEVELOPERS
CHOICE AWARDS

GDC

GAME DEVELOPERS CONFERENCE®

MOSCONE CENTER · SAN FRANCISCO, CA

MARCH 2-6, 2015 · EXPO: MARCH 4-6, 2015



REGISTER AT GDCONF.COM



10 YEARS, BUT NOT A SECOND OF MAINSTREAM

As you're holding this issue in your hands, it's nearly time again for this year's Game Developers Conference in San Francisco, also marking the 10th anniversary of Making Games. The concept for our magazine which saw the light of day under the title /Gamestar/dev was the result of a cigarette break at said conference in the year 2005. Ten years later, the magazine which was planned as a voice and a source of information for the German gaming scene has turned into the biggest professional developers' magazine in Central Europe whose ever increasing Facebook community is now to a large extent home to English-speaking industry pundits. But – as you know – Making Games is no longer just the magazine you're holding in your hands; it also includes our internationally oriented webpage makinggames.biz, our job exchange and the company register Key Players which, in cooperation with the GDC, makes German industry representatives known all over the world.

Abandoning familiar paths

In short: We could have easily added a post mortem of our own magazine as part of our cover story »F*#k Mainstream!« where the developers of »Shovel Knight« (p. 12) and »This War of Mine« (p.16) explain why they stuck to their unusual and disturbing game idea, or where the boss of Wargaming demonstrates how rocky the road to becoming a big player was (p. 22), and where an analysis shows why Edmund McMillen's games like »The Binding of Isaac« (p. 26) became so successful. Because

when Making Games was founded, articles about computer or videogames mainly focused on previews and reviews. A magazine that was aimed at developers was therefore not only an economic risk; nobody could tell whether the concept would be a lasting one and whether enough authors would be prepared to supply interesting and exciting content. We still exist to this day and that's not least owed to you, our readers, and of course the many industry pundits who, over the years, have contributed articles and shared their knowledge with others. Many thanks for that: It was thanks to your help alone that we were able to leave the editorial mainstream.

Enjoy the latest issue!

Your Making Games Team

Sebastian Weber
is Managing Editor of
Making Games Magazin.

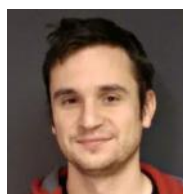


»It was thanks
to your help alone that we
were able to leave
the editorial mainstream.«

Bye Nicole, hello Dustin!

This issue marks the end of an era since it is the last issue with the participation of our longtime freelance employee Nicole Klinge. As Project Consultant since 2007 she was not only the central contact for clients and ad inquiries, but also a driving force behind the success stories of Making Games Talents, Making Games Job Exchange and Key Players. And last but not least she was pretty much the heart and soul of the team. We owe a lot to her and will miss her deeply!

At the same time we're looking forward to our new colleague Dustin Reichl who will from now on take over Nicole's area of responsibility and provide guidance for all non-editorial inquiries.





INDIE SUCCESS WITH 8 BIT GRAPHICS AND NINTENDO

The team of Yacht Club Games all quit their safe jobs to make their dream come true: They funded their very classic game Shovel Knight via Kickstarter and got more successful than expected.



David D'Angelo
is programmer
at Yacht Club Games.

Yacht Club Games is an independent studio in California, which just released its first game - »Shovel Knight«! David's main role at Yacht Club Games is programming engine and gameplay, but since there's just half a dozen of employees, they all take on a piece of everything from business and marketing to game design and implementation. Some past games David developed when working for WayForward include amongst others »Double Dragon Neon«, »BloodRayne Betrayal«, »Mighty Milky Way«, »Thor: God of Thunder«, »Batman: The Brave and the Bold« and »DuckTales: Remastered«.



All the following numbers and charts take only the first month after the release of »Shovel Knight« into account and it was pretty nuts to see our little fledgling Kickstarter all grown up and in the wild during this

time. Believe it or not, besides the desire to make a great game, one of the reasons we wanted to leave our safe and secure jobs behind and enter the indie world was to make the development process more accessible to the public - so that you all would have a better idea of how games are made and how to measure their success. That's our task today!

Unfortunately, most developers aren't permitted or aren't willing to just give away sales expectations or results! Fortunately, we're not like most developers. Hopefully we can add more pieces to the tiny pile of existing information to help out other developers, journalists, enthusiasts, and Grandma Swamp-a-likes for the future. You all are the reason this game even exists, and we feel it's our responsibility to let you know about every detail that went into its construction.

Alright, here we go! We're going to be stepping through the whole process from dream to Kickstarter to release, sprinkling in our sales numbers and some statistics throughout about how our game has done from June 26th - July 26th, 2014. Keep in mind, none of the numbers we'll share are precise as pinning down the numbers can be tricky for various reasons. We've given it our best shot!

It all began with a Kickstart!

To know what our current sales numbers mean, you have to know where we started! On April 13th, 2013, we finished our Kickstarter campaign, funded by our lovely, amazing 14,749 backers for a grand total of 311,502 dollars. We also ran a concurrent PayPal donation period, where 935 backers reaching a total of 17,180 dollars also joined in on the fun. When the campaign concluded, we asked all the backers to tell us what platform they wanted the game for ... and here are the results!

Surprisingly, Nintendo backers only provided around a third of the total support. We would have expected this number to be much higher given how focused (at least inwardly) our campaign was around Nintendo platforms. Were the naysayers right; do Nintendo indie games not sell well!? We'll come back to this point later when we discuss the full sales, but it's probably safe to say that early on, Kickstarter was still a PC dominated platform.

So in total, we went into full »Shovel Knight« production with 328,682 dollars in hand for the rest of development (keep in mind that 328,682 Dollars does not account for taxes, Kickstarter's takeaway, fees, backers that dropped payment, etc). That certainly is a large sum by itself, but not quite so when it comes to making a game!

It may be surprising, but even a small downloadable game can often cost upwards of a million dollars to make! We planned a game that was going to take five people more than a full work week (six if you count Jake) for two

years (this schedule includes a year for the game, with an additional year for developing stretch goal content)! That kind of game simply is not possible on 328,000 dollars, and let's break down why!

Cost of development

What does a game usually cost? How do you even figure that out? Well, it's tough to be honest, but a very standard metric is to determine how many employees will be on the team, and multiply their cost by the amount of time the game is in development. This isn't super accurate because in the game development world, we criminally underestimate each component's time and effort. Typically we compensate for this by overestimating the time, salary or quantity elements out a bit.

So let's do that for »Shovel Knight«! These days, most studios will put the average cost of a developer on a game at around a 10,000 dollars per man month. What does that mean? Essentially, each developer will cost the company 10,000 dollars a month or 120,000 dollars a year. Now, of course, not every developer on the team makes that much money and often, none of the developers on the team make that much money.

That's because this monthly 10,000 dollars go into much more than just paying the salary of the employee. It covers any and all expenses accumulated from having that employee on site. That includes not only individual expenses for the employee like salary, health insurance, etc. but also company expenses like rent, electricity, water, food and snacks, conventions, computer and other equipment, software licenses, lawyer fees, taxes, development kit expenses ... the list goes on and on. Given that it encompasses so much, we can use this figure to calculate the cost for the entire game. Also note that some developers may only be on a project for a few months, while others will work for the whole project's duration. The man month total number can vary widely for games, but for Shovel Knight, we figured it was something around 144 man months to finish the game.

Now, you too have the power to estimate game budgets! Just multiply it out! For example, a 50 person team working for two years? That would be 12,000,000 dollars! Twelve million dollars! You can see how game budgets really start to add up!

If you're running the numbers in your head, you might be thinking – you told us you were going to make and release »Shovel Knight« and its stretch goals within two years! There are six people behind the core team that made Shovel Knight. According to the standard rules we just established, to make »Shovel Knight« in the two years we thought it would take, we would need 1,440,000 dollars on hand! Well that just wasn't going to happen!

So what did we do to offset that cost? First, we took Jake Kaufman's salary out of the equation – he was generous enough to agree to a

post-release payment and diligent enough to provide most of the sound and music in a very short time frame. With that cost mitigating measure, we're now down to 1,200,000 dollars ... that didn't help much!

Next we decided that, yes, this was a two year game to make including the stretch goals, but we could probably squeeze it into a year if we took those out and released them as free updates in the future using the sales of the game to fund the cost. In case you were wondering, that's why we announced during the Kickstarter that the stretch goals would be post release content. We were planning that far in advance! Ok, so we get to cut the budget in half by shaving off a year: we're down to 600,000 dollars now. We're getting close.

Unfortunately, we've now run out of options – the only thing left to do is cut the amount allocated for each person by half. So now we're running on 5,000 dollars man months – which is extremely low. At first that sounds like 60,000 dollars per year, but with the company costs, taxes, etc. in reality it's more like 30,000 dollars for each of us over the course of the year ... and that's before any taxes.

At this point you have to be thinking, 30,000 dollars minus taxes for one year's salary, a grueling work pace (think: 12 to 18 hours a day, 7 days a week) and no stability whatsoever ... why bother? An NES game, that's not going to sell. The 80's are over, man, get with the times! The cool kids are all into the AAA explosive action sequences. You'll never make it!

Why bother?

Not a bad question honestly. Let's add to the equation the fact that we knew that signing up for this project was much more than just developing »Shovel Knight«. Because we were solely in charge of handling the business, marketing, publishing, merchandise, support, and so on for the game, we were committing all our waking hours to the game. We knew we would finish »Shovel Knight« and barely keep our souls and lives intact. We would be sleeping and eating »Shovel Knight« and not remember what the sun looked like from our windowless office space. It would be grueling terrible. And there was no safety net.

About Shovel Knight

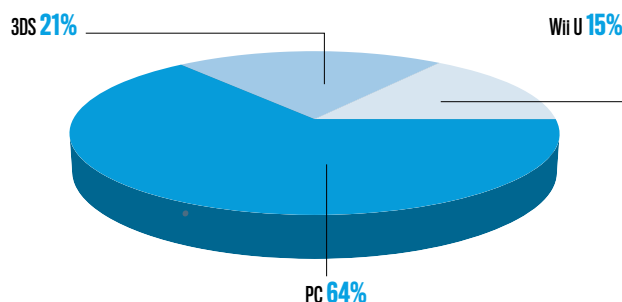


»Shovel Knight« is a mix of classic action adventure and platforming gameplay based on 8-bit retro graphics. The player takes the role as the eponymous Shovel Knight, a small knight with a huge quest: He has come to a valley with two goals, to defeat the evil Enchantress and save his lost beloved. This ends up in well balanced but hard platforming riddles and combat with crazy boss fights and lots of humor in a somewhat open world where the quest unfolds in a unique order.

The game was funded via Kickstarter where its campaign ended on April 14th, 2013. The game was finally released June 26th, 2014 for Steam as well as Nintendo 3DS as well as Wii U, a version for OSX was released September 12th, 2014.

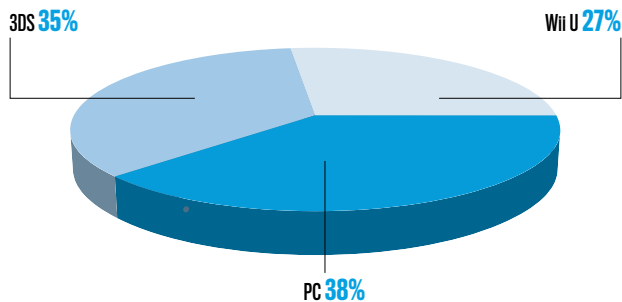


Kickstarter Reward Breakdown for 15,685 Sales



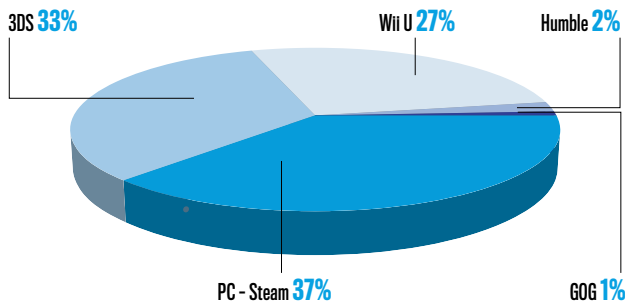
The Kickstarter campaign showed that users of Nintendo's consoles are not into preordering or backing games but prefer buying game day one of release (as was proven later on).

First week: ~75k Copies Sold



Within the first week about 75,000 copies of »Shovel Knight« were sold, outnumbering all expectation. This number of sold copies was quite evenly separated on all available platforms.

One Month: 180k Copies Sold



Even one month after release PC as well the Nintendo consoles had a more or less even share of sales, which by then exceeded about 180,000 copies.



Of course, the obvious reason for making »Shovel Knight« and going in with this doomed perspective was that we really wanted to make this game, and we believed in it! It was a passion project! Not only that, we met so many great people during the course of the Kickstarter campaign who believed in it, too! The world needed »Shovel Knight«, we needed to develop it, and we didn't care if we had to sacrifice ourselves to do it. We also wanted to control every part of the game, from marketing to development, and were not willing to relinquish any authority on the matter. This we think is obvious from our Kickstarter!

But what isn't obvious is that we knew that »Shovel Knight« would be a success. We didn't have any doubt in our mind because we had years of experience budgeting and developing games (and sacrificing ourselves to that ...), and we had some concrete numbers to back us up. That's why we felt confident planning so far ahead, and saying we would present the bunch of Kickstarter stretch goals as free updates in the future.

What concrete numbers do we speak of? We're talking the 15,684 people that supported us during the Kickstarter campaign. Other developers have estimated that preorders predict first-week sales by 200 to 400 percent. That means »Shovel Knight« was guaranteed to sell two to four times the 15,684 backers in the first week, or in other words, 30,000 to 60,000 copies in the first week. That of course seemed crazy to us ... as successful games in our portfolio barely broke 50,000 copies sold. But we weren't going to fight the numbers we arrived at. We felt a level of security, so we instead prepared ourselves for the brutal 12 months ahead.

This all goes to show that planning isn't everything! We felt confident, but in the end we missed our target – remember how we didn't release our game on March 31st? Yep, that was also the point that the budget was gone, and when our reserves ran low – also known as »we ran out of money« (actually it was March 1st).

We stopped any and all spending that wasn't absolutely crucial to the game and the game's Kickstarter. Having already budgeted out and frozen big amounts like Kickstarter reward costs, we were down to the day to day. Electricity? Needs to be on. Internet? Need that too. Dirt Letter Envelopes? Order 'em. Supporting ourselves? Well ...

We ended up operating for five months without money or payments to the team here. It was a difficult period, where some of us were awkwardly standing in front of cashiers having our credit cards declined, drawing from any possible savings, and borrowing money from our friends and family. But we made it to the other side!

First month sales

Finally, we released the game on June 26th! And now, the moment everyone has been waiting for! How much did the game actually sell?! Let's jump right into it! In the first week, if you take out our Kickstarter presales, we essentially sold 75,000 copies!!!! Absolutely insane! It blew our two-times to four-times pre-sale expectation out of the water!

There are a few reasons we think our preorder prediction ended up being such a conserva-



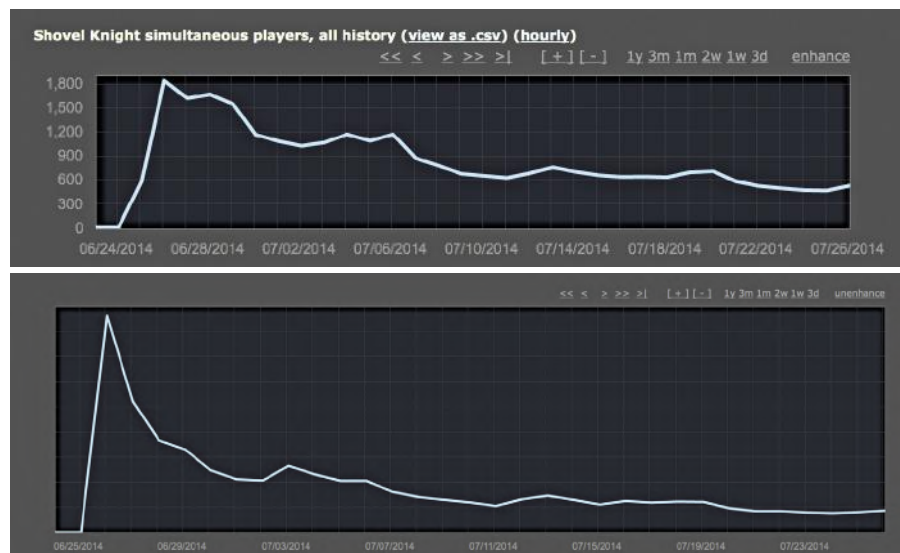
Valve did warn not to release a game during Summer Sale, but »Shovel Knight« could compete against AAA games that were insanely cheap.

tive estimate. One is ... we were a Kickstarter! That's not quite the same thing as a preorder. People could only attain the preorder for a limited time, so the stats aren't 100 percent aligned with what a preorder would do. We think due to the PC nature of Kickstarter, we saw a higher amount of sales on Nintendo's platform when the game actually got released. Nintendo users are more inclined to buy the game day one rather than through a preorder on Kickstarter. Finally, the preorder prediction isn't a set in stone statistic, but more like a guideline for what to expect. We had done a lot of promotion and marketing at conventions and on media sites to prove ourselves over the course of the year, and we think people responded to it in kind! Nintendo also did an amazing job supporting us through their store placement and own marketing channels.

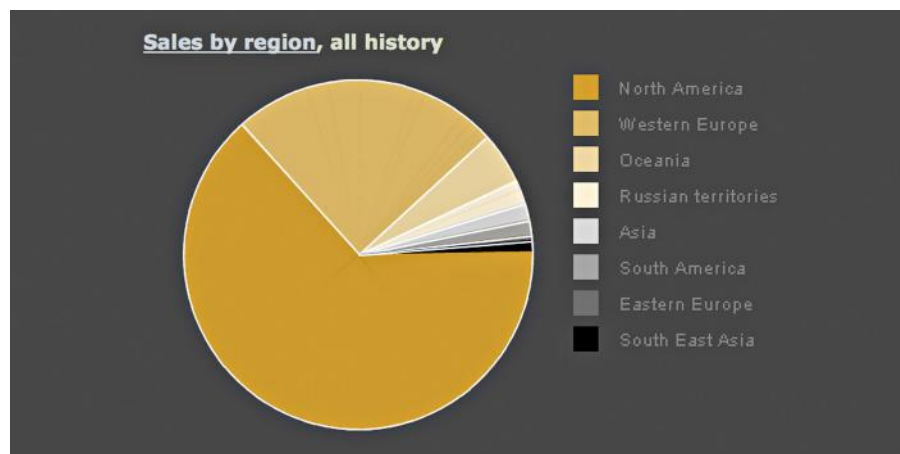
Not only did we outsell our expectations, but we also made it to the top of all of the charts! While we did everything we could to spread the word, this could not have happened without everyone's enthusiasm out there. Each and every positive impression, playthrough video, stream, tweet, fan art, recommendation, critical comment, email, vote, and review made this possible! »Shovel Knight« hit the number one spot on the 3DS, the number 2 spot on the Wii U, and we were among the top 10 games on Steam. Did we mention that we were complete buffoons and, against all of Valve's advice, we launched »Shovel Knight« during the Steam Summer Sale when we had major bargains as competition?! Yeah, we're crazy, but we still managed to crack the top 10! We don't know guys ... SteamWorld Dig sure sounds more appealing at two bucks. Also, while we hit 2nd on the Wii U store, it should be noted we were competing with a bunch of free games that were being given away as part of the »Mario Kart 8« promotion. I wouldn't expect us to beat a game that sold more than two million copies, where each person got to choose a free game on the eShop! So we're pretty proud we even came close to Wind Waker!

So that's a great first week, but of course, sales commonly take a quick dive after launch, so where are we sitting after a month? Here are the current totals up to July 26th:

Overall, we're extremely happy with the sales! We don't have sales data on many other digital games to offer a comparison, but we do know it's doing very well for our company – meaning the six of us! This means we can keep our current course and continue making the stretch goal content without fear of disbanding or seeking additional funding. It means we can continue living our dreams producing awesome games for our incredible fans (and soon-to-be fans)! And we still have lots of markets to release the game in ... although some of these markets are typically not selling as good compared to America, they will definitely help contribute to our next



On Steam a lot of statistics can be looked up and both the »Simultaneous players« (top) as well as »Sales« (bottom) show that a game is most successful in the first week after its release, while especially sale numbers drop quickly after the first hype.



During the first month after release »Shovel Knight« was not yet released in every territory, however Yacht Club Games did notice some sales in Asia or Europe which for example were still planned to get a release later on during that time.

efforts. The steady supply of free stretch goal content updates should also provide a nice boost in sales once each is released.

Thank You!!!

After the Kickstarter project ended, we all discussed our sales goals. The number that most of us settled upon was ... when all was said and done, 150,000 copies sold in total. That was our high point! So thank you, everyone, who supported us and made it possible for Shovel Knight to surpass that in just the first month. We are blown away! It's been a crazy ride!

But we're not done yet though. We're going to keep working our hardest to keep this game loved, played, and selling like hotcakes. Our new goal is the quest to one million sold, to truly solidify »Shovel Knight« as a modern platinum classic! For all of you who had doubts that a Wii U game or a digital game on a Nintendo platform could sell, we hope these facts show that a good game on any system, marketed the right way, can sell.

David D'Angelo





SHIFTING THE FOCUS

WAR FROM THE CIVILIANS' POINT OF VIEW

War games usually equal first person shooters. 11 bit studios took an entirely different approach: In »This War of Mine«, the team focuses on the struggle for survival of civilians. Karol Zajackowski explains the challenges this project posed during development and how its grim topic changed the team.



Karol Zajackowski
is PR & Marketing Manager
at 11 bit studios.

Karol has been in the industry for almost a decade. During that time he has been involved in some of the most successful projects from Poland, among them the first two »Witcher«-games and recently released »This War of Mine«. He secretly dreams of one day polish gaming companies will get tired of Eastern European winters and all of them will move somewhere south, where he could just sip drinks, lay back and do his job straight from the sunny beach.

»We wanted to show how war would look like for the majority of us.«

Making Games How did you get the idea for »This War of Mine«?

Karol Zajackowski We generally do some brainstorming once we finished a project. Everyone in the office is allowed to throw in some ideas. »This War of Mine« is the result of one of those discussions. When we finished »Anomaly« three years ago, we were looking for a new idea. Our CEO Grzegorz [Miechowski] said that he recently started reading about war and the situation of civilians in those conflicts. What he pointed out to us was that usually, when you read about war or watch movies about it, the civilians are not really part of the story. Usually you'll only see the soldiers or terrorists, but never the civilians. But obviously, they are a really big part of that. When you have a conflict, the army is about ten per cent of the people involved. The other ninety per cent are the civilians living in the territory.

Then we started digging on that topic and we found a lot of interviews, videos from Sarajevo and the Balkan conflicts for example. We saw that part of the story that hasn't really been shown so far and it was really touching. Those are not the people fighting for freedom. Those are stories of people trying to survive and keep their families alive. And they show how sudden things can change. If you think about Sarajevo, the Balkan conflict ... those were normal cities and normal European countries, people were going to work, kids were going to school. And suddenly, in a matter of a few weeks, everything

that people knew was gone, there were bombs falling down and snipers in the streets. It was really shocking for us. And it can also happen today, the world you know can turn into warfare in a few weeks if you have bad luck. We wanted to show that. We wanted to show that war can happen anywhere, anytime and how it would look like for the majority of us. Because, let's be honest: not many of us would be soldiers. I mean, we can play some games and stuff like that, but none of us were trained to be soldiers. So in case of a conflict, we probably wouldn't be the ones that are fighting but we would be the ones trying to survive in the ruins of the city.

Making Games Did the research for the game affect you on a personal level?

Karol Zajackowski We obviously knew from the very beginning that it would be a hard topic. But I guess none of us realized how creating such a game would influence the people. If you read such a story in a magazine then you can say »okay, that's a really sad story«, but then you move forward. However, when you spend days and weeks and even months working on and researching those topics and you hear hundreds of terrible stories of people that lost their families and their homes – you'll see that after a few months, you start to look at things from a different perspective. After you read 100 stories about kids losing their families or husbands losing their wives, the future of



The Siege of Sarajevo (5 April 1992 through 29 February 1996; the image shows heavily damaged buildings in the Grbavica district) was one of the main sources of reports for the team during the development.

people destroyed in one day because a bomb hit their house. Seeing the pictures however is even worse. You can read about a destroyed city, but when you see the pictures or videos, it really hits you. Humanity is flying to space and doing other really cool stuff, but at the same time, we are doing such horrible things.

Look at Ukraine for example. Three years ago, they held the football Euro Cup. They built the airport and the stadium for it. When you look at those pictures today, you see that it only takes some people and some time. I mean, Donetsk is no more. It's just a ruin of a city. And the airport? It's like a big hole in the ground. That really hurts you. We weren't really conscious about that until we went to one of the expos and were talking with journalists. And they started telling us »When you talk about that game, you look really sad and you look like you're exhausted because of that.« Suddenly we realized that it influenced our lives. I don't think anyone at 11 bit studios thinks about war the same way they did before. It changes you. Obviously, war is part of our reality. There has always been war and there will be war in the future. But I believe many of us started asking questions like »What can we do to prevent such things?«

Making Games You talked about article and reports that inspired the game. How close to that source material did you stay during development and where did you draw the line?

Karol Zajackowski We haven't actually copied any particular situations one-to-one to the game. We managed to contact real survivors and talk to them face to face. When we heard their stories, we made sure that, due to respect to those people and their lives, we not allowed someone's story to make it into the game one-to-one. When you play »This War of Mine«, it's rather the essence of war and the struggle of the civilians. We looked at the situations that were happening the most, the challenges that people faced. We also can't say that the game is about one particular conflict. We didn't want to do anything like that. But thanks to that, anyone playing the game can imagine that it might be their country. When people from Israel were playing that game, they thought it's about their

country somehow. When Palestinian's were playing, it's the same for them. When people who survived Sarajevo played the game, they also thought »Hey, it somehow resembles my situation back then.« So it's no specific war, but rather the general image of war.

Of course at war, brutal and bad things happen. We tried to keep the game as close to reality as possible in that aspect. There are people dying and there are hard decisions to be made. You can be a good guy, you can be a bad guy, a peaceful or a violent one. But we never wanted to provoke somehow. Take for example a school bus full of kids and it gets blown up. We could easily create such a scene in the game, but what's the point? We wanted to show the grim reality of war, but there's always a line. If there's violence in the game, it because you see violence in the world. People are killing each other, they're stealing food, beat each other for stuff. That's what you see in the game. But you're not able to slaughter the whole building just for fun. That's not how it looks like. All the violence we put in the game serves a purpose. There's a scene where you see that there will be rape. You have to make up your mind: you can run away or you can stop the man. But no matter what you do, you will not see it happening. Because that would be too much in our opinion. It's not

»None of us realized how creating such a game would influence the team.«



»This War of Mine« doesn't depict a specific conflict: »It's rather the essence of war and the struggle of the civilians.«

» We could easily put an exploding bus full of kids in the game, but what's the point?«

necessary to show that to deliver the message and confront someone with making a hard decision. If someone is threatening a woman with rape, what would you do? We believe that you can make a decision and really think about it without actually showing the bad stuff. There is a line of course and we didn't cross it, but you can see a lot of horrible things.

On the other hand, our game doesn't contain a lot of gore. When you play a lot of war games, you can see heads blowing up or arms being chopped off. You won't see such things in our game because it's not necessary. That's not our message. So we don't see the point in making it totally brutal just because we're capable of that.

Making Games How does development differ when you're working on a game that isn't supposed to be fun for the player but rather focuses on conveying a certain mood?

Karol Zajackowski I have been part of the industry for over ten years now and remember working on »The Witcher« and other games. What I noticed is that when we worked on other games, there was always some kind of spark, you know? People making jokes and stuff like that. When you create a game, there are always some easter eggs, stupid jokes, fun animations – a lot of funny stuff that somehow comes up during the process. What was surprising for me with »This War of Mine« was that everyone in the team knew just how serious this game was. Nobody even thought about creating easter eggs or make fun of a character model. Because for us, everything in the game – the models, the environment, the mechanics – those are the parts of the bigger, serious image. And we somehow all understood from the very beginning that we had to respect that idea. That we had to deliver a good game in every single aspect.

While we worked on the game, we were very quiet and serious for a gaming industry company. We were almost moody and not talking to each other. Everyone of us had an image in their heads and it was a big challenge for us

because when people think about war games, they think about heroes and bad guys. Our game might be the first one to show that every single situation can be shown in a different way. We can make fun of it or we can make it serious but exciting or we can make it a deep and interesting experience. We were really focused on that. That was a part of the development process on every single stage. We had that in mind when we were drawing concepts or when we were creating the music. The respect for the idea and the serious atmosphere that we wanted to keep in the product was present at every single stage. When we finished the game, we somehow felt relieved. We delivered the game and it met the expectations of the people and our own. It felt like that lifted a rock from our hearts. We were finally done with this somehow beautiful but very painful experience. For a lot of people in the team that was a really tough time. I'm glad we are done with it. I guess you could say we survived it. We delivered something that is being discussed by media throughout the world and will be discussed for years to follow. But we are still glad that we are done with it. Because now we can do something more cheerful and different.

Making Games Were there also differences in regard to gameplay mechanics?

Karol Zajackowski The biggest challenge we had here is that we wanted to keep the atmosphere and the feeling that you are the civilian; but it's still a game, which requires certain mechanics. There must be gameplay and things the player has to do. There has to be a challenge and progress. It was hard, because we had to transform those inherent elements of a game into something that would suit our idea. When you're trying to make a serious game about war and civilians, you can't just put a score in there, right? There's also health. You know that each character has to have a health indicator because you have to know what its current condition is. But you can't just put a health bar above the character because that would spoil the experience. We struggled with such things. How can we show health without destroying the immersion? Those are tricky questions. So when you find food, for example, you won't know whether it will heal you five health points. Or if a gun will deal eleven damage points. You have to transfer the stats and mechanics somehow to put it in the cloak of experience. To see the world like you really see it. When you see a vegetable and canned food and a piece of meat, you can imagine which one will be more valuable for you and which one will provide you with more energy. We had to put that into the game. That was the biggest challenge for us.

The game also had to have some ultimate goal. It was always about survival, but what would be the path? How does it start, what will the player have to do on the way to reach that goal? That's something that had to be



Adapting inherent gameplay mechanics has been a challenge: »You can't just put a health bar above the character, that would spoil the experience.«

done without using well-known mechanics. However, the game still had to be understandable and though the game is not fun to play, you still had to enjoy spending time with it. To reach that, we had to provide some gameplay. We finally came up with the idea for the shelter and other locations. We divided the gameplay: During the day, you can take care of your house, which is somewhat like a simulator. You have to maintain your shelter and take care of your characters. And then you have the night where it's more like an adventure game and you have to scavenge and make hard decisions, tackle challenges that come up and fight enemies. It's like two different games, but combined they cover all the things that we believe should be covered in such a game. So we had to keep the game a game, but make it something more than just clicking at the same time. That was the biggest challenge for us and I believe we succeeded.

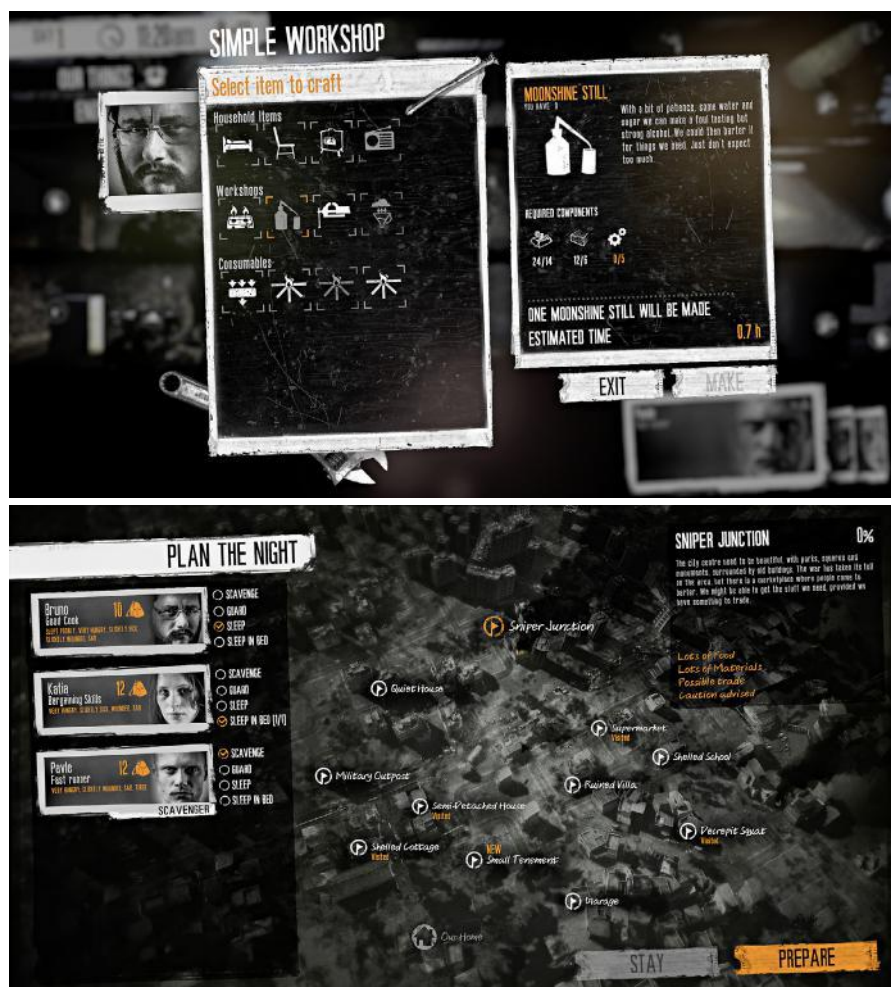
What we appreciated the most was that people that survived real conflicts, who had never played any game, heard about our project and they wanted to see it. And those people, who are not into games and are not players that are aware of the mechanics, felt the emotion that we wanted to put into the game. For them, it was an experience and for some it was so hard they couldn't finish it because the emotions the game caused were too much for them.

Making Games How much did the game you released differ from the first draft or what you intended for the game to be when you started working on it?

Karol Zajackowski I would say that the game hasn't changed much. I think it is still very close to the concept we had at the very beginning. Of course some things change during the development process, for example the characters. Should we create a custom character or should they be generated? But the game looked like this from the very beginning. We didn't want to make an artistic game, but we wanted to make a game that matters. And we were afraid that people are maybe not ready for such games. That's why we haven't changed the game much. At some point we decided that even if the game wouldn't be a giant commercial success, it would still be the game we wanted to make. It might be hard and some players might have trouble understanding it who are used to games like «Call of Duty» for example, but that's the game we wanted to have.

Making Games How did you figure out for yourselves how much players can actually take before it becomes too much?

Karol Zajackowski We had a few focus tests during development. Every time we work on a title, we decide at some point that we are so involved in the project that we might be no longer objective and honest about what we are doing. So we have focus tests with diverse groups of people: men, women, different age groups. We always



11 bit studios divided the gameplay - during the day, the player can craft items and take care of the house (upper image) while the night is used to scavenge buildings (lower image): «It's like two different games.»

try to have as many people as possible to give us feedback whether certain things are too hard or too easy or maybe not done in a respectful way. That's really valuable because they give you the most objective opinion on your games. Sometimes they tell you that you did a great job, other times it turns out that ninety per cent of the people don't like a certain part of your game. Of course if they don't like it because it makes them feel uneasy we don't have to change it - this game is supposed to make you uneasy and think about your life. But when people say «this scene of murder is unnecessary», that's bad. In case something like that happens, or course we have to think about it. We had three focus tests: The alpha and beta builds and a few weeks before release. That was our main source of checking what he had achieved so far and what the feedback on that was. We also showed it to a bunch of journalists from the very beginning and we were always counting on what they would say, as well. Because they see a lot of other games and can provide valuable feedback from that point. We have our internal feelers out, but it's always good to confront that with other people that don't have the emotional involvement and attachment to the project.

»We wanted to make a game that matters.«



Pointing out differences was very important when pitching the game: »When people hear that there is a war game, it takes about five seconds until they get the image of ›Battlefield‹ or ›Call of Duty‹.«

»We focused on highlighting the differences between our game and other war games.«

Making Games »This War of Mine« is about survival. How difficult was it to find the right balancing for this type of gameplay?

Karol Zajackowski Checking the difficulty curve was also something we addressed in the playtests. However, it was tricky because there are experienced players and for them stuff would be easy that would be hard for newbies. Even I have trouble finishing the war and I know the game pretty well. We released three patches already and all of them improved balancing. Apparently some things that were easy for us turned out to be difficult for the players. With the initial balancing, we mostly trusted our people in the company and the focus tests. Those were the two biggest benchmarks. And then of course we read the feedback after the game was released. We are constantly reading support e-mails, forums and facebook pages. If one person says »that's a really hard moment, I'm not able to survive past day 5«, then maybe the player is not meant for that type of game. But if hundreds of people are writing that something is wrong, you have to accept that they're right.

The good thing about survival games is that if you're into that genre, you accept the fact that failure is a part of the process. No-one has ever finished a survival game in the first attempt. It's just not possible because you have to learn in survival games. And you mostly learn by succeeding or ultimately by failure. You probably have to die two or three times in »This War of Mine«. The very first thing we noticed is that people assume that in a war game, you are the strongest guy around. So when people saw bandits, ninety per cent of them tried to kill those bandits. And obviously that didn't work because when you're a guy that used to be a mathematician or a cashier, you are not a challenge for guys that are holding guns and know how to use them. That's the way that people learn a different approach. Usually the first few times are failures, but that's okay. The good thing is that when you fail one or two times and then overcome the moment, it's the greatest reward for you. If you want the game to be easy and finish on its own, this probably isn't the right

kind of game for you. We're still implementing new things that balance it better and are doing some fixes, but balance is something that came mostly from our experience.

Making Games Was it harder to find a publisher for »This War of Mine« since it's not exactly a mainstream game?

Karol Zajackowski We only have a publisher for the boxed version. Obviously it was harder because usually, people are looking for games that are fun. »This War of Mine« isn't a game that you can turn on for 15 minutes when you come home from work, play a few matches and then move on to something different. It was harder because on the one hand, publishers are afraid that it's not a game people want and on the other hand it might be too brutal. So it really wasn't easy, but luckily, we found the people who could see the true depth of game and who didn't only see the bad sides and the risks. They also saw the benefits that this is a game that will be spoken about and that will make people think. After we released the game, discussions began about how games are evolving and becoming more than just games and entertainment. I hope that the success of our game, and the fact that we found a publisher, will allow more developers to create such titles. Because I believe that we have too many AAA games that are repetitive things. There are certain titles that will come out in October and November and we play them year after year. We just hope that through our game, both publishers and developers will see that it's worth to take the risk sometimes and to have the courage to create a game that you will like to make and that really matters. Maybe »This War of Mine« will help to find funding for more such things. Luckily we are an experienced company, so even if we hadn't found a publisher, we would have made the game. But there are some smaller studios that need some help and I really hope that after »This War of Mine«, it will be easier for those small companies to get some help from publishers or maybe Steam or organizations or in another way. Because then we'll see that

people have grown enough to see the games and play the games that are something more than just plain and simple fun.

Making Games What did you have to change about your pitch to publishers in comparison to a »normal« game?

Karol Zajackowski From the beginning, we were focused on showing people that »This War of Mine« is not just a game but an experience. We wanted them to see that it is like a simulator of being a civilian during war. When people hear that there is a war game, it takes about five seconds until they get the image of »Battlefield« or »Call of Duty«, where you just go and shoot someone. We knew from the beginning that we had to say from the first minute »No, when we say war game, we only mean the setting. It is a war, but there are a lot of differences.« So we had our focus on highlighting the differences that separate our game from other war games.

We put all our emphasis on experience and on the fact that this topic, civilians in war, hadn't been shown before and that we were referring to other parts of culture, like movies and books. It's somehow easier to find similar depictions there. We were trying to convey the argument that movies were also just for fun when they first came out. It took about 50 years to create the first very serious movie. We always said that the same happened to games. The first games were just about shooting alien ships or stuff like that. Luckily, we are now at the moment where games can be something more and »This War of Mine« is the best example. You have a game that is not just a game and that is also perceived as more than a game. And we are convinced that you can be a part of that movement. We can either try to stick to mindless, simple games or we can be a part of the idea that is changing the gaming market and industry.

What we also tried to stress when we were talking to journalists is that we spent hundreds of hours on doing research on the topic. Because we wanted to prove that the idea of war that we show in the game is another than the idea of the guys sitting in their office in Warsaw and make a game about how they imagined the war. We wanted to emphasize the fact that we read hundreds of documents and watched just as many videos that show the real picture so we would be able to bring them to a game for the first time. If you do a science fiction game or a fantasy game, it's always somehow on a different planet. However, our game is as realistic as possible with the capabilities we have today.

Making Games The games you made so far have been extremely diverse and you mentioned you're your next game won't be as emotionally taxing as »This War of Mine«. Why did you decide to go down that path?

Karol Zajackowski Our next game will be really simple and just for fun, like »Funky Smugglers«. That was a game we did in the past that

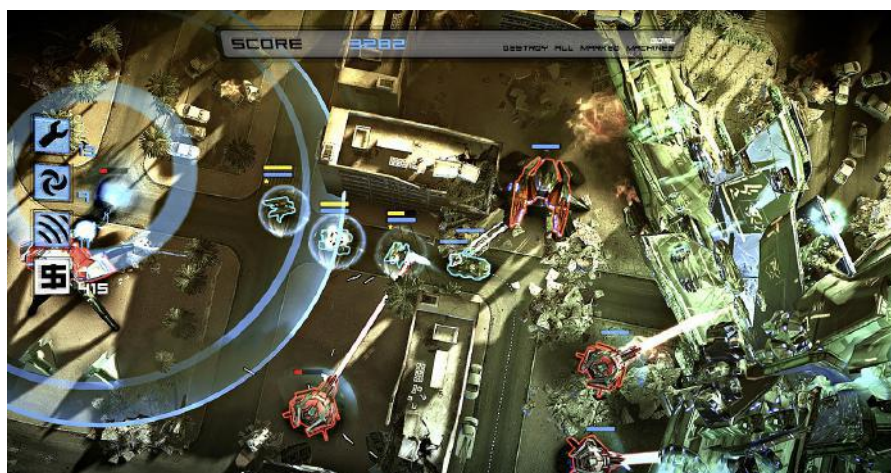
you can play for example when you're on the train to work. We realised that it's best to create an idea for a game and then to put all your heart and everything that you can imagine in that single title. When you are starting to think about »What will I do with the sequel?« or »Maybe we will make three games out of it«, you are also starting to remove things because you want to keep them for the future. However, when you focus on the first and only game that you want to do on a topic, it has to be the best game it can be. You can't hold back something for future installments because you want to prove yourself in the second part.

When you change topics, it's really refreshing for the team. Back when we were working on »Anomaly« 1 and 2 and Korea and the mobile version ... well, no one likes to work on the same game for 20 years, like in »Anomaly« we have dealt with future robots and future citizens. More or less it is somehow the same. And then when we moved people to »This War of Mine« we noticed that everyone was fresh and full of ideas because of the new setting. We've seen that people have a lot of great ideas when you're not making just one type of game because those ideas don't suit that certain type of game.

Right now we do the same with another project. There were some cool things we came up with during »This War of Mine«, but we weren't able to put those mechanics or visual ideas in the game because they just didn't fit. By saying that our next game wouldn't be so emotionally full in comparison to »This War of Mine«, I meant that it wouldn't make you totally sad. Our next game will be more about playing and enjoying it, but without the drama. We don't want to make another game of emotional torment for people. It won't be so hard, but it will still be deep and sophisticated, but on different levels. It will be different from anything we've done so far. It will also make people think, but about other topics. We will also try to engage the player – so when you close the game, you'll still think about some aspects of it after playing.

Interview: Patricia Geiger

»We don't want to make another game of emotional torment for people.«



Moving onto a new project after several »Anomaly« games proved to be refreshing: »Everyone was fresh and full of ideas.«



HOW WORLD OF TANKS CHANGED FREE2PLAY GAMING

White Russian developer Wargaming tried to create a blockbuster for several years but in the end a game no publisher was convinced of led to Wargaming growing to a global player with studios all of the world. Victor Kislyi shares some history about the company he co-founded.



Victor Kislyi
is co-founder and CEO
of Wargaming.net.

Victor founded Wargaming.net in 1998 turning his passion for gaming into professional practice. Wargaming shipped 12 titles in ten years including «Massive Assault» and «Order of War», yet it was not until the launch of «World of Tanks» in August, 2010 that the company became one of the leaders on the MMO market. As the company's CEO, Victor leads the team setting strategy and boosting the company with his experience and enthusiasm.



For its second game «Massive Assault» (the screenshot shows one of the successors called «Massive Assault Network 2») the team had learned its lessons and started attending events like E3 and GDC socializing with publishing companies.

Making Games Wargaming has been developing games for over 16 years now, but made its entry to territories outside of the Commonwealth of Independent States (CIS) about four years ago. When did you decide to reach new borders and what was the main cause to actually succeed in those territories?

Victor Kislyi It might be surprising, but Wargaming has had a global perspective since the launch of its very first game «Iron Age» more than 16 years ago. The small title that was developed by me, my brother and a couple of friends had only eight players – an American, a German, a Russian, and the team of developers.

It sounds funny, but it shows that we have never been a CIS-focused developer. We produced 12 games before «World of Tanks», and with every title we learnt a bit more about how to work with major publishers and talk to the players around the globe.

«World of Tanks» was a big breakthrough to the Western markets for us. The game has a universal appeal; it's easy to learn for any player anywhere in the world. We launched the title in Russia in 2010, and it took the market by storm in several weeks. The move to the European and American markets was a natural step forward, but it was a big challenge for our team. We were determined to deliver «World of Tanks» to gamers everywhere. Our focus, respect for the players and hard work propelled us to the top of the league.

When we created Wargaming 16 years ago, we had a vision to create better games. At the

start, there were less than ten of us. We were a close bunch of enthusiasts, united by a passion for video games and firm belief that we could make a difference.

And over the course of sixteen years, we plugged away at this vision. With a lot of hard work, never-ending enthusiasm, and help from our partners, we've grown Wargaming from a tiny development studio to what it is now – an established international business with more than 3,500 employees and 16 offices across the globe.

In 2000, when our first project was released – «DBA Online» – there were a lot of people who thought we could never make this a successful business. But we knew that the same thing had been said before about other game developers, and there was no stopping us.

After «DBA Online» we produced a series of futuristic turn-based strategy games under the «Massive Assault»-franchise. 2007 marked the year when we switched to a new genre – real-time strategy games. Two years and three projects later it became apparent to us that we were sort of swimming against the mainstream. The world was in the midst of a massive transformation. Information was exploding and entertainment was going digital.

Back in those days, we saw an opportunity. At a time when the games industry was largely the domain of boxed titles monetized through a subscription model, we had a very radical idea: free, online, user-focused games. We concentrated on quality and the end user instead

of focusing on extorting maximum money in the shortest amount of time.

»World of Tanks« has become the culmination of ten years' worth of work in different genres, and the introduction of a much more player-centric paradigm.

Making Games »World of Tanks« came out of nowhere and is still not at the end of its lifetime circle – how did everything start and what was the key to success?

Victor Kislyi We came up with the idea for the game back in 2008. The concept was revolutionary even for us. Wargaming had produced a number of strategy games before, but in our opinion »World of Tanks« was fresh and unique. The game was different from anything else you could find on the market, and it was a bold decision for us to start developing it. We had a gut feeling that we were doing the right thing. We invested all the money we had in creating the title, put a lot at stake, and in retrospect, it was the best business decision we ever made.

There are many things that played their part in our success. We had a unique team that was experienced and passionate enough to build a great game. We listened to our players. We talked to the media. We believed in the title that we were making. We think that »World of Tanks« stands out from other MMOs due to a neat combination of accessibility, simplicity, rapid re-playability and choice that it offers the player.

You don't have to be a hardcore FPS player to join in, the player can figure out enough about tank combat within the first few minutes, and can have just as much fun as the very best players without worrying about ranking or ratings or whether or not you're using everything to its fullest potential. At the same time, the game has enough substance to appeal to players with different skill levels. Although you can get to grips with combat mechanics in ten minutes, the true joy of the PvP combat of »World of Tanks« opens up when you realize how much there is to learn. For example, it doesn't require great skills to drive a light tank around in circles, scouting for the big guys, but if you'd like to be able to dodge blasts and dish out some return fire from the best places on any particular map, there's a steep learning curve to climb.

Whereas in most MMOs, the mechanics force a player to keep up with a predetermined path of growth like gear or level grinds in a game, progression in »World of Tanks« opens up new tanks that are unique to their play styles. Battles are rewarded according to how well you and your team did in the battle. Credits and experience are then used to work your way through a tech tree. Each tank has equipment that needs to be researched, bought and installed. Tank crews come with a list of skills they can master, too. The better the crew, the better they can use the tank to its fullest capacity. Of course, crew training is as important as getting better tank



In 2000 Wargaming published its first game »DBA Online«. Due to bad accessibility the game was only moderately successful.

modules, because crews can make two identical tanks perform vastly different and equipment further improves tank characteristics.

The Free-to-Win monetization model is another factor that contributed to the popularity of »World of Tanks«. You can play it absolutely free. At the same time it offers ways for people to spend some cash. They can buy customization items and rare vehicles that don't affect gameplay. They might pay to level up faster and unlock items earlier than someone who plays the free mode. However, it doesn't mean they will perform better, because the gameplay is built around skill, not money spent on premium content. Microtransactions were added not because we wanted to force payments, but because we wanted our audience to have

»The Free-to-Win monetization model is another factor that contributed to the popularity of World of Tanks«.



The strategy game »Order of War« was released 2009 by Square Enix and despite being a different genre reminds a lot of »World of Tanks«.



In 2010 Wargaming finally released »World of Tanks« and decided quickly to not only focus on Eastern European countries but also on Western markets. Today more than 60 million registered users are playing the game that reached two Guinness World Records.



The next installment in the »World of«-series will be »World of Warships« for which no released date is yet set. But Wargaming is still thinking about games in different genres for the future.

»Innovations, experiments and disruptive changes shape the industries today.«

choice - about the personalization, the status, the progress, whatever it is that excites them. The business model we use grants equal opportunities to all players. They see that we treat them with respect and treat us the same way.

But acclaim didn't come overnight. We achieved success through hard work and overcame many challenges along the way. Servers couldn't handle the load. We had to deploy extra data centers, build the entire system of customer support from scratch, and manage the game updates at the same time. The list of issues needed to be addressed was long, and the team grew.

Making Games You were showing an early version of »World of Tanks« on various trade shows and events. What was the reaction of the major classic publishers?

Victor Kislyi We received a lot of comments about the game. Some said that players wouldn't feel »a deep bond« with the tank, some didn't understand an MMO without fantasy characters. Honestly, I don't remember all the negative feedback. Every extreme reaction was one more reason to push harder.

Making Games Did you have to adjust elements of the game to be successful in other territories?

Victor Kislyi Initially, we aimed at the CIS market only. The success of »World of Tanks« in Russia and CIS empowered us to expand globally. It was kind of a large-scale – and stressful at times – focus test that we had to analyze before entering other regions. And we did.

In retrospect, the experience we gained while getting the game up and running in Russia also helped a great deal when we were tuning technical and informational support for it in Europe and North America. We experimented a lot when we were developing the title. »World of Tanks« has always been a work in progress title. We have been evolving the game for four and a half years. It's the beauty and the curse of this huge title. You can always change something, improve and upgrade.

Making Games Has there been any point, where you thought »That's it – we will not make it into new markets«?

Victor Kislyi We see every market as an opportunity to meet new players. We try to bring a fun gaming experience to gamers, whatever their language, different mentalities and playing styles are; but they all have one thing in common: they enjoy action-packed battles and military machines.

New markets are a great test for a company. You can learn a lot from new challenges. There might be some peculiarities in different regions, but we're more than satisfied with the results that we showed so far. We receive huge support from our players in all territories, and that is a major motivation for us. We have never considered going away from any country or market.

It does not matter how hard it might be to deliver or support our games, we are doing it for the people that matter most – our gamers.

Making Games »World of Tanks« was not planned to be such a huge success, when did you realize, that it can become such a phenomena?

Victor Kislyi We launched the game in Russia on August 13th, 2010 and just a few weeks later we realized the success of the title to beat our expectations. The number of players was way bigger than we predicted. We understood it could be a massive hit, and we started working even harder.

Making Games Wargaming is in private hands from the very first moment until now, how hard was it to resist offers from investment companies or buyout options?

Victor Kislyi Me, my brother and a couple of our friends created Wargaming more than 16 years ago. Together we went from a five-man start-up to a renowned publisher and developer with more than 100 million players. It has not been an easy journey, and we went through our fair share of tough moments, but we have never considered selling the company. I am passionate about the company. I have always been inspired by the people I work with. I do love games – there is simply no need to sell what you love!

Making Games Our cover story is named »F**k Mainstream«, what do you think how this suits Wargaming?

Victor Kislyi Wargaming's story of success is pretty much about this slogan. Over the course of fifteen years, we've been lucky to span several genres, get the best out of them, and contribute to them with our games.

Wargaming started as a development studio. The company grew from what was a bunch of programmers and artists, working in dormitories and rented apartments. The company's first project »DBA Online« enjoyed very moderate success, because we somehow missed the entire accessibility issue. The topic of »military strategy« was too niche, too hardcore to drag a massive player-base; the elaborate game-play setup, manifold units, and sophisticated mechanics scared off the bulk of casual players instead of attracting them. We learnt the lesson and avoided the similar pitfall with the second title, the turn-based strategy project »Massive Assault«. While the dev team was prepping our second project »Massive Assault« for its release, the marketing team started attending events like E3 and GDC, building connections with publishing companies. That was the time when we learnt about the intricacies of publisher negotiations, contracts, and advertising deals.

Initially, Wargaming marketed its projects through third-party publishers. We worked with Matrix Games on the »Massive Assault«-series and with Square Enix on »Order of War«. When »World of Tanks« was shaping up, we started

looking for a partner to publish the game. And this is where we encountered a major hurdle. »World of Tanks« utilized the Free2Play monetization system, and no company would agree to manage a title like that. Looking for a momentary gain, they didn't believe our MMO game would make any profit. At some point we realized that it would be easier to publish the game ourselves than keep on looking for a partner.

Making Games How do you see various copycats of popular games, like yours? Isn't this a sign of being mainstream?

Victor Kislyi It's not easy to copy a great game. There have been many attempts to replicate our success, but so far they failed.

Making Games Would you agree to say mainstream means successful?

Victor Kislyi In today's IT- and gaming-world you have to find something new and unexpected if you want to create a great product. Innovations, experiments and disruptive changes shape the industries today. Being mainstream doesn't bring success, new approaches do.

Making Games Wargaming is quite famous for its impressive booths at trade shows or its events and parties – is this the way only successful companies can make a point or is this part of your way to do business?

Victor Kislyi Gaming is a bit like show business. You have to show yourself and your games, talk to the journalists, and entertain the players. Every trade show is a chance to brag about our achievements and meet the players – that's why we try to make every show legendary.

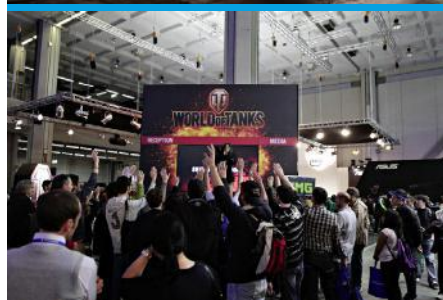
Making Games The name »Wargaming« is established in a lot of countries, although it could indicate that you only do games concerning war and the military. Do you think this could be a problem in the future?

Victor Kislyi Right now, we're focused on building the Wargaming universe with »World of Tanks«, »World of Warplanes« and »World of Warships« which is our latest title that will become the third part in our war series. As a naval combat game it seems like the perfect complement to our universe.

We have a very clear plan for the upcoming three to four years. I can assure you that we won't release anything as non-conventional for Wargaming as, for instance, horse games for girls. However, we might consider going beyond the MMO genre and developing a game that is nothing like the »World of«-series.

Making Games Let us talk about yourself as a person – how mainstream have you been in the past before being the boss of nearly 4,000 people, and how mainstream are you today?

Victor Kislyi Before Wargaming was started I was the guy with a master's degree in laser physics obsessed about military history and



Wargaming likes to compare gaming to show business and therefore always tries to impress with huge booths at trade shows or parties.

strategy games, which is not exactly the type of person you call mainstream. But at the same time I was not much different from my friends, I was just more passionate and curious about some things. Wargaming has changed greatly over the past years. As the company embraced new offices and new cultures, I had a lot of challenges, discoveries, trials that changed me. When you lead almost 4,000 people across the globe you have to learn and develop yourself.

Making Games Do you have any eccentric stories or funny anecdote to tell which underlines the slogan »F**k mainstream«?

Victor Kislyi I think that one of the ways Wargaming shows how different it is from other huge game companies out there is in the way we approach industry events and the parties we throw for the Wargaming-family. Just look at when we announced our partnership with FURY at E3 2014 – we had a real-life M4 Sherman tank near the entrance to the event! If you're at an industry event, then walk around the booths and tell me Wargaming's wasn't memorable. It's the way we interact with the crowd, the whole atmosphere we bring to these shows. There's a real energy there, and I challenge you to find that at any other booth. We're just fans who are as excited about our titles as those who play the game. We're all family.

And when we put on a show for ourselves, we go all out. For our 15th anniversary, we had tanks, aircrafts, live music, games – huge is an understatement, there were thousands of people present, but we believe that it's important to reward all the hard work that people put into our games. Our normal is another person's crazy – and it's definitely working. I'd say we're the most eccentric games company out there, and that's the way we like it.

Interview: Sebastian Weber

»We won't release anything like horse games for girls.«

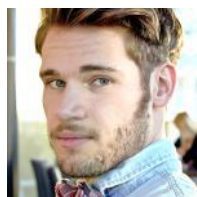


In 2011 Wargaming attended Gamescom with a somewhat small team (upper image) as the comparison to the team from 2014 shows, which is a good representation of the company's growth.



THE SUCCESS STORY OF THE BINDING OF ISAAC

Not even developer Edmund McMillen believed in his indie game *The Binding of Isaac*. And yet more than three millions copies have been sold. We're telling the success story of an unusual game about a naked boy in a basement.



Benjamin Wojkuvka
is freelance writer.

Benjamin writes about video games, web design and outer space. Other than that, he is currently working on his master's degree in business administration and is constantly looking for the perfect free2play-world.

Career suicide!« That's how developer and designer Edmund McMillen describes his game *The Binding of Isaac*. A deliberate suicide though. It seems that McMillen is almost determined to push the envelope and, at the end of the day, even to fail with his little Flash game. However, it can be said already the inventor of »Super Meat Boy« has been proven completely wrong: *The Binding of Isaac* has already sold more than three million copies.

A childhood shaped by religion

In order to understand how Edmund McMillen's mind works, one has to venture back to the developer's childhood and youth. McMillen comes from a difficult background strongly shaped by Christian faith and often struck by tragedy. According to the developer, the paternal side of his relatives consists of so-called reborn Christians. On the one side, they're trying to live a life of very strict faith, on the other they're all struggling with alcohol and drug abuse. They lecture little Edmund, among other things, that »Magic: The Gathering« and »Dungeons & Dragons« will make him end up in hell. But then there's his grandmother who is also an avowed Christian, but a more liberal one, encouraging the boy to make use of his talents. It is through her that McMillen learns to accept and appreciate religion as a part of his life.

Up to this day, McMillen has been looking for ways to help him come to terms with his childhood. His mechanism is to push the enve-

lope. He discovers that his drawings provoke reactions he enjoys in other people: disturbance, incomprehension, anger. He would draw dead babies, feces and basically anything that would upset the people around him. He enjoys being the black sheep of the family. It is that artistic approach that later also defines his career as a developer. He loves to upset people. However, with his first big game *Super Meat Boy* he doesn't stick to his guns. Out of love for his wife.

The »safe« *Super Meat Boy*

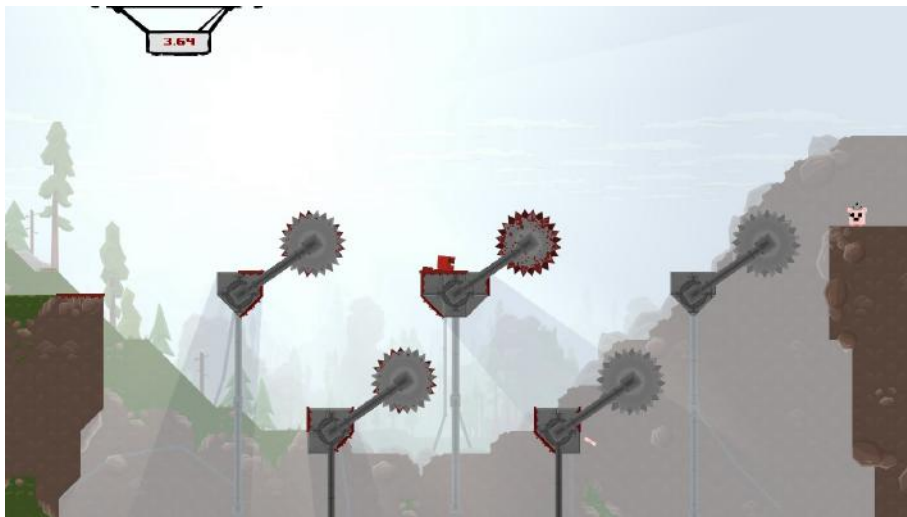
Super Meat Boy is released in October 2010, yielding sensational sales. Critics and fans praise the jump&run as one of the best of its kind and put it on the same level as *Super Mario*. In early 2012, *Super Meat Boy* has already sold more than one million copies. And yet McMillen is unhappy with the game. Not because it's a bad game, but because it's playing it safe. *Super Meat Boy* doesn't upset. The game about a cube of meat that sets out to rescue his girlfriend from a fetus with a PhD is much too calculated in the eyes of the designer.

He simply can't risk the career of his co-developer and best friend Tommy Refenes for the sake of a risky project. In addition, he wants to build a life with his wife Danielle. *Super Meat Boy* seems to be the best choice: The jump&run has been available for a while on the Flash-game website Newgrounds, enjoying great popularity when Microsoft enters the stage as a publisher. The rest is (successful) history.

The meat cube is a safe bet. And although McMillen sees the need for *Super Meat Boy*



The bosses and mini-bosses in *The Binding of Isaac* are a perfect example of McMillen's morbid design.



The hardcore platformer *Super Meat Boy* (left) quickly became a big seller. McMillen designed the game together with Tommy Refenes (right).

to be successful and is happy with the result, inside he still remains dissatisfied. After all, playing it safe is the last thing he wants; he much rather yearns to scandalize and to shock, just like he did with his game »Cunt« in 2008.

Genitals and religion

»Cunt« is the exact opposite of *Super Meat Boy*. It's calculated risk. In *Cunt*, a penis shoots sperm onto a vagina, just in the style of the Arcade classic »Galaga« from 1981. Obviously, McMillen wants to attract attention and to provoke, just like he did as a child with his drawings. But he also pursues the goal to make a name for himself as an (inconvenient) developer. The plan works out: Microsoft of all publishers comes along and offers the designer the opportunity to develop *Super Meat Boy* for Xbox Live Arcade.

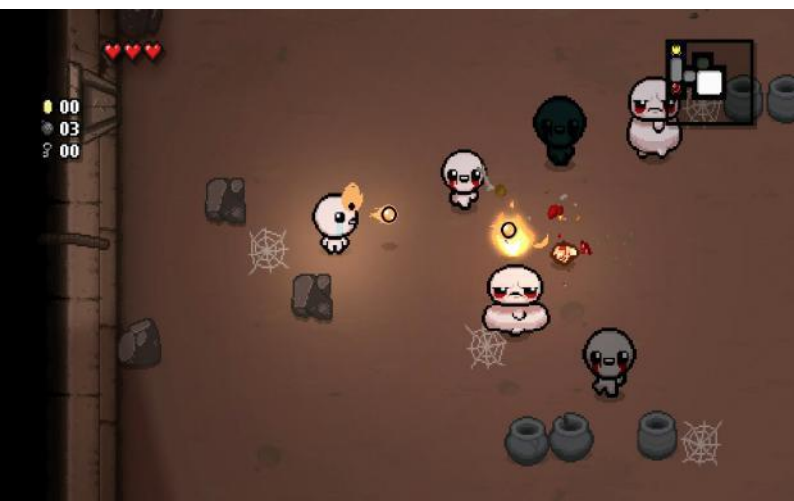
With the financial security that *Super Meat Boy* brings, McMillen starts to create something which corresponds more to how he sees himself as a developer: *The Binding of Isaac* can be put into the same category as *Cunt*. The designer wants to break away completely from the constraints of a publisher and societal values. In his own words, he wants to create »a fun little game« which »should harm his career«

and »upset people«. That's the kind of reward he wants for the success of *Super Meat Boy*: His partner Tommy Refenes takes a vacation while Edmund McMillen destroys his career. At least that's what he thinks. The game about the little naked boy who is driven down the basement by his obsessed mother looks promising. Isaac destroys the monsters in the dungeon with his tears. Items that are reminiscent of *Zelda* slowly disfigure the protagonist; the little boy turns into a monster. The levels are generated randomly as it is usually the case for roguelikes. With permadeath, the players only have one screen life, which means at Game Over they need to start all over again. Not only does *The Binding of Isaac* feature a taboo subject as its basic motive (child abuse and the consequences); the gameplay is also a hard nut to crack.

For McMillen, *The Binding of Isaac* is an outlet to confront faith. It is based on the biblical story of Isaac who is supposed to be sacrificed by his father Abraham to please God. McMillen changes the figure of Isaac to a boy who he partly sees himself in: a creative mind that is raised experiencing both the negative and positive aspects of religion. It's simply a highly unusual game. McMillen thinks it might become a cult



Edmund McMillen, designer of *Super Meat Boy* and *The Binding of Isaac*.



The Binding of Isaac: Rebirth features rooms that are up to four times bigger than in the original game.



On McMillen's request, *The Binding of Isaac* is offered on Steam for five dollars (or five euros) - against Valve's recommendation of ten dollars.

hit for a few people; but above all, it's supposed to take the developer out of his comfort zone and maybe even harm his career. McMillen wants to be the unconventional underdog and not »everybody's darling«. And in the beginning, his plan actually works out. Enter Nintendo.

Conservative Nintendo

The developer thought about which platform Isaac could be released on. Microsoft was dropped from the race due to contract squabbles over Super Meat Boy. Sony hadn't warmed up to indie games yet. So, only Nintendo was left; they already had proven success with small indie titles with their e-shop. However, as the designer tweeted, Nintendo rejected The Binding of Isaac due to its »questionable religious content«. McMillen doesn't explain why out of all publishers Nintendo, with its main focus on family-friendly entertainment, ended up being his first choice, but one may assume it was all part of the plan to be able to say "Look, Isaac is too much for Nintendo!". On a side note: In Germany, The Binding of Isaac has a USK 16 rating – due to its »blasphemous message«.

An unusual sales history

McMillen takes several commercial options for Isaac into consideration. Due to his cautious expectations he is even prepared to sell the game to American cable network Adult Swim for as little as 40,000 dollars. He would be okay with that, but considering how successful Isaac turned out to be three years later, it would have been peanuts. Several of his friends try to talk the designer into putting The Binding of Isaac on Steam. Fortunately, Valve loves the little indie game. They recommend a sales price of ten dollars. Since he is still skeptical, McMillen lowers the price to five dollars. »Thank God for Steam« is what he tweets at the time.

At first, Edmund McMillen's expectations turn out to be true. Despite good reviews, The Binding of Isaac only generates moderate sales. The reviews always say pretty much

the same: a solid roguelike; maybe too hard for the non-hardcore gamers, but no mega hit like Super Meat Boy. Isaac stays mainly below the radar and strangely enough, the reviews hardly mention the offensive content. It seems nobody is upset about it. At least The Binding of Isaac yields over 40,000 dollars within the first month. McMillen is happy with the result, but a little disappointed nonetheless; nobody seems to take notice of the developer's offensive intention. Four months later there's a big surprise. Sales increase dramatically. McMillen has no explanation for this development. It doesn't make sense and is, according to him, »unprecedented in the gaming history«. But eventually, he finds out what caused sales to go up: YouTube.

The surprising formula for success

At peak times, about 200 Let's Plays of the game pop up per day. It's a catalyst for sales. While in the beginning the game averages 100 sold units a day, YouTube increases demand to 1,000. Within nine months, The Binding of Isaac exceeds the one-million mark and, to this day, sells more than three million copies. YouTubers love the experience that The Binding of Isaac has to offer. McMillen suspects it's a collective »internet thing« to uncover all of the game's secrets. Some fans like Northern-Lion and CobaltStreak even gain internet popularity thanks to Isaac.

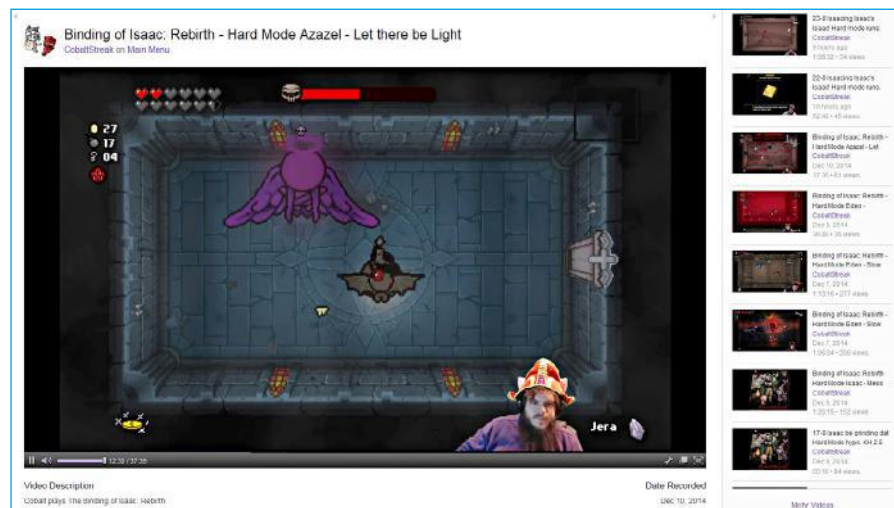
By now, CobaltStreak completely owns the Isaac audience on Twitch and owes his full-time job as a streamer to the indie game. Together with his colleague Richard Hammer, he even founded the tournament »Balls of Steel«: a race in which the world's best Isaac players compete against each other at the same time. Thanks to CobaltStreak and his competitors, The Binding of Isaac has now turned into a kind of e-sports, which is strange considering it's purely a single-player game. The Binding of Isaac offers ever new adventures and challenges. The rules and the structure of the roguelike motivate the fans to share and exchange their gaming experience. After all, there are 187 secrets in the game. Only those who find all of them receive the »Platinum God« achievement which only 2.9 percent of all Isaac players have reached so far.

McMillen's »fun little Flash game« has become more than a mere underground hit – against McMillen's estimate and plan. What should give him something to think about is that up to this day apart from Nintendo nobody has really taken considerable notice of or taken offence at the religious content. Obviously, there are numerous speculations: Are we too cynical? Is the look of the game too abstract? Are the mechanics undermining the message? Whatever the reason is in the end: The reward for Super Meat Boy turned into a success story of its own which – now in its third year – still entertains millions of players and will surely continue its success with the remake Rebirth. So much for career suicide, Mr. McMillen. You've got to try harder next time.

Benjamin Wojkuvka



Edmund McMillen's (late) cat Guppy found her way back into The Binding of Isaac: Rebirth.



Fans such as CobaltStreak turned into full-time streamers through The Binding of Isaac and made the single-player game a kind of e-sports.

TOOLS

You need great tools in order to build great games. Experts from the industry share their view on recent updates as well as new software solutions and reveal their favorite little helpers.

SCREENPRESSO WWW.SCREENPRESSO.COM



Teut Weidemann
Senior Online
Games Supervisor
at Blue Byte

As **supervisor** you not only evaluate internal titles during development, but also external ones. You continually need to add parts of screenshots to your analysis, label them, document them and attach symbols to them. The freeware Screenpresso makes this particularly easy: Via the »Print Screen« key the image is copied to the program clipboard and may immediately be equipped with notes, symbols and arrows and edited further. And not only that: Screenpresso also copies the image immediately to the clipboard, so this »copy & paste« technique speeds up the process considerably. The Pro version is available for 29 euros and also includes video capture, Windows transparency, PDF/HTML or Word format export.



Screenpresso lets you attach notes and symbols directly to screenshots.

SUBLIME TEXT 2 WWW.SUBLIMETEXT.COM



The Multi Edit Tool which can edit several lines simultaneously is particularly impressive.

Sublime is a **cross-platform text editor** with numerous functions and a flexible plug-in system for custom configuration. All functions are accessible via convenient keyboard commands, which allows you to keep both hands on the keyboard. In addition to macro functions, a build system and a project management feature, there are also pre-installed color schemes for more than 70 programming languages. Should the basic version be short of a function, Package Control – the package manager – grants access to over 2,700 available extensions (<https://packagecontrol.io>). Sublime is available for 70 dollars, but can be tested for free.



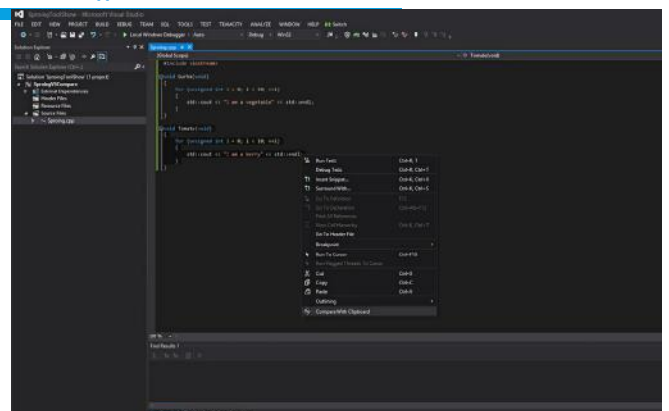
Hans Hamm
Head of R&D
at Deep Silver FISHLABS

VISUAL STUDIO COMPARISON TOOL [HTTP://COMPARISONTOOLS.CODEPLEX.COM](http://COMPARISONTOOLS.CODEPLEX.COM)



Bernhard Raml
Programmer
at Sproing Interactive Media

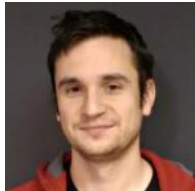
Those who work on complex projects like computer games know how important it is to have a clean source code. Among other things it is advisable to avoid repetitions in the code, and with the Visual Studio Comparison Tool it is now possible to compare code segments quickly and easily. For example, you may compare the code of two functions in order to spot common features more easily, which can then be combined to form a new function. This increases the code's reusability and facilitates the maintenance. The tool is open source and works with all common diff tools such as WinMerge, P4Merge and Beyond Compare 3.



Visual Studio Comparison allows uncomplicated comparison of code segments.

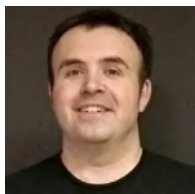
THE EVOLUTION OF TURN-BASED AI IN SHADOWRUN

To improve the AI of their game the team of Harebrained Schemes decided to create a new script language of their own instead of using an existing one. The result is an AI easy to use for anyone.



Sheridan Thirsk
is Software Engineer
at Harebrained Schemes.

Sheridan has programmed video game systems since 2008 with a wide range of genres including console sports, family friendly MMO, main stream first person shooter, and gritty turn based RPG. His primary focus is to create intelligent systems and work with others to design the most fun game experiences.



Kevin Maloney
is Game Designer
at Harebrained Schemes.

Kevin joined Harebrained Schemes as a Level Designer at the start of production of »Shadowrun Returns«. These days they call him just designer and he designs features, mechanics, helps out with narrative and audio, still makes levels and lately got into the more technical side of AI. Before joining HBS, Kevin worked on AAA console games but likes working more in a small dynamic shop like Harebrained Schemes with all its opportunities and challenges.

Our most recent release, »Shadowrun: Dragonfall – Director's Cut« (DF-DC), features an in-house AI system that we developed called »Gumbo«. In this article, we will detail how Gumbo came about, what our goals for the system were, and how, in the end, it largely accomplished them. We hope that our takeaways from this project will be helpful towards your own turn-based (or real-time) AI endeavors.

Background on Shadowrun

First, a little background. After a Kickstarter campaign that ended successfully in April of 2012 with 1.8 million dollars in funding, »Shadowrun Returns« shipped on July 25th, 2013. Seven months later we released a major DLC expansion entitled »Shadowrun: Dragonfall«, and on September 18th of 2014, we released »Shadowrun: Dragonfall – Director's Cut«.

The »Dragonfall«-DLC began as a stretch goal in our Kickstarter campaign, and was originally known simply as »Berlin«. By the time it went out the door, however, it had grown into a full expansion that improved on many aspects of the core game (RPS review).

Both the team and the community responded strongly to the improvements in »Dragonfall«. Still, even with those improvements, the team felt that a stronger version of »Dragonfall« was possible. Thus, we embarked

on creating »Dragonfall – Director's Cut«. Free to all owners of the DLC, DF-DC was not only an improved version of »Dragonfall«, it was also now standalone, so new fans could jump right in to our latest and greatest.

When we came up with a list of areas that we wanted to invest in for DF-DC, smarter AI was high up on the list (especially given that combat was also being overhauled). Both »Shadowrun Returns« and »Shadowrun: Dragonfall« used AI derived from a branching tree of xml edited in Unity. They also used custom code to read data and perform actions in the game. While this system succeeded in giving a variety of enemies basic behaviors, we felt that there was room for improvement on several fronts. With the existing system, any tuning or unique behaviors that designers wanted to make required engineering support, followed by the lengthy process of rebuilding our assets and code. This made iteration very slow and hid the AI's decision-making process from the designers.

Goals for improved AI

Below were our primary goals for the new AI technology:

Easy

Designers should be able to set up characters, items, and maps in line with their vision of what combat should look like. The AI should act as the final piece to realize their design goals. Each decision the AI agents make should be visible and editable by the designer if desired.

Safe

Recursion or failure in a turn based game can be quite bad. In our first game, we implemented »brain freeze timeouts« if we detected that the AI tree was stuck or looping. This was a practical solution, but not an ideal one. A new AI solution would need to make all possible problem areas fail gracefully without being obvious to the end user.

Expandable

The point where an AI has a basic set of actions and operations is just the first stage of development. Because of this, additional features in any potential area would need to be easy to plug into the AI. Common patterns would be condensed into a single feature to create consistent designs, reducing the potential for error.

Fast

While a turn-based game may seem slow, the AI's decision-making process needs to happen very quickly. When an AI agent arrives at a tile and decides to perform an attack, a pause between those two actions is detrimental because it occurs frequently. Similarly, all NPCs on the whole map get a slice of time to think even if they are »non-combatants«, because their combat status could change at any time. Our new AI would need to be fast to account for all of this.

Our Solution

We decided to create a lightweight new script language called »GumboScript«.

One member from our Kickstarter community asked why we didn't use LUA, or another script that is established and documented. While the extra capabilities that full-featured scripts like LUA provide have their advantages, they were not as valuable to us as writing our own language proved to be.

To reduce string parsing, our GumboScript is read once and stored in C# classes like the rest of our code. There is also a fixed small bank of memory with a few registers for a few data types which is allocated once in a block. These

data types are also restricted to a basic set which avoids problems with discrepancies. We also wanted to avoid script bindings, or connections, as much as possible. Finally, because we ship on multiple platforms, it was safer not to use external libraries or runtimes with limited memory access and file reads.

Core Objectives

Readable

Programmers have become accustomed to seeing brackets of all kinds [, { , (, < and words like »void« or putting an »f« or »ox« next to a number. However, this is not very natural, and is an easy point of failure for a new user writing a line of code. In GumboScript, we use some basic nouns and verbs, some algebra, and a little shorthand to cover the rest of the math operations.

In this case, a bitwise match is written as the word MATCH, and the list of attributes is separated with commas instead of a bitwise »or«. Nesting is done with whitespace. Verbs are always written before nouns, and adjectives and math operators combine to create formulas. We hope that it reads closer to an English sentence than to what some may call gobbledygook.

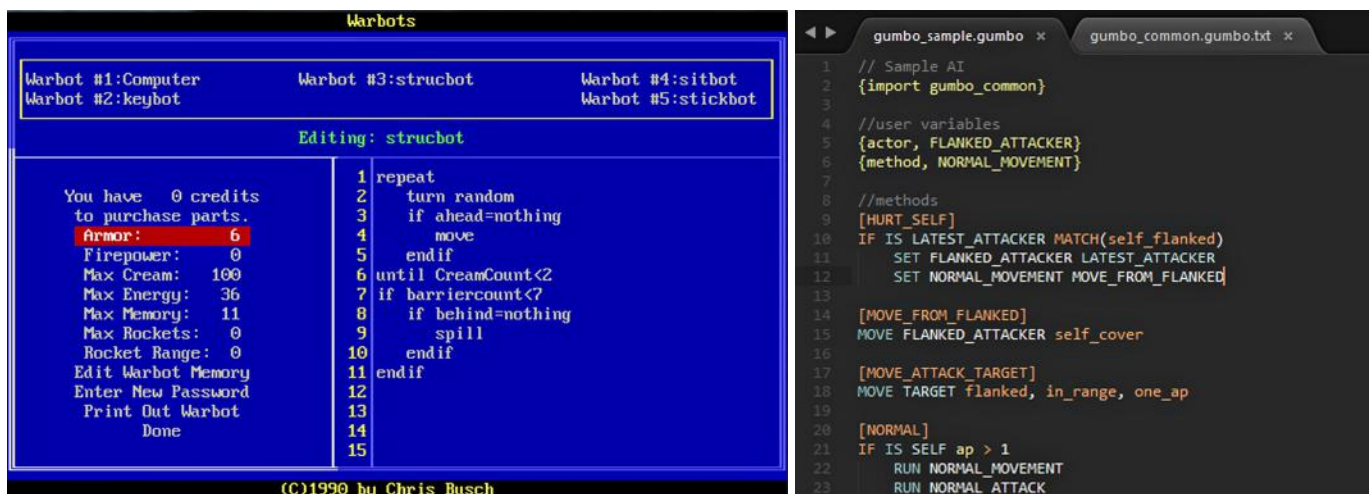
Our inspiration for a very approachable script comes from the modern scripts LUA and Python, but also from a 70 kilobyte sized game made in 1990, »Warbots« by Chris Busch. The objective of »Warbots« is to script a combat robot using a basic knowledge of logic, minimal formatting, and simple commands like move and shoot.

Failable

Sometimes failure is intended, and sometimes it is unexpected. In all cases, the AI agent should respond gracefully. In GumboScript, every action can pass, fail or become busy. Using the debug tools, we can see where the flow failed on an action, and the designer can choose to write fallback behavior at that point if desired. For example, if a gun fails to fire, it may fall back to a reload action or switch to a punch attack.



»Shadowrun: Dragonfall - Director's Cut« was originally planned to just be a stretch goal of a Kickstarter campaign but later grew into a full expansion.



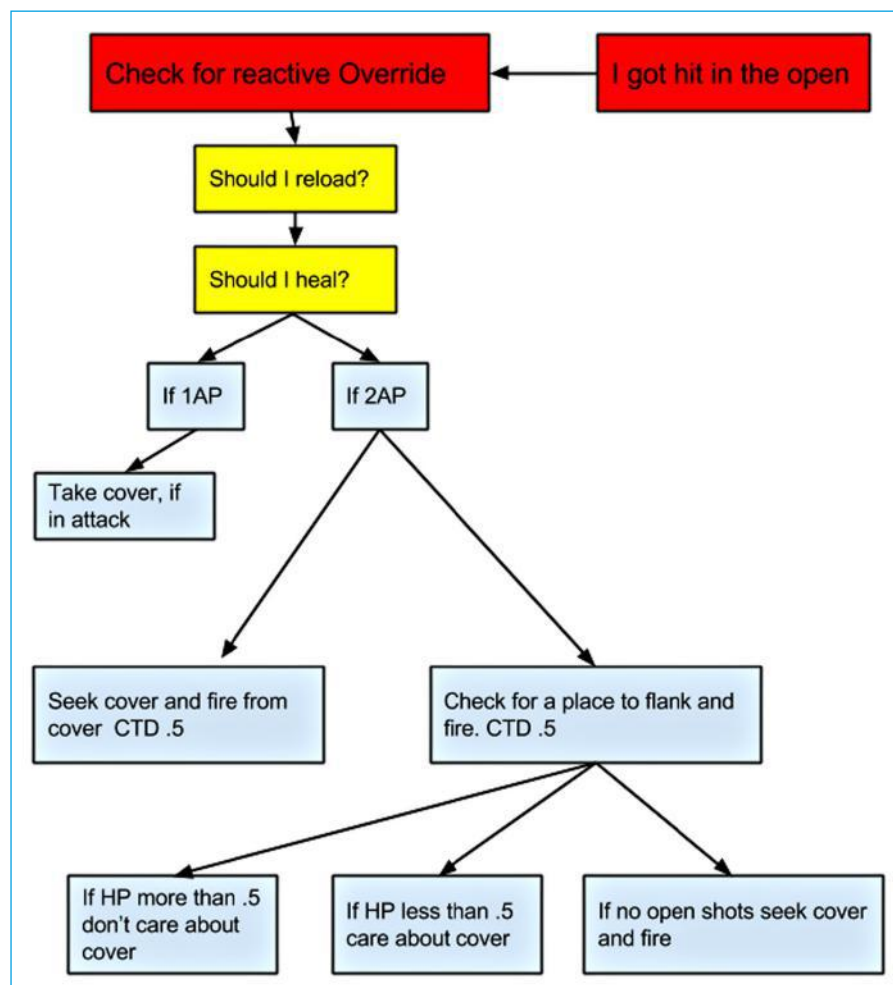
This figure shows a comparison between warbots pascal as used in the game »Warbots« (1990) and GumboScript. As you can see in both cases the system works with a few verbs and nouns to make it easier to code and to make sure that everybody in the team can work with it. In addition this approach makes sure that less errors occur while coding.

Perpetual

When an AI agent starts thinking, it should remember information about its own actions and scans, and sometimes its allies' information as well. We populate the AI agents' memory with a few common entries like the



The figure shows how the newly coded AI works within the game and how characters in the game behave.



This diagram was made to start creating the main combat loop for the Soldier which is the most common type of AI agent.

latest attacker or longest ranged weapon. Additionally, designers can define their own variables to be stored. One script may keep track of their most useful item or a specific enemy who flanked them. Later in the game, they can use these to factor in the decisions. This is not a new concept to AI, but it was an important addition to our script allowing the designer to decide which things are remembered.

Event driven

For real-time AI, there are many different events that can cause AI agents to rethink their actions. Turn-based AI can be limited to the turn start being the sole event, but we wanted some more control and timely reactions. Currently we can have the script respond to a handful of events including a friend being attacked, or a stranger walking into sight range. The designer can then start a sequence of events to fit their design instead of waiting until the next turn.

Restrictive

While we wanted the most flexible system possible, we also wanted speed and safety. This is why some operations are not possible in script. Currently, our methods have a default loop maximum of three loops. The distance and visibility to all threats is evaluated internally only when the positions have changed. External scans to the world and internal scans to one's own abilities or attributes are cached at turn start and only reevaluated when necessary. This allows the AI script to use complicated combinations of selections or evaluations multiple times without having to redo potentially slow work.

With these concepts established and the code in place, we set about designing profiles for our various enemy types. We established that our primary design goal for the AI was: »not look dumb«. Although that may sound glib, it's a pretty practical way to look at AI; more than anything, what's important is how the AI appears to the player.

On the left is the diagram we made to start creating the main combat loop (method) for our most common type of AI agent, the Soldier.

The loop evaluates on the start of every AP (Action Point, the currency used to take actions) spend. The reason for this is that AP is variable. Typical enemies start with two AP, but any number of things could be happening on the battlefield to deduct AP. Thus, there is no guarantee that the AI will still have AP to spend during evaluation.

We expanded on this model as we progressed, but the central principles of the combat loop remained until DF-DC shipped:

- **Reactive overrides (red)** are something that has happened to me not on my turn and that I must do something about, such as »I got hit in the open,« as shown in the diagram.
- **Special conditions (yellow)** are secondary priorities. They aren't matters of life and death, but they're still pretty important. These fire one after the other in a priority sequence. This allows the AI do things that

are really important, but not at the expense of risking their lives. Having a degree of self-preservation was key to our goal of »not dumb-looking«-AI.

- **In the standard combat part of the loop (blue)**, you will notice that cover is very important. This was high on our list of ways to make the AI agent appear »not dumb«. Standing in the open while a team of feisty Shadowrunners have various forms of hurt (both technological and arcane) pointed your way certainly qualifies as dumb.

We took advantage of the human tendency to see and infer a meaningful pattern where there isn't one. In the diagram, CTD stands for »Chance To Do«, and the decimal value is converted to a percent chance. In the diagram, the AI has 50-50 chance of either seeking cover and attacking, or looking for a flanking shot. For our Mage AI agent we used CTD in even more places, and we were happy with the result. For example, most of the time Mage AI will wait for enemies to be grouped to fire their Area of Effect, but every now and then they will fire on a single target just to keep it interesting.

The overrides, special cases in priority order, and somewhat random primary combat loops together create the impression of intelligence that appears to be doing sensible things.

Objectives vs. Execution

The goal when creating the profiles was to make a systemic type of logic that would »not look dumb« in all situations, but this was not how things worked out in practice. We found a major issue in the form of the AI taking cover twice and not taking an action. There were various flavors of this, from the sort-of-odd-looking to the really silly looking. For example, the AI might:

- Take cover twice while advancing
- Take cover twice while moving away from the player
- Leave cover and then take cover in the space that the player just vacated

In the name of both speed and safety, a brute force approach was taken in the script. Once cover is entered, it is more or less guaranteed that AI agents will either take an action or end their turns. Whenever the script could be at point where the AI wanted to take cover, it either did so, took an action, or ended its turn. The reason why we did this at every possible branch rather than handling it systemically was due the number of behavior permutations that could occur.

Given another shot with some more time, a more systemic approach to the cover behavior would be possible now that we know where the landmines are. We still think it was the right call to make something safe rather than something elegant as our shipping date approached. It's plenty exciting to bolt things onto a moving train, but it's a little bit too

thrilling to change how the engine works while it is running at full speed.

Fortunately, our cover issue was our only significant speed bump. Having our designers able to iterate on behavior by themselves was an order of magnitude faster than our previous system. If additional functionality was required in GumboScript, it could usually be done quickly, which allowed the design team the necessary time to construct the behaviors. The end result was a higher-quality AI experience with less engineering time. Even on the tight timeline of DF-DC, investing in tools paid off.

Conclusion

GumboScript for DF-DC is only the beginning; a first version designed to support basic turn-based actions. We intend to advance the system features for future Harebrained Schemes games by adding more prediction, goal-based decisions, and the ability to learn from past results. In game theory and economics, the term utility, or usefulness, describes the ability to satisfy an agent's needs and wants. Evaluating goals and sub-goals with some informed predictions will allow agents to do the thing that is most useful: utility maximization.

In our games, this is not just a single action but a sequence of choices. Because many actions require previous actions to be completed first, a sequence is strung together and evaluated as a whole. Turn-based games with a strong random element can easily break down when estimating utility value before the dice is rolled. Our challenges will be properly estimating utility and failing gracefully to a smart fallback if something is interrupted or a dice roll fails. Also, in a dynamic world, agents will prefer some goals based on the success rate or their designed behavior pattern. We hope that these systems can work in both turn-based and real-time environments for more responsive AI in all our future games. See you in the Shadows.

Sheridan Thirsk, Kevin Maloney

Additional information and links

For more information on games that do implement a goal-based system, we suggest Jeff Orkin's work in »Goal Oriented Action Planning«, which is a practical application of some advanced AI concepts implemented in the first person game »F.E.A.R.« (among others).

Link:

www.makinggames.biz/goap

To try out Gumbo script for yourself, you can create custom content in our online UGC workshop for »Shadowrun: Dragonfall – Director's Cut«. Internally we have used »Sublime Text« as the editor with our own syntax coloring. Further documentation and the sublime text package file are available on our wiki.

Link:

www.makinggames.biz/gumboscript



Since the release of »Shadowrun: Dragonfall – Director's Cut« the team worked on their new game »Shadowrun: Hong Kong« for which in January 2015 a Kickstarter campaign was started that reached its funding goal within a single day. The campaign ended on February, 14th.

SOURCES OF UNCERTAINTY

Games that are too predictable are getting boring for the player easily. Therefore there are a lot of ways to create uncertainty by specific game design choices, but not all of them help to make a good game at the same time.



Fabian Fischer is Master of Computer Science and prospective professional game designer.

Fabian considers games an unique form of art that deserves to be analyzed accordingly. As a passionate hobbyist, he has been writing critical articles focused on game design and ludology for several years. Alongside his personal blog, he also regularly writes for GamersGlobal and Gamasutra. His recent master's thesis was concerned with the analysis of modern game design principles in the context of digital game-based learning. He is now looking into opportunities for starting a creative career in the games industry.

Some time ago I thought about strategy games and why uncertainty is of central importance there, because without it there can be no decisions. In the broader picture though, uncertainty is actually necessary for any kind of gameplay to be interesting. In the case of absolute certainty when there's nothing more to discover, we usually just stop playing. Funnily enough, while we're always striving for certainty in real life, it's the death of any game. Therefore it's imperative for game designers, critics and analytics to understand the mechanical means by which uncertainty is created and maintained, but also where it can go wrong. The following article, among other sources based on Greg Costikyan's »Uncertainty in Games«, aims to shed some light on the topic.

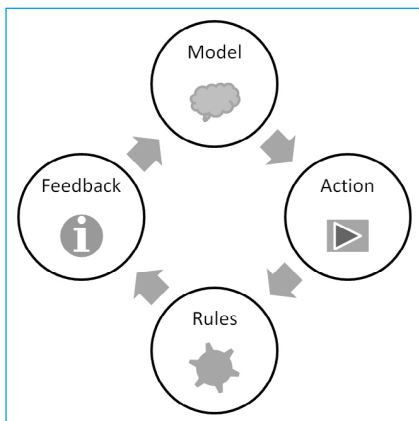
Defining Gameplay

The reason why uncertainty is so important can be understood by taking a closer look at the nature of any gameplay process. In essence it is all about learning, i.e. gaining competence. In this context Daniel Cook introduced the very useful conceptual idea of gameplay »arcs« and »loops« (see link 1). A gameplay arc consists of four elements: The player holds a mental model of the game as a collection of knowledge (you could call this his »skill«), he then takes actions while playing, which are then evaluated by the game's rules, resulting in feedback. This sequence already resembles scientific theory construction through hypothesis, experiment and observation. These similarities become even more evident when the arc is closed and becomes a gameplay loop: After receiving feedback, the mental model is adjusted accordingly and thus refined over many iterations – just like a scientific theory becomes more and more robust over time. Therefore, as for example Carlo Fabricatore (see link 2) concluded before, without learning there can be no gameplay.

While the arc structure for example fits linear puzzles or narrative games, loops are usually found in more skill-based and strategic games. In both cases though, the player is actively learning by interacting with the game system. Once the mental model is fully optimized, the end of any arc is already known beforehand and going through it again becomes a rather tedious affair. Similarly, if additional iterations of a gameplay loop do not result in further adjusting the model, they're gameplay-wise completely redundant and uninteresting. Uncertainty is therefore of great importance for any kind of game: If we already know about all the possibilities a sandbox game (or »toy«) has to offer, exploring it doesn't yield any gameplay value anymore. If we already know the solution to a puzzle, it's not exciting to solve it. If we have perfectly mastered a motoric discipline and reach the maximum amount of points every time, we quickly get bored. And if we figured out a complex system of decision-making so thoroughly that we can find the best action in any situation without a problem, there are effectively no decisions to make anymore. With the importance of the topic established, the following sections will describe some of the possible sources of uncertainty.

Complexity

A basic principle in trying to keep players from absolute certainty lies in creating a sufficiently complex system. If the space of possible game states is big enough, players can't possibly incorporate all of them in their thinking process in every situation. An important requirement for this to work properly is of course a solid internal balancing between all the available actions. There must not be large parts of the possibility space that are effectively useless. A classic example is Chess. The heavily branching decision tree makes it practically unsolvable for a human mind. In combination with the usually highly limited time banks in tourna-



Daniel Cook introduced the idea of gameplay arcs and loops. This figure shows how a loop works and how a mental model is changed by actions, rules and feedback.

ment settings, this forces players to develop and employ heuristics, more or less rough estimates of what could make a relatively good decision in any given situation. In other words, players are adding to their mental model of the game over time. Of course, Chess still has its problems in that regard: Especially the more restricted and predictable early-game has been partially solved. This forces seriously competitive players to memorize a huge number of openings, essentially taking away some of the importance of actual »on-the-fly« decision-making. And even later parts of the game are very prone to requiring rote calculations and inducing »analysis paralysis«. So, while it is an important factor, complexity alone is generally not enough (see link 3).

In contrast to strategy games, puzzles are meant to be solved. Nevertheless their typical form of uncertainty generation relies on complexity, too. The most interesting puzzles are those whose difficulty lies right on the edge of what the player is able to handle. Since this makes the optimal amount of complexity largely dependent on the individual player, many prominent examples like »Portal« or »Braid« make heavy use of the principle of incremental challenge. Initially the player faces relatively easy tasks that only demand manipulation of a few components and are usually manageable without too many problems. In a sense those are extended tutorials aiming to get players with less puzzle experience »up to speed« without them having to resort to frustrating and unsatisfying »trial and error« methods. All the while they are allowing more experienced players to breeze through before entering the later levels where all the previously introduced mechanisms have to be combined, in the end resulting in an optimal challenge (»flow«) for the maximum amount of players.

At last, even sandbox games are often based on complexity. If the possible interactions between all the existing mechanisms are rather obvious and quickly exhausted, not too many players will be motivated to invest a lot of time into the »toy«. Successful examples such as »Minecraft« on the other hand rely on an enormous breadth of possibilities. Not only are players able to move freely through giant worlds, basically allowed to build whatever structure they want, but the relations between all the individual components are also numerous and diverse. The interplay of blocks, tools, items, NPCs, animals, weapons, monsters and other elements makes for a network of enormous possibilities that beg for long-term experimentation.

Input Randomness

In addition »Minecraft« also employs randomness to generate a large variety of game states. The game world is randomly generated and even expands further and further dynamically as the player explores it. That way he can never be quite sure what to expect behind a forest, in a

desert or on a mountain top. This so-called input randomness (see link 4), happening before the player acts, is an effective way of consistently presenting the player with new kinds of situations and challenges. These days it experiences a revival in the course of the »roguelike renaissance« that made randomly generated content popular again, forcing players to explore edges of the system they might never see with a static setup. Actually though, this source of uncertainty is much older than video games altogether: Drawing from a randomly shuffled deck of cards, while much »closer« to the current game state than pre-game map generation, can just as well be interpreted as input randomness.

In strategy games this kind of random generation can also serve to combat solvability by making partial solutions much less useful. Again, this can be understood using Chess as an example. When the initial (symmetric)

Further reading

This article takes reference to works of other game designers or scientists (those are marked as »see link«). For further reading please go to our website where we you can find a collection of those links mentioned:

www.makinggames.biz/sources-of-uncertainty



Games like »Braid« make use of the principle of incremental challenge and therefore become quickly very hard to beat.



Roguelike games like »Brogue« are based on randomly generated content, forcing the player to explore the system fully.

setup of the pieces is determined randomly (as is the case in the variant Chess960), memorizing opening books becomes a much less valid method of increasing one's skill. Instead the importance of the players' systemic understanding, i.e. the accuracy of the developed mental models and heuristics, is emphasized. With this in mind, the designers behind the recent Kickstarter success »Prismata« made another good case for why this form of randomness is particularly useful for competitive strategy games.

Motoric Execution

Even if players already know what the best action is, they can still be challenged by actually executing their plan. In turn-based games this is usually not a factor. Merely moving the pieces on the chessboard is of course not a reason for the game's longevity. Especially video games tend to rely on real-time and direct control, though. In doing so they introduce a dexterity component into the system. A game like »Super Mario Bros.« creates its uncertainty, and therefore its interestingness, almost exclusively by how hard it is to master the controls. Conceptually it's in most cases pretty clear what has to be done to complete a level. However it's not obvious how exactly this can be accomplished. In which exact moment do the buttons have to be pressed? An even clearer example of this idea might be rhythm games such as »Guitar Hero«.

Interestingly those kinds of games can also be interpreted as puzzles: There is an optimal execution (solution) the player aims for. In rhythm games this results in a maximum amount of points, platformers on the other hand have developed the subculture of »speedrunning«. In contrast to classic (mental) puzzles that can only »recover« from being solved if the player actually forgets the solution, these »execution puzzles« tend to maintain their uncertainty for longer periods of time. Usually there always remains some variance as to how perfect the motoric execution can really be, which in turn supports the contest qualities of these systems. Most players simply won't be able to perform perfectly every single time (and if they were, the validity of the contest as a measurement of skill has to be questioned). For example, in a shooter game not every single attempt to headshot an enemy will actually hit. And even a professional »League of Legends« player will, from time to time, miss a »skillshot«.

Hidden Information

Another, somewhat more immediate, way of inserting uncertainty into a game system lies in explicitly hiding information. If it is not even possible for a player to know the full game state, there will usually not be complete certainty as to what the best possible action might be in any given case. A classic example is the »fog of war« in real-time strategy games that hides the opponent's actions but can be com-

bated by deliberate scouting. An even simpler example is Poker. The hands of the opponents are unknown and only probabilities for all the possible results of a round can be determined by any individual player. Optimal play is still possible in this case by performing the actions promising the highest chance of success, but even then there's still the danger of losing if a more unlikely random event occurs after all.

Collectible card games such as »Hearthstone« carry this concept to extremes since not only the cards the opponent is holding are hidden but also his complete deck. While the separation of possible decks by hero class (in the form of class-specific cards) provides a little remedy in this case, many tournament organizers still think of this as overdone hidden information, basically making it impossible for the game to reliably test the players' skill level. Therefore special tournament rules are sometimes applied, as was the case in the »Prismata Cup«, forcing players to reveal their deck lists before playing. This does not only lead to increased, yet far from complete, predictability, but can also result in some strategically very interesting matches coming down the last cards in both players' decks.

Multiplayer

Even if »Hearthstone« were to be played with completely open (and deterministic) cards though, it wouldn't be completely obvious what would happen in every situation. In the end there's always the uncertainty factor of the opponent's mind in multiplayer games, although it can be predicted quite reliably. After all it must be assumed that the adversary will, in any given situation, try to do what's best for him. However, his exact evaluation of the game state is usually unknown and therefore could simply be seen as another form of hidden information. Depending on the accuracy of the involved players' mental models, what they believe to be the best action (for themselves and also their opponent) might differ dramatically. Last but not least creativity, coming up with bold and surprising lines of play, takes on an important role in this context, too.

On the other hand, games in which the opponent's action evaluation cannot or just barely be derived from the current game state itself, tend to come down to mere luck. That's for example regularly the case when the mechanism of simultaneous actions is involved. A classic example is Rock, Paper, Scissors whose counter-mechanism is used or adapted completely in many other games such as the card battler »Yomi«. Apart from the myth of »mind reading«, relying on this mechanism alone results in a game of, more or less, pure guessing. In this case there's so much uncertainty involved in the game that it can barely be of any interestingness for the human mind. If there's no order to be found inside the chaos to begin with, there can be no learning and therefore no



Uncertainty often is created by the question of how to master the controls of a game. In »Super Mario Bros.« for example it is almost impossible to predict the moment to press a button exactly.

actual gameplay. At this point it's worth noting that some specific methods of creating uncertainty that were established over time actually appear quite suspect upon closer inspection. In many cases they do not only hurt the overall system, but also the player himself by making very inefficient use of his time. In this regard some warning notices can be found below.

Fake Complexity

Many games, at first sight, seem much more complex and interesting than they actually are. This pseudo-complexity can be created by several means, one of which is the concept of »quantity before quality« or »mass content«. A system bloated with loads of different components can seem quite impressive initially. »Magic: The Gathering« with its more than ten thousand cards or »League of Legends« and its more than one hundred playable characters are successful representatives of this philosophy. Mostly this concept tends to be primarily fueled by an according business model underlying the game though, and not the intention of designing a high-quality game. Ultimately these systems crumble under their own weight: Only a fraction of cards and characters are actually valid options. The rest is either buried in oblivion or alternating between »too strong« and »useless« over the course of seemingly infinite patch orgies. Huge parts of the game become meaningless noise and the barrier of entry is unnecessarily high. Camouflaged as the »metagame« this problem is then euphemistically regarded as added depth. Ultimately though, players merely have to internalize the new list of »overpowered« gameplay elements after every new balancing patch. In the end these systems simply can't be balanced because they hold more content than they are able to bear mechanically.

Another way of making a game seem more interesting than it is includes lots of busywork. Forcing the player to perform a few chores on the side while fulfilling a completely trivial task can convey the impression of actually doing something worthwhile. It induces the comforting feeling of putting in effort to accomplish something. And indeed a lot of time in many action adventures is spent running down corridors or streets. The trick lies in presenting trivial tasks as difficult. Other examples include grinding that can dramatically prolong games (»Final Fantasy«), spectacular animations accompanying trivial gameplay challenges (»God of War«) and mere mathematical complexity obfuscating rather shallow mechanisms (»World of Warcraft«).

Unfair Randomness

The above-mentioned randomness can, in spite of its great potential, have severe negative consequences when used the wrong way. Even input randomness is problematic if it reduces the competitive consistency of a game. In

single-player games this can be observed when the possibility range of the random generation is too loose, i.e. it goes so far that multiple playthroughs are not comparable anymore. In »Spelunky« this for example leads to serious players restarting over and over again until they finally (and randomly) get a favorable setup. A similar phenomenon exists in the »Civilization«-games. In multiplayer games such as »Hearthstone« even the initial luck of the draw can decide a whole match. In any case, this leads to a very inefficient use of the players' time. It should always be a design goal to keep the input randomness as varied as pos-



If information are hidden to the player this fact can help to motivate the player to go on. A good example for this is »Hearthstone« where a player can not predict what kind of cards an opponent might have at his disposal.



In the mobile game »Yomi« so much uncertainty is involved that the game can barely be interesting for the human mind in any way.

sible while at the same maintaining a reasonable level of fairness and comparability.

Output randomness on the other hand, happening after the player decided to take an action, will in general lead to unfair results. It might »even out« over many matches but the immediate result of a round will still be severely affected by it, which results in compromised or at least problematic feedback. This form of randomness directly interferes with the gameplay loop mentioned previously: The feedback is distorted since the system behaves inconsistently, thus the learning process (i.e. the gameplay) suffers.

Unfair Hidden Information

Hidden information can come in unfair forms as well. A very clear example was given by Jonathan Blow in his »Design Reboot« talk (see link 5). He compared the top-down shooter »Smash TV« to its spiritual successor »Total Carnage«. The former displays a room without showing the bottom entrance. The player is therefore never quite sure of whether there are enemies coming in from that side. In this case hidden information is used reasonably, to

increase excitement and maintain the interestingness of the system. Without it the optimal behavior in most situations would be rather obvious. The situation is never unfair because the player always has enough time to respond once the information is revealed. »Total Carnage« on the other hand uses hidden information in almost sadistic ways. Mines are placed behind trees, completely invisible to the player so that he has to fail on his first try without standing a chance. It's after all imperative to look for items behind every tree in the game. It's all about memorizing the positions of the mines in the end. Simply letting the player fail without any warning is frustratingly bad design. However it still exists even in some modern titles, for example in the form of enemies attacking out of nowhere in action games.

A related idea that's even more common in today's games is the philosophy of »wiki design«. In its case the player's lack of knowledge is not punished quite as harshly as with the mines in »Total Carnage«, but in a more subtle manner. In games like »The Binding of Isaac« and many rogue likes the player is at a severe disadvantage, simply because he isn't told all the rules. If the game doesn't even explain what the different elements do, then there's of course also uncertainty as to what the player should do. As an analogy, think of a match of Chess wherein one player loses because he didn't even know how the queen could move. This method of hiding rules is after all a pretty absurd way of going about the issue. And if it's really necessary for a game to not break down immediately, how interesting can the system actually be? If it however isn't necessary, you might as well not prevent players from realizing how interesting your game is in the first place. Isaac's designer Edmund McMillen recently had to somewhat admit that himself (see link 6) and added (if only vague) descriptions of all the items to the remake of his game. Most players circumvented the design weakness anyways by constantly consulting one of the many fan-made wikis.



Fake complexity can make games seem more interesting than they are - a good example for this is the grinding in »Final Fantasy 13«.



Output randomness happening after the player decided to take an action like in »XCOM: Enemy Unknown«, will in general lead to unfair results.

Story Choices

Finally a few words regarding narrative decisions in games, as they for example exist in »FTL« or »Out There«. Their immediate problem is the obscurity they inherently introduce into the game. Game and story are two fundamentally different bearers of information. If a story decision now affects the underlying game state, this can never happen in a clear and coherent way. Again, let's look at a Chess variant to illustrate the problem. Imagine there's a story choice every turn: »The queen doesn't feel too good today, should she go for a walk outside?« Depending on the player's pick, the queen will more or less randomly move around the board. This example seems absurd, but that's essentially what happens in the above-mentioned games or classic »game books« such as »Lone Wolf«. A possible solution to this conflict lies in

minimizing the effects the story choices have on the actual game (see »Bioshock«). In this case however, it's questionable as to why the player should even care about these choices in the first place. After all the game teaches him that they're irrelevant. The other possible route the design could take involves eliminating any form of strategy, resource management and game-play altogether. One example of this approach is »The Walking Dead« by Telltale Games, who could actually have gone way further. In many cases the characters play dumb causing huge consistency problems in the plot, just to leave a bit of trivial puzzle work to the player so that it can be called an »adventure game« after all.

But even on a lower level, even when the gameplay has already been mostly eliminated, there's still quite some potential for problems in story decisions: Which point of view is the player actually supposed to take? Does he decide as an author trying to create a consistent and exciting story (»I pick X, because it fits the previous events and the general attitude of this character!«)? Or should one simply act based on personal preference and for example save the attractive lady or guy? Of course these conflicts can be reduced further, for example by not even putting an actual protagonist character in front of the player that could play a role in terms of story consistency, but instead a »blank sheet« the player can then fill with his own personality. However there are still problems with this approach in single-player games. Only a very limited amount of decisions and according reactions can actually be pre-planned. Especially the combinatorial state explosion in widely branching decision trees will necessarily call for somewhat preset characters with some assumptions made at the very least. As long as the transition to truly free collaborative storytelling (as it exists in some pen-and-paper role-playing games) is not made, there will always be potential problems.

Closing Remarks

In conclusion, taking a closer look at uncertainty in games reveals one thing: As important as it is for any kind of gameplay, as complex is the task of smoothly generating and maintaining it. Most modern videogames especially shy away from focusing on systemic complexity, because while it might potentially account for the most intellectual value, it is also much less easily perceivable than other factors. It's comparatively easy to make a game seem interesting on the outset. Audiovisual spectacle and mass content can be advertised with impressive trailers and large numbers, and are thus well-tested means of generating attention. Another cheap trick is drowning a system in chaos and unfair situations so that there's indeed always uncertainty involved when playing, but there's at the same time not much actual gameplay value left. Creating a deep system whose loops and arcs stay interesting long-term and that stands the



In games like »The Binding of Isaac« and many roguelikes the player is at a severe disadvantage, simply because he isn't told all the rules.



In games with story choices like »The Walking Dead« the characters playing dumb causes huge consistency problems in the plot.

test of time without multiple content expansions a year requires a fundamentally different approach, though. It's a challenge only few modern designers have yet taken on.

However, from a historical perspective this is completely understandable. The academization of game design is still in its infancy, especially in comparison to the immense underlying industry. It has only been a few decades that true specialists of the craft have even existed. Only just for about ten years have there been regular scientific and literary publications on the topic. There's a lot of progress made on a theoretical level, but it has only been able to make its way into everyday design practice rather slowly. Again, that's due to the rigid conventions of an industry that financially grew faster than it could artistically handle. High quality art demands a critical examination of the corresponding craft and especially full-time artists will begin asking serious and uncomfortable questions about the nature of what they're actually doing. Slowly but surely we're reaching this point in game design. That's why we can assume to see better games in the next ten years than any human being has ever played before. And that's really something to look forward to!

Fabian Fischer

THE DEVELOPMENT OF SCHEIN

When a student project that was needed to get good grades at University becomes a full scale game that can either be the big breakthrough for a young team or a hard journey into an uncertain future - in case of Zeppelin Studio and its puzzle game Schein it was both at the same time.



Tiare Feuchtnner
is Marketing & PR Manager
at Zeppelin Studio.

Tiare joined Zeppelin Studio in 2012 and has since been juggling what feels like a full-time job in marketing for »Schein« and her Master studies in computer science at the TU Berlin.



Michael Benda
is CEO / Co-founder
and Project Manager
at Zeppelin Studio.

Michael began working on »Schein« in the course of his Master in Game Engineering & Simulation. He is project manager, developer, level designer, animator and self-designated maker of coffee.

In mid-2011 the Game Engineering and Simulation students at the FH Technikum Wien, Austria were given the wonderful task to conceive and implement a game concept, without any bounds or constraints. The lecturers, all veterans of the games industry, emphasized the uniqueness of this opportunity: This might well be the last time that the students could realize their own ideas in a game - in the future there will always be someone else deciding what needs to be done and how to do it.

How it Began

Motivated by the great opportunity in face of this bleak outlook, Michael Benda paired up with Kilian Reisenegger and during brainstorming it soon became clear, that they both envisioned a game, which first of all they themselves would enjoy playing: A game full of contrast that challenges both the player's dexterity and mind in the same way.

When at a later stage workgroups were merged, Philipp Schäfer and Sabine Pölzlbauer joined the team. Together they thought up a simple mechanic using the light radius of a radiating source to illuminate and transform the surroundings, bringing forth new elements from previously invisible worlds. The whole game mechanic was intentionally kept simple. The interactions were designed to be easily

grasped by the player, while allowing the team to flexibly scale the difficulty throughout the levels of the game. Furthermore the concept met another self-imposed condition: The game did not pose any major difficulties in terms of technical challenges. Since this was the first proper game they had ever built, the team knew they would need to face a multitude of non-technical problems. Therefore the concept only included the most basic artificial intelligence, no physical simulation apart from gravity, and most certainly no multiplayer mode. Run, jump and think outside the box - those were the game's most important goals.

Usually such groups of game developers with a purely technical background are able to create the most elaborate game in terms of programmatic finesse, but unfortunately they mostly lack skills and resources regarding graphical art. Luckily a friend, Philipp Schürz, happily stepped in to fill the position of the team's artist and brought color and life into the game. So the team of 5 was complete, and as the game mechanics were refined, they also found a beautiful story to go with it. It was inspired by the myth of the will-o'-the-wisp (»Irrlicht« in German) and went nicely with the concept and the light radius mechanic: A young man in search of his lost son wanders through a perilous swamp, where he encounters a ghost light named



The puzzles in »Schein« are based on switching between different light settings that reveal certain objects in the game world that are only visible in a certain color – this makes for some challenging levels that start out to be easy but quickly get hard to beat

»Irrlicht« that lends him its help and power to continue his journey. The wisp reveals hidden dimensions within its glow. Thus, the important role of light and the magical power that shows unseen worlds led to the game's name »Schein«, which in German can mean both »shine« or »illusion«.

As was required by the curriculum, all the decisions and ideas regarding the game found their way into the game design document (GDD). This file was practically the whole game in written form: From its target audience to its in-game menus, all the details were covered in one single document. The GDD served as a reliable guide and good companion throughout all future decisions and changes. It was kept up to date and made sure that the team pulled together and headed in the same direction. After the game was completely defined on paper it was time to implement something playable and, especially, testable. Since the team had the longest experience with this engine, the prototype of »Schein« was programmed in Unity. And so, within a couple of days, the first playable version of »Schein« was born.

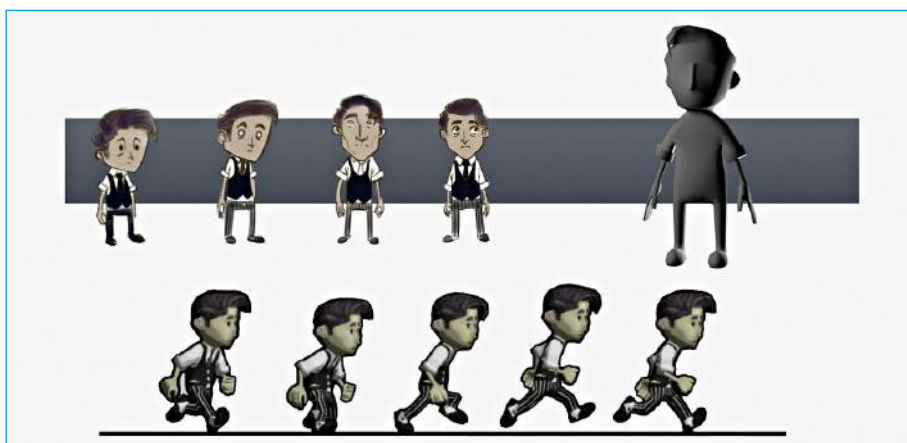
The reaction of both colleagues and supervisors was great, the project was fun and the grades were good too, so the team decided to push on further. Against the advice of the

supervisors, Philipp and Michael set out to develop their own engine, because they felt that a game with the unique setting and atmosphere of »Schein« was just not meant to be made in Unity. After all they were students, they were curious and they had the time, so what could possibly go wrong? Over the course of the summer holidays the basic game framework was finished. By the end of their Master studies in early 2012, after innumerable night-shifts and a good amount of feedback by their supervisors, the »Schein«-team had finally implemented a fully functional game with three tricky levels in their very own engine.

And it all paid off! The FH Technikum Wien awarded them not only with their Master certificates, but also with the Award of Excellence for the skillfully executed »Schein« project.

Publicly Approved

Many other student projects stop at this point, for reasons that range from simple lack of confidence up to lack of time and money. However, urged on by friends and former supervisors, »Schein« moved on and was presented to the public for the very first time at Gamescom 2012 in Cologne, followed by the »Subotron Live Pitch«-event in Vienna. These occasions were a great opportunity to



After the team had released a public Dev-Demo of »Schein« the feedback of the players helped to redesign the look of the protagonist, who then »grew up« to better suit the dark and mature atmosphere and setting of the game.

The series at a glance

- **Part 1**
The development of »Schein«
- **Part 2**
Creating a proprietary engine as an Indie



Beside of the game's hero the graphics in general were overhauled to remove the inconsistency that the assets from the beginning of the development showed compared to the newer ones.

pitch the game and receive impartial opinions on the concept and its execution from game industry veterans. They also led to the first contact with the national press and even the well-established local games industry conveyed their interest by offering valuable tips and feedback. Excited about the impact of their little student project on the public and desiring to gather even more opinions, the team published a Dev-Demo at the end of 2012, which allowed anybody to download and test one specially designed level of the game at the current state of development.

The accumulated constructive criticism generally agreed on two major points: Firstly, the game was fun – the puzzles were tricky and the unique game mechanics made the players think out of the box. Secondly, however, the art style was inconsistent – as the game development progressed, so did the art, but this led to a huge divergence between the first assets created for the game and those that were added later on. This led to the drastic decision of putting »Schein« through a severe redesign phase that took the team the whole winter of 2012/2013: the graphics were completely reworked and the main protagonist grew up a little to better match the darkness and maturity of his environment. Although this meant a severe postponement of the game's completion, the changes were vital to the success of »Schein«. If the team had kept the game hidden from public and not received this criticism, the game might never have become more than just another student project.

By then the decision had long been made, that the development on »Schein« would continue until the game could be independently released. But now it was time to set this in stone: Michael Benda and Philipp Schäfer founded the independent game development company Zeppelin Studio in early 2013 and Tiare Feuchtnner joined the team to help with marketing and PR. Most team members worked part time on »Schein«, while earning their living with more conventional jobs. And because no game could be complete without sound, friends at leed:audio, a sound studio in Germany, lent their support by composing ethereal soundtracks and doing the English voice-over for the dialogs between the »Irrlicht« and the young protagonist.

Two attempts were made simultaneously to engage the community and directly involve them in the development process of »Schein«: Crowdfunding and Steam Greenlight. The private financial reserves of the team were running low and to be able to finish the project they needed to accrue the necessary funds. One highly popular approach at that time was crowdfunding. Since the most popular funding platform Kickstarter was only accepting projects from the US and UK, they had to fall back on the more open but less known crowdfunding portal Indiegogo. Simultaneously, »Schein«



Most of the team behind »Schein« was working two part-time jobs, but still funding of the development was needed. After a disappointing campaign at Indiegogo winning a prize from Microsoft that was awarded with \$50,000 helped the team a lot to go on.

was introduced on Steam Greenlight for the first time. Unfortunately both campaigns proceeded unsatisfactorily. The progress on Greenlight was incredibly slow and seemed to stagnate at some point, and the crowdfunding did not accumulate the desired amount of money that was needed to continue work. In hindsight, however, both actions helped »Schein« greatly in two ways. On the one hand they provided a massive boost to community building and raised worldwide awareness through press coverage. On the other hand the team learned important lessons in regards to the online presence of »Schein« and its presentation, which did not properly convey the core aesthetics of the game. But this was about to change.

With the fully redesigned game and an improved pitch a major breakthrough was reached when »Schein« won in the Games category of the Microsoft Imagine Cup 2013. The exhilarating trip to the Imagine Cup finals in Russia was an amazing experience and brought constructive feedback about both the game and the business plan by renowned personalities such as Alexej Pajitnov (the maker of »Tetris«!). Apart from the long sought-after recognition from the international press, this prize also provided Zeppelin Studio with 50,000 dollars – funds that would make completion of the game possible. »Schein« had a future!

The Final Polish

When the team had recovered from their delirium of joy about the Imagine Cup, a phase of somber reflection, restructuring, and redesign followed. While Kilian and Sabine left the young startup to pursue other careers, Bernhard Klemenjak came aboard as second part-time engine programmer, making Zeppelin Studio once again a 5-person team. With heads completely filled with feedback and improvement ideas from the Imagine Cup, the Greenlight campaign was paused and the lacking online presence of »Schein« was finally overhauled.

Throughout the project several release dates had been scheduled and they all came and went with the game still being unfinished. So once again a new release date was fixed and Michael drew up a final project plan. But this time, to avoid the embarrassment of having to



Zeppelin Studio was always more or less a team of five, but members changed a bit since the start of the development. This picture shows the team who finished the game, from left to right: Philipp Schäfer, Philip »Flip« Kasper, Michael Benda, Tiare Feuchtner and Bernhard Klemenjak.

publicly announce any further delay, the date remained undisclosed and the team learned that it easily suffices to announce a release one month in advance. Meanwhile »Schein« took part in many other competitions, collected a couple of prizes, and the press wrote beautiful reviews that kept the spirits high.

In the following months, the voice-over was completed by leed:audio and the soundtracks were refined, the graphic engine was tweaked by Philipp Schäfer, Bernie rebuilt the in-game menu and Michael designed the final levels and created animated sequences for the cut-scenes, while Tiare attempted to keep the community and press entertained. When Philipp Schürz left the company to continue his education at university, it seemed as though the graphical development of »Schein« had reached a sudden halt. Fortunately though Philip Kasper also known as »Flip« came along and as the company's new artist he gave »Schein« a whole new level of colorfulness and detail.

The Release

In 2014, after three years of development both part-time at university and later almost full-time, »Schein« was finally complete. Or at least the team learned to accept the code freeze deadline and prepared for release, since like any other piece of art, games are never truly »finished«.

The Greenlight campaign was relaunched and on July 14th, 2014 »Schein« was released on multiple online distribution platforms – including GOG, Desura and the Humble Store. Thanks to all the positive press, the awards and the new online presentation of »Schein«, the game was greenlit after only 74 days having reached only 64% »of the way to the top 100«. This shows that luckily Steam does not only rely on the votes of players (who can be so easily manipulated and »bought« by rewarding votes with a free copy of the game). We love the idea of giving players the power to decide which games get on Steam and which don't, but if not all participants are prepared to play fair, it cannot work without some higher intervention.

So after creating achievements and embedding the Steam API, »Schein« was released on Steam on October 10th, 2014, marking the final step in development.

Both release dates, the one for all the other online portals and the second one for Steam, were accompanied by a wave of press releases to hundreds of national and international online news portals. Dozens even hit the mark and »Schein« received articles and reviews on popular sites such as RPS, 4players.de and IGN. The social networks of »Schein« were buzzing with news, giveaways and inquiries. All the online marketing effort, however, did not result in the desired sales boost. One month after the first release date only 264 sales were recorded across all online stores (excluding Steam). One month after the second release date on Steam »Schein« had reached 922 sales in total, 406 of these on Steam. This showed that publishing the game on Steam also affected the number of sales across the other stores, which confirmed the team's assumption that many players tend to buy games that can be activated on Steam.

In general these sale numbers were far below the team's expectations. Most of the reviews speak in favor of the game and all the awards and nominations give credit to its inventiveness, however this was not reflected in the sales. Despite the effort the team made in terms of marketing, community building and press, the visibility was still too low, seen in context with the vast amount of games that are being released at the moment.

What's Next?

The initial goal was to fund the next project mainly through the sales of »Schein«. Since this plan did not work out properly, the team must find other ways to secure their future development. Currently there are ongoing negotiations with publishers and console manufacturers concerning a port of »Schein« to consoles. And in the meantime contract work ensures the company's survival while new game ideas are designed, tested, prototyped and pitches are being prepared.

For all who are intrigued by this tale of a student project that became an award winning, published game, don't miss the next edition! You may gain interesting insights into techniques of level creation that are put to use at Zeppelin Studio and learn about all the right and wrong reasons to build an own engine.

Tiare Feuchtner, Michael Benda



After several delays the game was finally released in July 2014 and followed on Steam in October. Sales numbers did not turn out as the team had expected them to be and Zeppelin Studio is therefore negotiating a console port of the game.



The first prototype of »Schein« was created in Unity since the team knew this engine best. Later on the young developers decided that Unity does not fit their needs and their game.



As the first prototypes of »Schein« led to good feedback the team decided to go on with the development but to use an engine of their own instead of Unity, because it would fit the unique setting better.

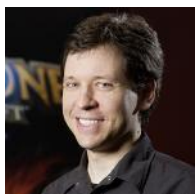
BLIZZARD'S SECRET FOR SUCCESS

The card game »Hearthstone« started as some kind of experiment with just a very small team of developers but quickly became a hit that millions of people play. In this interview Blizzard shares insights about the development and the secret why this game is that popular.



Jason Chayes
Production Director
at Blizzard Entertainment

Before Jason joined Blizzard Entertainment in 2008, he worked for The Walt Disney Company and Electronic Arts as a 3D Artist. After switching to Blizzard, he took on the role of Lead Producer for Battle.net and helped developing »Starcraft 2« and »Diablo 3«. Today, he is heading the Hearthstone team as Production Director.



Mike Donais
Senior Game Designer
at Blizzard Entertainment

As Senior Game Designer of Team »Hearthstone«, Mike is responsible for the card design and the related game balance as well as for the design of adventures like »Curse of Naxxramas«. Mike joined Blizzard Entertainment in 2013 after working as Senior Designer for »Marvel Heroes« at Gazillion Entertainment. He also was Design Director for the »World of Warcraft Trading Card Game« at Cryptozoic Entertainment. For ten years, Mike was involved in the development of »Magic: The Gathering« and »Dungeons & Dragons« as Senior Game Designer, to name just two examples.

Making Games According to reports, »Hearthstone« had 20 million active players in September 2014. Did you ever think the game would be this successful, or were you just as surprised?

Jason Chayes We generally try not to have any particular expectations on how big the game will be or could be. Our focus is really more on creating the game that we're all excited about, the one that we always wanted to play. We usually get pretty good feedback through the beta; whether people love our projects as much as we do. This system works really well and helps getting a sense about whether this could be a big game, but overall we never know just how big a hit with the players it will be in the end. We are certainly thrilled with how many people out there play »Hearthstone« today.

Making Games What do you think is it that makes »Hearthstone« so popular, and where do you think the fascination for card games like this comes from?

Jason Chayes Everybody in Team 5 – that's the group that develops »Hearthstone« – has been a fan of card games for a long time. Some of us have been playing them for more than 20 years, simply because we feel that there are some awesome sides to this particular genre, even though some parts are almost too deep. Every time we talked to friends about it, we got the feedback that although they liked

the games overall, they thought a lot of them weren't suited for the mass market. It was a big challenge for us to find a way to take this fun genre that has been inaccessible and try to turn it into something that anybody could just pick it up and play. And that's what »Hearthstone« was all about: We wanted to take our passion and create a game that anybody could learn within five minutes. I think that's been a big part of the success of »Hearthstone«.

Making Games How did you make sure that literally everybody was able to quickly understand the principle of the game? Were there any game elements that needed to be changed during development in order to guarantee easy access to the game?

Mike Donais One of the things other card games struggled with was waiting for your opponent to respond during your turn. That interaction was really slowing down the game, making it less fun. We took this out for our game and came up with a system where you take your whole turn and when you're done, it's your opponent's turn. That way it makes it really easy to play and offers a very relaxing and smooth gameplay. That's only one of the many things we tackled. All the other streamlining during gameplay like not having to declare your blockers has really made it faster and easier to play. The other thing is that

»Hearthstone« has a very good sound and animation team, so every card in the game has its own sound and animation for when you attack or when you die. That alone makes »Hearthstone« a fun experience to just watch and see what happens during a turn.

Jason Chayes That's right, and also the cards themselves play an important role. We invested a lot of time making sure that they are easy to read and it's obvious what they do. The player can just look at them very quickly to get an understanding of what's going to happen. So it's this plus all the things Mike mentioned that make »Hearthstone« a very accessible game.

Making Games Don't you think the fact that Blizzard was the developer and the »Warcraft« universe the basis for »Hearthstone« was the actual key to success?

Jason Chayes The »Warcraft« universe is definitely an important aspect of what makes »Hearthstone«. For people who are used to playing »World of Warcraft« or who love the »Warcraft« strategy games, there's been a lot of déjà vu in terms of the cards in »Hearthstone«. But apart from that gamer group, we got a lot of people for whom »Hearthstone« was their very first Blizzard game; who haven't been playing »World of Warcraft« or any of our other games. The fact that »Hearthstone« appeals to them and reaches players who have come to Blizzard for the first time, is a huge part of the success of how »Hearthstone« actually got bigger than it might have been otherwise.

Making Games So that means you're trying to use »Hearthstone« primarily to attract players who haven't been exposed to Blizzard or your games yet?

Jason Chayes Absolutely. And as I said we think that has happened already. Once people get introduced to »Hearthstone«, they are often really interested to try out our other games that are related to »Warcraft«. For instance, they may give »World of Warcraft« a try and like the fact that there are the same spells, character classes or minions that they already know from »Hearthstone«. So we definitely think »Hearthstone« could be an entry into the world of Blizzard for some people.

Making Games Do you think it's beneficial for the development of a card game like »Hearthstone« to completely rule out trading cards amongst players?

Jason Chayes That was indeed a very conscious decision for us. We discussed it a lot in the beginning whether we should allow the players to trade cards or not. In the end there were a lot of reasons why we decided not to include that feature. Since most of us already had a lot of experience with trading card games in the past, we knew it would be hard especially for new players to know the value of a card that they just got out of the card pack. In all these

card games, there are different levels of rarity – there are common cards, epic cards, legendary cards – and a new player simply can't know the exact value of a particular card. From our experience we can tell that it can happen that you may have a really good card and be offered two other cards for that, and while it seems like a good trade, if you know what you're doing you might realize that the trade is not in your favor after all. We really wanted to make sure that everybody had a good time with »Hearthstone« without having to worry about the value of a card. That was one of the biggest factors for the crafting system as a means to improve your collection rather than having to trade with other players.

Making Games So trading cards among players won't be an option you may consider for the future either?

Jason Chayes No, at this point we have no plans to add trading to the game.



Blizzard's primary goal when developing »Hearthstone« was to design a card game that would be easily accessible and that players could pick up in just a few minutes.



The cards in »Hearthstone« were based on the »Warcraft« universe which led to many players also trying out »World of Warcraft«.

Making Games How did »Hearthstone« change during the development in general? Have there been a lot of modifications over time?

Jason Chayes I would say that the core of the game was largely understood from the very beginning. We knew what key elements and game mechanics we wanted to focus on; we also knew which character classes we wanted to design and what their skills should be, like what abilities the mage, the priest or the warrior should have. All of that we understood pretty early on. The big changes that happened were more along the lines of how we tweaked their abilities. Early on in the development we had some really powerful abilities, but we didn't really understand how good they were until we went through the alpha and the beta process. A good example for that is the rogue whose special ability summoned a dagger, just like in the final version; however, you could increase the attack value of the weapon one by one if you didn't attack but used the ability again and again, and thus created a kind of super weapon. So after a while we realized that some of our initial ideas didn't work out in terms of the balancing. So we kept modifying a lot of skills and creatures in order to make »Hearthstone« a great experience for everyone.

Making Games How hard is it to properly balance a card game like »Hearthstone«? Particularly with regard to the number of cards in the game, it seems this could be quite tricky.

Mike Donais Card games consist of a lot of game elements that can also be mixed in very creative ways, so it's almost impossible to predict what players are going to do next. But at the same time that's one of the great things about card games because players always come up with new crazy combinations and strategies that we never expected would work at all. So card games are indeed tricky to balance. The challenge for us is finding the cards that are dangerous and putting extra attention to those cards and be extra careful with them while the cards that are not as dangerous can be pushed in different ways to add more fun. So, yes it is tricky, but since we have a very solid team that puts a lot of time and effort into it, we've done pretty well so far.

Making Games How do you usually proceed when you design a new card?

Mike Donais When we design a new card, we come from two different directions. About half of the cards are designed by coming up with the idea, the art, that kind of thing. And then we try to design what the card does in the game. The best example is the »Piloted Shredder« card; we start with an initial concept and then pitch about five different ideas, thinking

what a »Piloted Shredder« could do. What we eventually came up with is that when it is defeated it summons a random 2-mana minion. But of course there were more aspects that needed to be clarified such as the question of how much mana the card itself is supposed to cost and so on.

The other direction we come from when designing a new card is having an idea for a certain gameplay mechanic. For example, we might think that we'd like to have a card or a spell that's triggered immediately when you draw it. Then we might think that this should

be a rare thing to happen and what effect it should have – flames, for instance, and then these ideas slowly merge into a new card.

Making Games A lot of free2play games have a kind of ingame currency – be it diamonds or other things that the player can

buy in order to trade them for items, a new life or such things in the game. Why did you decide against an ingame currency – apart from the gold the player can earn – but to show real prices instead?

Jason Chayes That's right, the only ingame currency we have is gold that the player can earn. We decided not have a currency for content you can purchase in order to keep the game as simple as possible. A fantasy currency always creates a level of abstraction for the players where they need to know, for example, how much money to spend for 1,000 diamonds that they can then use to buy certain content in the game. We didn't like to have this system for »Hearthstone« because we wanted to make it obvious how much a card pack was. This traditional free2play system might make sense for games in the future, but we evaluate that on a case-by-case basis. In case of »Hearthstone«, we wanted the players to always know how much money they needed to spend in order to get a certain number of card packs.

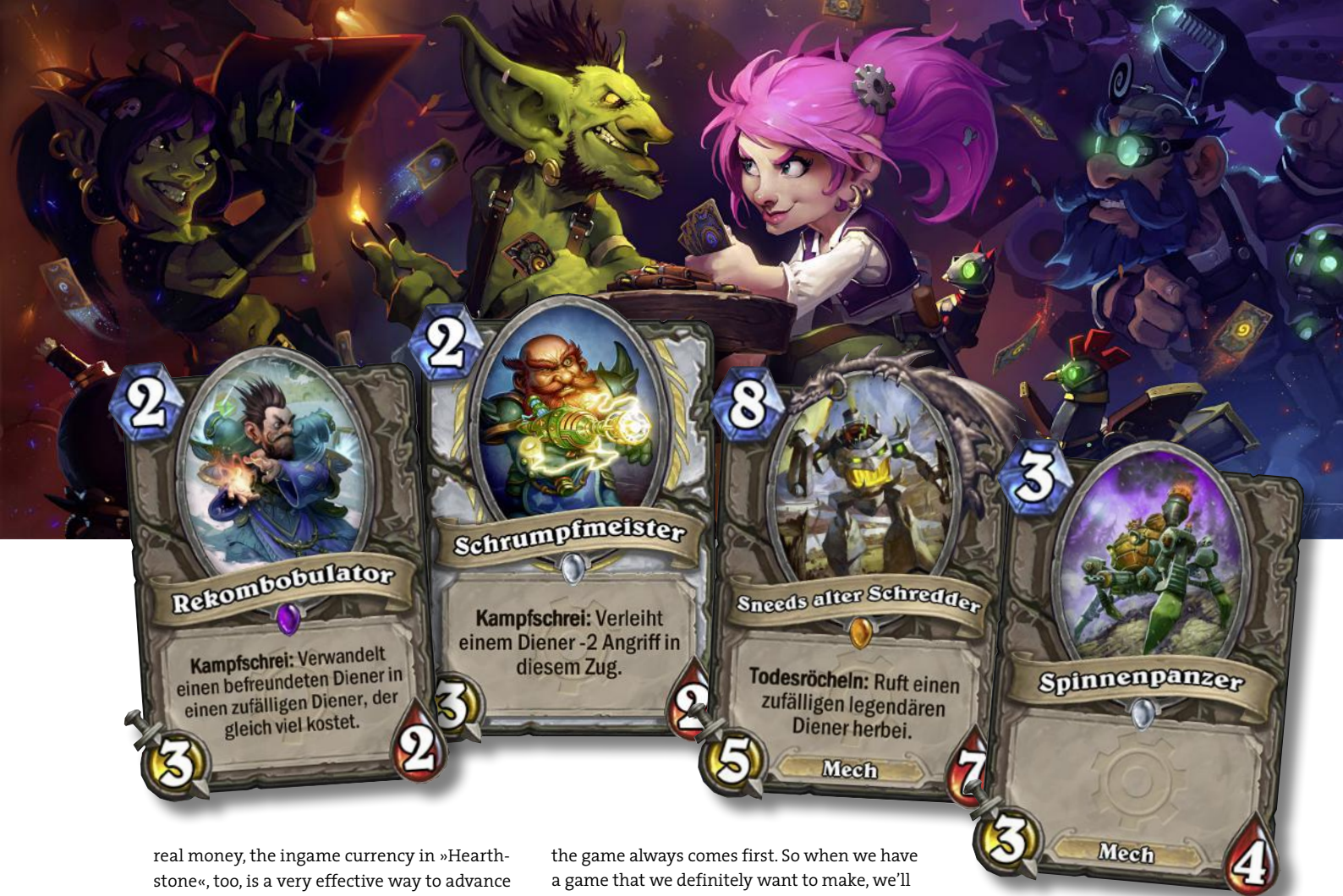
Making Games Do you see changes in the buying behavior of the players because you're using real money pricing rather than a fantasy currency; maybe because your system is more transparent and no conversion is necessary?

Jason Chayes Obviously it's hard for us to compare it because we launched the game with this system. However, we monitor how many packs are purchased with gold compared to real money, and what we noticed is that there are people who are competing at the very top level who have never put one cent of real money into »Hearthstone«. So there are people who earned all their gold through quests or the arena, and they use that gold to expand their collection, to play in the arena and just work their way up in our ranking system. Therefore we can say that in comparison to

»We wanted to take our passion and make a game that anybody could learn within five minutes.«



The »Hearthstone« team designed five different versions of the card »Piloted Shredder« until they agreed on the design that's in the game now.



real money, the ingame currency in »Hearthstone«, too, is a very effective way to advance through the game.

Making Games It's quite uncommon for a free2play game that the player can advance through the game without spending a single cent. Did it never cross your mind during development that this could become a bit risky in terms of revenue? You have to consider the bottom line after all.

Jason Chayes The way it works here at Blizzard is that we always focus on the game first. We want to build a great foundation and brainstorm what's best for the game rather than talk about a business model. In the past we had great successes with traditional retail games, but also with titles based on a monthly subscription like »World of Warcraft«. So when it came to »Hearthstone«, it was clear that we wanted to create a game that really everyone could play and that would be easily accessible. That's when we came up with free2play because that way anyone could simply download »Hearthstone« and try it out. It's not like there's a big hurdle. That's how we arrived at the decision to make »Hearthstone« a free2play game. When it comes to the risks of such a business model, we always thought: If we manage to create a game that people like and are eager to play, we'll also find a way to hold on to this project. So in the course of the development there were definitely some conversations with our business and marketing people where we all brainstormed how to make sure that »Hearthstone« would be successful. One of the great things about working at Blizzard is that

the game always comes first. So when we have a game that we definitely want to make, we'll always find a way to make sure that our players will be able to enjoy it for a long time and to provide more cool products in the future.

Making Games An important aspect about »Hearthstone« being a card game is that the players always strive to collect more and better cards. Do you think this kind of gameplay could be a boost to the free2play business model?

Jason Chayes We know there are players out there who set themselves certain goals. Some of them strive to get a complete set of cards, others just want to collect golden cards; and that's of course great, but that's not the focus of »Hearthstone«. Our goal was to make sure people could play the game for free. You don't really need to collect the full set to be able to compete with other players. In fact, you can actually create a strong deck with only a few cards, regardless of the class you're playing. So whether you're choosing to play »Hearthstone« because you like collecting cards or because you're interested in high-level competitive play – there are different levels of investment you can make, and all of them work great with both types.

Making Games At the moment, free2play games are huge on mobile devices, and »Hearthstone« was the first game Blizzard launched for that market. Did you always plan to release a mobile version of the game or did you first want to see how it would do on PC?

Jason Chayes When we started developing »Hearthstone«, it was supposed to be a PC

When designing new cards, two procedures have proven effective: Either they come up with a design first and then think about its effect, or they first define the card's attributes followed by the design.

»We definitely think »Hearthstone« could be an entry into the world of Blizzard for some people.«

»A fantasy currency always creates a level of abstraction for the player.«



It was at a random internal tournament during the alpha phase when the team found out whether »Hearthstone« would be suitable as an eSport or a spectator game.

game only; we weren't planning on releasing it for mobile devices. But during the course of development we noticed the rise of the tablet, and a lot of games that we were playing internally were tablet games. We loved to talk about them, we loved to analyze them, talk about what potential and play patterns they offered; and when we started to see that more and more people were using tablet PCs we thought that »Hearthstone« would actually be perfectly suited for these devices. We got really excited about the idea of dragging the cards onto the table with our fingers instead of using a mouse, simply because it felt so natural and intuitive. Another factor that made »Hearthstone« the perfect game for tablets was the duration of the matches which, on average, don't last for longer than ten minutes. So we started working out how »Hearthstone« could run on tablets, because before committing to

making a mobile version, we first wanted to be sure that we could make it a great experience. So we brought in a new engineer, discussed the game design and thought about the best way possible to program »Hearthstone« for tablet PCs, so it wouldn't just be a mediocre port of a PC game. After doing a lot of reviews and play-testing etc. we finally came up with a prototype which actually felt like »Hearthstone« for tablet PCs and which we ended up putting into production.

Making Games How hard was it for you during that time to enable cross-platform matches?

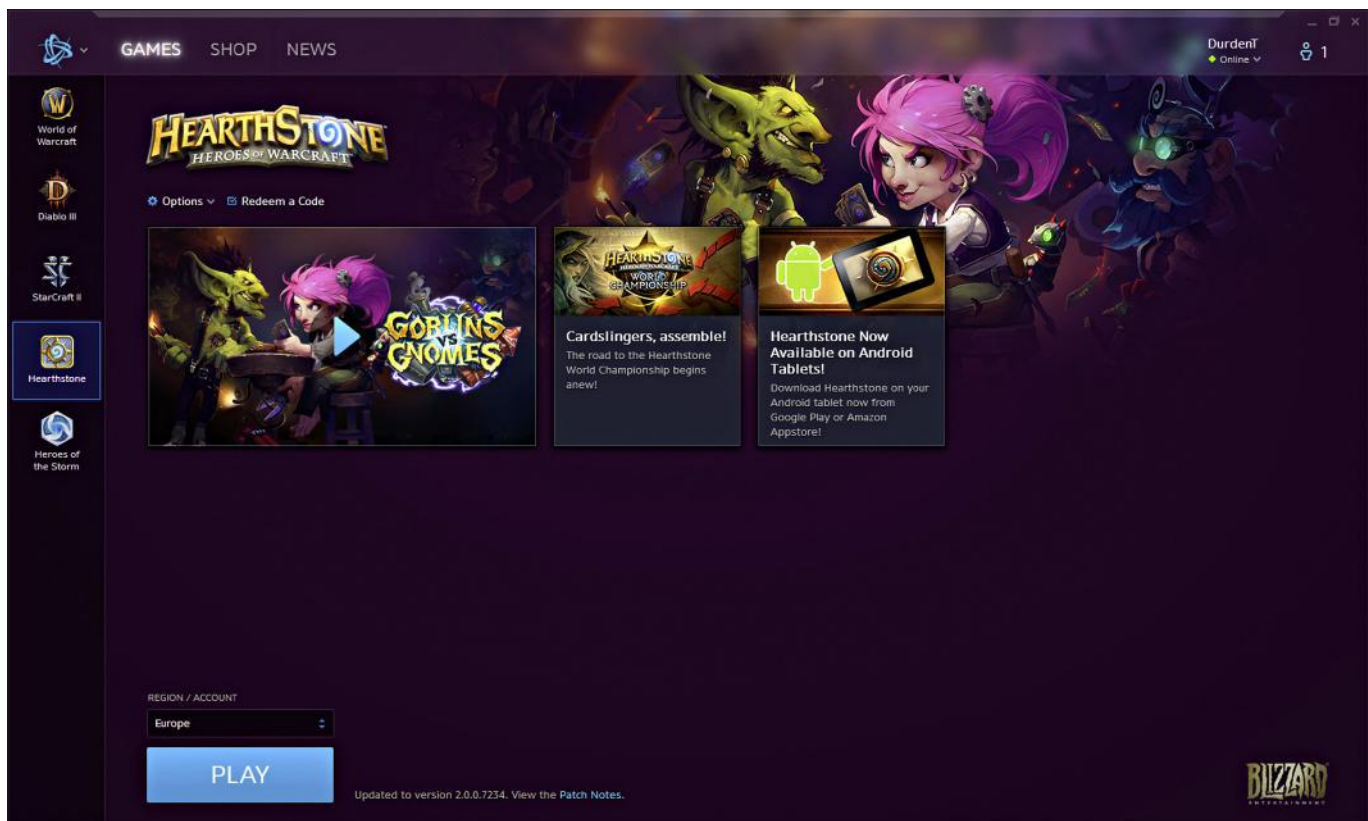
Jason Chayes It was actually fairly easy since all data of an account is stored and managed in our Battle.net system. With all data being on one server, it's quite easy to add an additional client to the system, for example a tablet PC, and access the existing information from there.

Making Games »Hearthstone« started out as a small project with a very small team. It has now become a game that is enjoyed by millions of people. Do you think this approach might be the future of game development - to experiment with concepts in small teams instead of trying to plan a AAA project from the beginning?

Jason Chayes I think this approach will definitely be on the rise; in case of »Hearthstone« it has been very successful for us. But in fact it's not something new. Blizzard have been working on new game prototypes with small teams for many years now, and we think there's a lot of room in terms of development to try out different approaches. When you look at a game like »World of Warcraft«, for example, it's obvious that a lot of people are involved in order to guarantee a constant release of new game content. That's been our business model for a long time now. On the other hand though there's a lot of benefit to having small teams



When testing the game the team realized that watching others play was a lot of fun and started an internal tournament. Today big eSports-events entertain thousands of spectators, just like at BlizzCon World Championships 2014 where winner Firebat was awarded with \$100,000.



Thanks to Blizzard's online system Battle.net where all player data is stored, it was easy to implement the cross-platform function of »Hearthstone« when the tablet version was released.

that are able to experiment without necessarily having a lot of people because it minimizes the costs when you're not sure what you're making. That's always a big challenge because a lot of our process is very iterative; you don't exactly know the perfect formula to making a really fun game right at the beginning. So the experimentation process would be a very challenging thing to do when you have a very big team. Therefore working in a small team is an exciting way to get creative at the beginning of a project. Once a game is in production, we use the small teams to get out new content.

Making Games Do you think that a smaller-scale project like »Hearthstone« is better suited for this kind of development approach than a very complex AAA title like »Diablo 3«?

Jason Chayes I think all kinds of games can benefit from starting small and growing larger over time. It makes a lot of the development process more straightforward when you have a small team; additionally, you can make decisions on game design much quicker in order to create a game that people are going to enjoy.

Making Games Last but not least, is there an anecdote from the development of »Hearthstone« that you would like to share with us?

Jason Chayes We had a process of looking at each other's work and reviewing what we were doing. At that time people started crowding around each other's work stations, looking over their shoulders, commenting on what they

were working on or on the test matches they were playing, like »You should do this instead of that!«. At first we didn't really have any specific goals, whether »Hearthstone« might be an eSport or a spectator game. This work process created a kind of giant internal tournament during the alpha status: The people from the team loved watching each other play, so we ended up having a big tournament in our campus theater where about 100 people showed up. It was the first time that matches in »Hearthstone« were played with spectators, and people were just roaring, celebrating the game and the player tactics. That was really the first step towards planning »Hearthstone« as a spectator experience where not only the players, but also spectators would enjoy the game.

Interview: Sebastian Weber

»Hearthstone« was supposed to be a PC game only; we weren't planning on releasing it for mobile devices.«

THE EVOLUTION OF HORROR

Amnesia creator Thomas Grip explains how horror games work, what scares players and how gameplay should be designed on the basis of *Alien: Isolation* and a lot of horror classics.



Thomas Grip
is Creative Director and
Co-Founder at Frictional Games.

Thomas Grip has been part of making the »Penumbra« and »Amnesia« series and has a disturbing fascination for anything horror. He is currently hard at work on »SOMA«, a sci-fi horror game about the deep sea and the mystery of consciousness.

Alien: Isolation« is an interesting game. It is the latest entry in a lineage of games that I refer to as horror simulators. It does an excellent job at creating tension and uses a lot of the knowledge built up over the years to great success. But, because it has such a laser focus on a certain type of play a bunch of problems arises and other parts of the package suffer. It is a great game in many ways, truly excellent really, but there are some fundamental problems. These lead to, for me at least, a devastating flaw: At its core it fails to be a faithful emulation of the original »Alien« (1979) movie.

The history of horror games

Before we can properly discuss the game, we need to talk some video game history and de-

sign theory. Over the past, there has been two different schools of horror games. One that has a horror wrapping on top of standardized gameplay (horror wrapping) and one that tries to recreate the happenings of a scary movie/novel (horror simulation). The former is quite well known and started with games like »Lurking Horror« (1987). Mechanically, the game played like other contemporary adventure games, but took place in a scary setting with events meant to frighten the player. The latter one is a bit harder to nail down precisely, but I would say it started out with »3D Monster Maze« (1982), a game that is neatly captured in its name: the player is trapped in a maze and needs to escape a monster (in this case a heavily pixelated T-Rex).

While the horror wrapping design has thrived over the years, being a design corner

stone for games like »Phantasmagoria« (1995, an adventure game), »7th Guest« (1993, a puzzler) and »Resident Evil« (1996, an action shooter), the horror simulation is much rarer. After »3D Monster Maze«, it took 13 years for the next game to do it somewhat properly: »Clock Tower« (1995).

It is now time to dig into the distinction between these two types of design. What sets »Clock Tower« apart from »Resident Evil« is that its core mechanic is not there to entertain, it is there to put the player in the shoes of a protagonist in a horror story. »Clock Tower« does this by having a single monster (little guy with giant scissors) hunting the player in certain scenes. The players can choose to hide in closets, under beds and hope that the monster does not find and kill them. Compare this to the core mechanic of »Resident Evil«, where the player needs to scavenge for ammo, weapons and health potions and then combat the monsters encountered. The gameplay of »Resident Evil« could really work with any sort of theme and setting, while the one in »Clock Tower« is much more focused on being a horror experience. It is important to mention that »Resident Evil« does tons of things to crank of the level of scariness; scarce ammunition, inventory management, limited saves, etc. But none of these give rise to any specific horror scenarios; the game is still all about shooting various enemies in order to progress. There are very few sections in »Resident Evil« that would fit a horror movie or novel, but »Clock Tower« is filled with these moments.

The core difference can be summed up as »Clock Tower« being focused on having plenty of verbs that are related to a horror movie: hide under bed, looking into mirror, run away, push monster, etc. As the player plays the game, they reenact scenes of horror simply by following the rules set out by the gameplay. »Resident Evil« hardly does it all, the focus being much more on standard tactical combat with certain scariness attached to it.

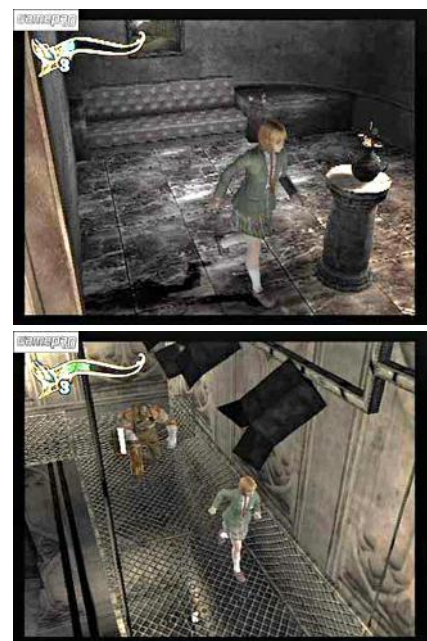
As a horror simulator, and really horror game in general, »Clock Tower« is not very successful though. First, much of the gameplay is actually pretty standard adventure game affair. Second, the actual chase moments are quite clumsy and frustrating to play, rarely resulting in any proper feelings of dread and terror. Despite these, quite major shortcomings as a horror simulator, »Clock Tower« is well worth studying. If you look past its flaws, there exists a focus on making horror elements playable that just didn't exist in games at the time. In fact, it is really hard to find similar games until quite recently. »Hell Night« (aka »Dark Messiah«, 1998) has the player running from a single monster and features cool stuff like a look-behind-you-button and having to carefully choose a companion. But it also suffers from lots of problems and is often a very frustrating experience.

Even such horror classics such as »Silent Hill« (1999) have little focus on trying to do proper horror simulation. Most of the game is based around solving puzzles and fighting enemies (while running is sometimes advisable, most encounters are best handled through combat). Instead a lot of your typical horror moments are not simulated, but put outside of the player's control. For instance, in »Silent Hill 2« (2001), when the protagonist hides in a closet it happens in a cutscene. Remember, this is a scene that played out through gameplay in »Clock Tower«. If you look at the large body of the gameplay the player does in a »Silent Hill« game, little of it resembles what you would see in a novel or film. In fact, it resembles little of how a rational person would behave in similar scenarios. This does not make the games bad by any means, but it is a crucial concept to keep in mind. The games heavily rely on standard, often narrative-wise nonsensical gameplay in order to create the foundational engagement for the player.

Mental modeling

The next game, after »Clock Tower«, to really give horror simulation another proper go was »Siren« (2003). Here the player's actions are a lot more aligned with how it makes sense for them to act. For instance, it features a map without any indicator of the player's position, and also allows you to see the world through the monsters' eyes. In all, it creates a much stronger sense of being inside a horror story. The problem with »Siren«, like »Clock Tower«, is that this focus amounted to a very frustrating experience, which in return easily diminishes the overall immersion and scariness of the game. Once again, this type of design didn't gain any traction.

I now need to briefly discuss a design concept called »mental modeling«. A concept that is closely related to the difference between horror wrapping and horror simulation. When



»Clock Tower« which was released 1995 presents the player horror (to be specific a single monster) like known from movies or novels and could work for both media.



The original »Resident Evil« (1996), the figure shows the remastered edition, uses horror elements but focuses more on survival aspects for its gameplay - that clearly sets it apart from many horror games from the same era.

you play a game like »Resident Evil«, every encounter is a very tactical and precise decision. You look at what kind of opponent you are facing, what weapons you have, the current ammo supply, health levels, etc. The model in your head is much less about the appearance of the creatures, and much more about pure numbers. This can be very stressful, and combined with a horror thematic the total experience can feel quite scary. But on the whole very little is left to the imagination and narrative-related intuition. In a game like »Siren« however, you are much more worried about what you cannot see, and your mind is racing with trying to predict future happenings. Combine this with a map that forces you to think yourself as part of the environment, and monsters that you mostly keep track of by inferring their position, and you get a mental model that is far more horror-like and vivid. The problem is that since you lack the sense of numerical certainty that »Resident Evil« supplies, it becomes much harder to formulate correct tactical decisions. This is a fact that I think has a part in making these kind of games so frustrating. It is clear that the mental modeling that »Siren« has is better suited for a horror game, but it was not yet working properly gameplay-wise.

Now back to game history again. The release of »Siren« marks the beginning of the end for what I would like to call the golden age of survival horror. It had brought us gems like »Resident Evil«, »Silent Hill«, »Fatal Frame« (2001), all of which brought a very fresh take on the horror experience. But after that, the well seemed to dry up and horror games got increasingly action oriented. »Resident Evil 4« (2005) really got this trend going. One reason for this change in focus is partly diminishing sales and increasing production costs. Another major – and connected – reason was almost certainly the lack of evolution in the genre. All of the big games had been based on horror wrapping, with most of the gameplay being quite standard. The wrapping did not allow for much change, and when the focus was put on the underlying mechanics, the horror rapidly faded.

The revival of horror

It took a while before anything new came along, and this is where we enter the picture. When we released »Penumbra: Black Plague« (2008), we made our attempt at doing a game without having any weapons. It had a bunch of issues, but showed promise in how it changed the player's perception of the game's world. The decision was based on lessons learned for »Penumbra: Overture« (2007) and from the lineage of games discussed above. The following year sees the release of another combat-free horror, »Silent Hill: Shattered Memories« (2009) which also puts emphasis on crafting more of a horror simulation, but also manages to get a bit too frustrating. Upon releasing »Amnesia:

The Dark Descent« (2010), there is one thing I am especially proud being part of creating. Simply by using the game's basic system, we can get the players to hide in a closet and fearfully watch as a monster pass them by. It felt like we basically managed to recreate, through gameplay, the closet cutscene from »Silent Hill 2«, and properly simulate one of the main mechanics from »Clock Tower«. I feel way too biased to say how much effect – if any – this had on horror games in general though, but for me personally I feel it is one of our biggest contributions to the genre.

Two years later, the short and free game »Slender« (2012) is released and this what I think really kickstarted the horror simulation. »Slender« is a simplistic game with not much in terms of standard gameplay. The player is simply tasked to collect a few notes in an open, but still slightly maze-like environment. Where the game gets all of its engagement from is how it manages to simulate its horror. You cannot look at slender man, you have to move carefully, you cannot use your flashlight too much, you need to stay away from spooky sounds, etc. All of these gameplay elements are quite vague and together they give rise to a rich mental model. The end product is that you got a very engaging horror experience from almost nothing. Remember that »Slender« has very little story and interesting goals. It is all about existing inside a certain virtual space. It is all about raw horror simulation and little else. While other indie games like »Hide« (2011) have done similar things in the past, »Slender« proved it was really a viable method of making a game. If you just make things vague enough, and allow the player to play based on that vagueness, you can make engaging games of pure horror simulation. In some ways, »Slender« is sort of like the »Super Mario Bros« of horror games, a distillation of what makes the genre work into its purest form.

Since then a bunch of similar games have followed, one of the most successful being »Outlast« (2013). The story in »Outlast« is paper-thin: A journalist enters an old asylum where experiments are rumored to have taken place, is met with the place being overrun and now has to escape. There are no puzzles, no standard gameplay, it is all a bunch of levels where you run or hide from insane inmates. Also included are a few breather levels where you simply walk to (or try to find) the next destination by going through some spooky environments. Scattered around are also documents detailing background lore, but you gain very little from reading them. You can pretty much go through the whole game ignoring them and still get a coherent narrative. What makes the game engaging is the situations that it puts you in. Hide from monsters in lockers, sneak past them as they talk gibberish, walk past weird looking denizens not knowing if they will jump at you or not, try to make out dangers using a grainy



The next game after »Clock Tower« to approach horror simulation right was »Siren« (2003) which aligns the player's action more believable than many other games and creates a strong feeling of being part of a horror story.

night vision camera, and so on and so forth. Almost any gameplay scene that plays out could be directly baked into a horror movie or novel. »Outlast« is a pure horror simulator, and a narrative naturally arises from the situations the game throws you in.

The analysis of Alien: Isolation

And now, finally, we get to »Alien: Isolation«. This is the latest breed of horror simulations, and a very successful one at that. The big feature in this game is that there is only one monster that can appear anywhere and at any time. In many ways, it is really the 2014 version of »3D Monster Maze«, in which the whole game was built around trying to figure out where the monster was located and how to avoid it. »Alien: Isolation« obviously has complexity far beyond that, but it is striking how similar the basics are.

There are bunch of elements that all work together to craft a mental model that perfectly fits the game and make »Alien: Isolation« really scary. First off, like in any other horror game, the sound is vital and you need to constantly listen for clues. This can be clues from other survivors (who are hostile towards you), malfunctioning androids, or the alien monster itself. Not only do these paint a terrific ambient soundscape, but they are also very important for you in order to progress. You also need to be conscious about how much sound your own actions are causing. The game warns you to not use the notorious bleeping motion scanner too near people as they might hear you, and after that you get really paranoid about any sounds you might be making.

On top of that, the save stations, which are crucial to locate, give off a faint and distinct beeping noise. As the environments are dark, and the stations can sometimes be found in a hard to spot location, you have another reason to listen carefully. The difficulty level in the game is quite harsh, and the tiniest mistake can lead to sudden death, so you are always eager to save the game. This leads you to listen extra carefully when you have gone a long way from a save, making you vulnerable to any other sudden noises. Remember those jump scare videos that told you to watch or listen extra carefully only to throw a spook at you when you least expected? It is sort of like that, but without you feeling annoyed afterwards. Add to the fact that the alien can really appear anywhere in the game, and you get a mind model that is primed for picking up and seriously considering even the slightest sound.

Because the alien feels so random in when and how it appears, it is hard to get a grip on what sort of signs to watch out for. The mind model becomes vague, and more prone to paranoid imagination, which is obviously great for a horror game. This is combined with the fact that unless you have one of the beefier weapons, if the alien spots you, you are dead. With



»Penumbra: Black Plague« from 2008 did not give any weapons to the player who therefore was completely defenseless against the games' monster. This fact created a lot of tension during gameplay as the player always had to be aware of his actions and movements.

no possibility to run away, the player gets very conscious about their movements, the alien's possible location, and – yet again – the sounds that they hear and might be causing.

Combined with all this is an assortment of items (noise makers, flares, a revolver, etc.) that all have pros and cons (e.g. flares light the dark, but also attract attention) along with a certain interplay between the different types of hostiles. For instance, if there is a room with human adversaries you could throw a noise maker in among them, attract the alien and then have that kill them off. But that also means you now have an alien in your vicinity, which might prove more dangerous than the humans. And since the systems are vague, it is hard to predict outcomes and you live in this constant uncertainty.

Another aspect I love is how the alien monster forces you to sneak. It is quite common in



Even though »Slender« (2012) is very simplistic in terms of gameplay its creepy setting quickly created a lot of hype about the game and about the horror genre in general – that led to a lot of similar games and a lot of Indies experimenting with this genre.

stealth games that you can simply lie in wait and club down patrolling hostiles, and that that tactic is far better than the (narrative-wise, better fitting) approach of sneaking past them. But in »Alien: Isolation«, you do not want to risk any sort of alarm, and staying hidden almost always feels like the better option. In turn this makes most humans that you encounter seem more alive. You mostly only see their vague outlines and hear their conversations from afar. In your head you build a much more vivid picture, since you are never pushing the game into displaying their often immersion breaking behaviors that occur when they are up-close.

All these actions available coupled with the resulting mind model lead to game-play spaces that properly simulates horror. Sometimes even unscripted horror shows play out in front of your eyes. For example, a band of survivors can be caught off guard by the alien as you hide in a locker and you end up staring at the onslaught terrified that the monster will come for you next. And like with »Outlast«, just about any scene that plays out would be fitting in a book or movie. A horror story unfolds as you play and is affected by your actions. And unlike similar games based on horror wrapping, e.g. »Dead Space« (2008), the actions that you do feel sensible and realistic. Apart from some more gamey mechanics like the save station and scrap collection, you behave just like a character would do in a movie or book version of the same story. This is horror simulation at its best.

The bad about being a horror simulation

But this super focus on being a horror simulation, also starts showing cracks in the game as a whole. For instance, just like in older games of the same genre, »Alien: Isolation« can be very frustrating. The tension built up from being 20 minutes from your last save, quickly

turns to anger and frustration when you are killed seemingly out of nowhere. While still vague (which is essential for giving rise to the right mind model), it is predictable enough for you to be able to get past any threats if you are just careful and cunning enough. Still, this part is divisive, as can be seen by the review scores and I have myself felt extremely frustrated with the game from time to time. There is a certain sweet spot in how to approach situations properly. Too aggressive, and you will die a lot. Too passive and the game's pacing gets messed up. The trick is to be able to move forward at a steady pace and still be cautious enough to avoid death. A way to fix this would have been to make the alien AI better adapt to the player's style, so if they hide a lot in lockers, it backs off quicker and keep things interesting and the pace at the right level. The developers should have probably focused a bit less on challenge alone to provoke a certain mindset and also have had some elements to dampen the difficulty for players that were behaving properly, but are being a bit too passive to get a good experience.

While the frustration and bad pacing are clearly issues, I do not think they are that bad and, as noted above, it should be relatively easy to fix. What is a much bigger problem is how these systems give rise to a very simplistic narrative. The first part of the problem are the save stations. Having these are a crucial part of making it all work, partly based on the simple reason that after a little while you get too comfortable around the alien for it to be intrinsically scary. There needs to be some gameplay aspect that keeps the tension up, and keeps you in the right mindset. The save stations do just this. But the side-effect is that the save stations become the biggest objective for you as a player. Finding the power station that you need to activate is not at all as important as finding the next place to save. This means that your personal narrative becomes



»Hide« was released one year earlier than the comparable »Slender« and used a similar gameplay mechanic, making things vague enough to encourage the player to go on and to create a pure horror simulation.



Despite the narrative as well as the gameplay of »Outlast« (2013) being paper-thin the scenes the game puts the player in are working perfectly in terms of the definition of horror and could directly be taken from a horror movie or novel.

dumbed down, and ends up being a simple hunt for loot and save locations, without any thought on what the higher level fiction behind your actions really is.

The problems with objectives do not stop there though. Another issue is that they are all extremely simplistic and without any interesting narrative significance. They are all about powering up things or finding keycards. It is old school mission design with a thin layer of narrative coating. While these sorts of boring objectives are pretty common in games, I think »Alien: Isolation« has an especially hard time getting away from it. Because the game is constantly so dense with information that you need to keep track of (save stations, motion tracker, alien signs, loot, resources, etc.) you really cannot manage to keep any complex objectives in mind. It is also crucial that the player has a good idea of where they should be headed, because too much disorientation and the mind model starts breaking down. This leads to basically decent objectives (e.g. find a keycard on a dead doctor by figuring out what rooms he has visited) becoming very handheld experiences, as the game constantly marks your next destination. The simple reason for this is that less hand holding would have made it too frustrating. You can see similar mission design in »Outlast«, and while I think there was more room for improvement there, it is a design decision that probably stems from the same issues.

Another connected issue is that the notes and audio logs you find in the game never really feel that relevant. It is another issue similar to »Outlast«, although I think that the content was way more interesting in »Alien: Isolation«. But the real issue is that you cannot really have too much important information in these, because accessing terminals means a danger to the player. It is really hard to explore and think about the environments and backstory when you are constantly worried about the threats the game might throw at you.

Story and horror go hand in hand

As you look closer at these flaws in »Alien: Isolation«, it becomes more clear that it really is just a pure horror simulator, like »Slender« or »3D Monster Maze«, just with more sections to play through. The aim of this game is not to tell a longer horror story. The aim is to put the player in different scenarios involving hostile human survivors, creepy androids and/or an alien. Any interesting narrative then arises from how these scenarios play out. It has no real story at a higher level apart from the basic »get the hell out of there«. There is nothing wrong with this of course, and the basics of the game work fine as is. But the issue of »Alien: Isolation« is that it just goes on for too long. If you (like me) want to get some sort of deeper narrative experience from your game to keep going, the game wears thin after just a few



Basic gameplay of »Alien: Isolation« has a lot in common with »3D Monster Maze« from 1982 where the player was facing just one monster in a maze-like environment. »Alien: Isolation« utilizes those mechanics greatly even if facing a lot of problems at the same time.

hours. This is a bad thing for a game that lasts for at least 15 hours. »Outlast« takes about 5 hours to complete, and that feels like a better fitting length for this sort of gameplay.

»Alien: Isolation« has another big problem. If you went in hoping this would be an interactive take on the first movie, you will be disappointed. Rather than being some sort of video game adaption of the original film, it is more like a version of »Alien 3« or »Alien: Resurrection«. The first movie is the birth of a monster, it is a long build-up of a creature followed by a final confrontation. It is much more about discovering some hidden lovecraftian horror, than about people fleeing for their lives from a well-established creature. So what the game ends up being about; sneaking from place to place, just constitutes a minor part of the movie, and is a far cry from replicating the experience.

It is hard to blame the developers though. In order to deliver a similar narrative they would have had to rethink the alien quite a bit, and most likely diverge from the source material. Something that I think Sega or 20th Century Fox would not be too happy about, as the creature is the game's foremost selling point. But even if we disregard that, it would be very hard to use the game's given form to emulate the make-up of the movie. If we trace the game's gameplay lineage way back to »Monster Maze«, we see very little about the other things that make the movie so great. All of these games are about putting the player in a predefined situation and to let that play out. The great strength of the genre is to provide the thrills of being the hunted, and to have a super focus on this.

»Alien: Isolation« does some attempts to better replicate the source material. For instance we are taken on an expedition to the iconic derelict spaceship, but for me all of this falls flat. The eggs, facehuggers, and all that are already so well established that

you never get a sense of mystery from it all. Worse still, this section differs so much from the movie – which comprises of sneaking about – that there is no tension. The game has trained you to look for save stations, be aware of your resources, to listen for certain sounds, etc. in such a specific way that when you are taken out of this space, the game loses a lot of its impact.

I think that horror simulation is the sort of design we want to strive for when making horror games. I love how it is possible to set up scenarios where the player gets to experience terrifying scenes, fitting for a horror movie or novel, simply by playing. This is the sort of storytelling that we want from games, and I do not think going back to merely using horror as a wrapping is a good idea. However, it is pretty clear that there is a big problem here. Horror simulation as it is currently done, is very limited in what it can let the player do, and what higher level narratives it can tell.

We had similar issues in »Amnesia: The Dark Descent«. When playing the game, all you do is go from room to room, solving puzzles and avoiding monsters. But as you read about the backstory you hear about opening tombs in the desert, the first visit to the Brennenburg castle and much more. All of these are moments that an ideal horror game should have let the player play, not just read about in some diary entries.

The issue is not just to recreate these sort of scenes, but to make sure the experience is fitting. As can be seen in »Alien: Isolation«, the trip to the planet lacks the immersion that the sneaking around has. When you try and hide from the alien, your mind model is aligned with what is happening and you act like a protagonist would do. But when visiting the derelict spaceship, you could have just as well been watching a cut scene. There needs to be something more in order to properly bring this sort of scenes into the standard that proper horror simulation has set.



»Dead Space« (2008) is based on horror wrapping just like »Alien: Isolation«. But: The actions of the player don't feel sensible or realistic in any way in »Dead Space« which would be best for a horror simulation.

Two important learnings to create tension

First of all I think there needs to be more restraint on how much of the basic gameplay is used. Even though »Outlast« takes about 5 hours to get through, it has a hard time keeping the gameplay fresh and the tension up throughout. For me, the game started to feel slightly drawn out the last third or so. »Amnesia« also suffered from this. Enemy encounters got pretty predictable and less scary after you had gotten through half the game. While the exact time at which this sort of gameplay stops feeling engaging is highly subjective, I think most can at least agree that the less practice you get with the encounters, the more intense they feel. Therefore lies in the best interest of a horror game to use its monsters sparingly.

Second, there needs to be much less reliance on difficulty in order to build the proper mind model. The way that games like »Alien: Isolation« and »Siren« are set up really increase the tension, but it does so at the cost of making the game's scope very narrow. As explained above, it is hard to have much else going on in order for the player to enjoy the experience. Also, as soon as you remove any of these elements that rack up challenge (like save stations) a lot of the tension goes away. What games like these are after is to couple the sensation of stress of losing progress together with a firm horror setting. The psychological effect is that the player takes the tension they are feeling and projects it on the frightening visuals and sounds. But building purely on this is a fragile structure as it means you always need to present a mechanical device of stress to the player, and this leaves out a lot of horror scenarios.

Branching and player decisions help immersion

I think that the answer lies in having some sort of uncertain outcome as an ingrained gameplay device. Instead of having the player fear »what if I lose my progress?« they should be thinking »what if I affect the world in a negative way?« This is a kind of thinking that can be applied to a much wider range of things. Probably the best example of this in action is »The Walking Dead« (2012) series. Here one can clearly see the range of scenes that can have tension, purely based on an uncertain outcome. And what I especially like is how personal the choices can be, e.g. deciding between which person gets your trust. The most apparent problem with this sort of gameplay though is that it requires branching, but this is not as big of an issue as it seems. »The Walking Dead« has in fact very little branching, but even if you know this, you still feel tension by the choices. There is still enough uncertainty for you to feel immersed and care about the decisions.

However, »The Walking Dead« is quite different from how a horror simulator works: In a horror simulator the player needs to be in

control most of the time, and cutscenes should be minimal. So it is not possible to just rip out the mechanics and apply them as is. What we need to do is rethink how choices can be set up and what the effects can be.

We did some very early testing of this in »Amnesia: Dark Descent« and its free extra story »Amnesia: Justine« (2011). In »Dark Descent« we have no proper choice, but there are some slight consequences to failure. If the player is killed by a monster, the world is slightly changed. This was not enough to be on the level of the tension gained from fear of losing progress, but it gained a certain level of uncertainty that helped to keep some of the fear. In »Justine« we also tried having optional puzzles that resulted in a person getting killed if you failed. This also worked pretty well, and it meant that puzzles – a previously pretty separate element of gameplay – got more of a horror simulator feel (somewhat recreating scenes from movies such as »Saw«).

Conclusion

These are just small things though, and the big problem is to have it work on a larger scale in a smooth, coherent manner. I think the first step is to see if you could remove the penalty of death from standard sneaking gameplay, but still retain the same sense of tension. There are three big things to gain if this can be achieved. First, to remove a lot of the inherent frustration connected to the »fear of progress loss«-design, and have less risk of breaking immersion. Second, it will allow for more integration of exploration and more complicated objectives, as you remove the forced repetition coming from a design based on difficulty. Third, it will give us a glimpse of how we expand the horror simulator and have it cover things beyond sneaking by and hiding from monsters. My hope is that we then could take a stab at making a proper horror simulation, a recreation of experiences like in the original »Alien« movie.

This is what we are currently experimenting with in our upcoming game »SOMA«. Since I do not want to spoil the system I cannot go too in-depth on our approach, but I can give a basic outline. The idea is that by having choices inserted directly into the game world and have the way you chose to handle these change how the narrative unravels. These choices can simply be whether you interact with a certain object or not, or it can be more vague things, like how you behave around a certain creature. The narrative effects will not be any sort of heavy branching, but smaller things like making upcoming sections scarier (e.g. by removing lights), killing people you encounter, changing the way you perceive a character or even how you feel about yourself. Our hope is that by having these sort of decisions as an integral part of the game world, the players internalize them and make it part of their mind model. Then, just like the tension you feel



Beside of monsters or violence even situations where the player has to take decisions or branching of the story can create tension in horror games as the results of those choices are unclear – as the Telltale adventure »The Walking Dead« (2012) clearly shows.



For their upcoming game »SOMA« Frictional Games is experimenting with gameplay mechanics to create tension. The player will have to take a lot of choices which won't always lead to branching of the story but most of the time will affect the game world directly, e.g. lights might be removed in a certain area due to the player's choices before. This is meant to create tension on the long run.

by wondering where the next save station is in »Alien: Isolation«, you will feel tension by pondering what ramifications a certain set of actions might have.

This is just a little step forward, and until we release the game, we will not know how well it works. I do think it is crucial that we start to think about these things though. Just as the horror genre stagnated in the mid-2000s, because horror was merely a wrapping, the same might happen if we fail to move beyond »chased by monster«-scenarios. While there is nothing wrong with these sort of games, I think it would be foolish to be satisfied with just that. There is so much more to explore in horror, and the success of recent horror simulators gives me hope that video games can handle it.

Thomas Grip

MOON HUNTERS

There is no formula to run a successful Kickstarter campaign. However, promoting the game over several months before the Kickstart and including little things like gifs instead of screenshots certainly can do the trick, as the campaign for Moon Hunters proves.

A KICKSTARTER POST MORTEM



Tanya Short
is Creative Director and
Co-Founder of Kitfox Games.

Tanya edited and produced the feminist games journalist weblog *Gamer-Girl.org* from 2003 to 2007 and published a science-fiction tale of colonial independence, «Excerpt from the Mars Address», in *Mars Magazine* in 2006. She earned a Master's degree in Interactive Media from the Guildhall at Southern Methodist University in 2008. Tanya was an A.I. Designer and Senior Designer at Funcom Montreal from 2008 to 2013 and worked on «Age of Conan» and «The Secret World».

Kitfox Games is a 4-person studio in Montreal, Canada. Our Moon Hunters Kickstarter was our first such campaign, and we asked for 45,000 Canadian dollars as a base funding goal. When it closed, we had earned 178,986 dollars, which was nearly four times the amount we had asked for, from 6,000+ fans. By all measures, it was a great success.

If you're reading this hoping to understand our strategy, here it is: we tried to convince our fans it was worth their time and energy to tell their friends, family, and the world. Facebook, Twitter, and Reddit were our top sources of backers, getting us much more attention than any press coverage.

However, please understand that there are many avenues to success. I have spoken with other Kickstarter managers and their strategies are sometimes completely different. One individual who had helped with a series of high-profile indie campaigns advised that he always chases press first, and that tends to be his highest source of income. I'm glad it works for him, but ours was the reverse.

Furthermore, due to Survivorship Bias, it is generally more useful to learn from failures than from successes. So I recommend everyone to do some hunting and find those devs brave enough to talk about their unfunded campaigns ... and give more weight to their learnings than you do to mine.

I wish the best of luck to all game developers, and simply urge you to assess your game's appeal honestly, and strategize accordingly.

Note: all dollar amounts cited will be in Canadian dollars, as our campaign was hosted

by Kickstarter of Canada, despite the fact that most of our backers were Americans.

Preparations

We first put Moon Hunters on the Square Enix Collective site in January 2014. We were one of the projects in its first cohort, proud to be alongside «World War Machine» and «Game of Glens». We managed to scrape 1,000+ newsletter signups of that initiative, which was covered by all the major press, even if the reporters showed a bit of healthy scepticism.

We built our community via blog and newsletter for eight months before launching the Kickstarter campaign on August 26th. We had almost no press coverage during that time, yet the number of our newsletter subscribers slowly grew. This newsletter was, in my opinion, our #1 advantage.

While the rest of the team finished up «Shattered Planet», our artist Xin Ran Liu developed the Moon Hunters art style, iterating several times to find his voice in pixel art. Similarly, I used that time to work on the core feature and world design, plus get some practice talking about them to players and journalists. So it was six months before our programmers, Jongwoo Kim and Mike Ditchburn, started full-time on prototyping.

By early August, we had a very simple prototype, with stand-alone scenes of combat, dialogue, and scripted sequences. It wasn't production-approved code, it wasn't a «game», but we knew what we needed from it to make a representative gameplay video. And maybe more importantly, we knew what we thought was cool about the game, as a concept, and had ideas of how to show that to players. So

Xin and I took two weeks to make a video and prepare our page. By the end of August, we were finally ready for launch.

Expectations

Seeing our funding goal, non-developers usually commented, »Yeah, 45,000 dollars sounds about right for that kind of game.« One snide individual even said, »I bet I could make that game in two months.« But developers usually commented, »45,000 dollars? What the heck could you afford to do for that? You have four full-time developers! You're making an RPG! That takes YEARS!«

Before the Kickstarter, we had access to enough cash to barely scrape by making the game. However, our existing funds (from Shattered Planet sales, an external investment, and a government fund) allowed us no buffer zone or extra bugfixing time.

We pay ourselves as little as we can, but multiple members have to pay debts in addition to their usual eating, rent, etc. We literally cannot »work for free« (i.e. make the game without paying ourselves), and even minimum wage adds up over time. Those of you who can afford to do »sweat equity« have a great advantage over us. But we knew, looking at the budget, that despite our funding, we would probably have to do Early Access, which isn't ideal for a game like Moon Hunters.

So the Kickstarter served three purposes, in order of priority:

- **Money:** enough to develop the game to a polished, complete state
- **Community Building:** a word-of-mouth and press push worth the time/effort
- **Risk Management:** failing now would indicate the game itself wasn't good enough

So, at a minimum, we figured 22,000 dollars of profit would cover the Kickstarter labor and also help fund a two extra months of development ... which, combined with Quebec taxes, fees, backer reward costs etcetera, meant we needed to ask for about 45,000 dollars as a funding goal to realize the project.

Before we launched the campaign, each team member guessed how much we'd end up with (we do this for all launches, as an experiment in predictive powers). The lowest guess was 48,000 dollars and the highest was 80,000 dollars. In short, we expected to reach our goal but not double it. Healthy optimism.

All signs from our research of other Kickstarters pointed to a quick first few days, a flat line, and another peak at the end. This is how most Kickstarters work. Other than a few lucky tweets or press coverage (which is harder and harder to acquire), usually donations flat-line. We braced ourselves for a long haul. But then ...

Results

We were funded on day 3, while Jongwoo Kim (one of our programmers) and I were off man-



Twitter was one of the top sources of backers for the Kickstarter campaign.

aging the Kitfox booth at PAX Prime, honored as part of the Indie MegaBooth.

It was a bit stressful to have half the team away while the remainder was trying to manage an unexpectedly successful campaign, especially since there was no wi-fi at our PAX booth and I hadn't bought an international data plan. So every few hours, I would run across the convention center to check my email and messages from the team to see if there were any emergencies. Xin responded to questions, we hurriedly solidified our stretch goal plans, strategized how best to proceed, and then we hoped.

The campaign never quite flat-lined like we expected it to. It did calm down to 1,000 dollars a day at some points, but ... that's obviously not the same as suffering the (normal) terrors of 0 dollars/day.

Other benefits received as part of the Kickstarter:

- 800 new Twitter followers (from 700 to 1500)
- 500 new Facebook Likes (from 700 to 1200)
- 800 new mailing list subscribers (from 1300 to 2100)
- A few thousand bonus website visitors
- A few dozen extra Shattered Planet sales & coverage requests
- New relationships with other indies that Kickstarted around the same time

Of course, our friends and family joke that we're rich now. Hahaha. No. We continue to pay ourselves essentially minimum wage, to reach the point where we can release Moon Hunters at its highest-quality point. In fact, we pay our contractors more than we pay ourselves, in the hopes that the game's sales will let us live more comfortably when it releases.

But all that is in the future. Post-mortems are about the past.

First, the happy part.

What Went Right

There is likely an alternate dimension in which we didn't reach our initial funding target in the first few days. In that scenario, it is likely that we didn't reach it until the final week, in which case we would have earned »only« 60,000 or 70,000 dollars. That would have satisfied us. So why did things go unexpectedly well?

It's important to note that we did not follow a formula, and there is no formula to be fol-

lowed – no matter how much preparation or planning, no matter how cool a game concept is, there is always an element of luck in any success. We were lucky, Survivorship Bias, etc.

However, a few things we did seemed to pay off as planned, that we feel proud of.

Groundwork: The six to eight months of promotion (even just once a month) did seem to create the potential for a snowball effect. The video we produced for the campaign, using completely new footage, seemed to resonate with the fans we'd built up, satisfying their expectations.

Square Enix Collective: As a relatively new studio, the added legitimacy of a Square Enix stamp of approval couldn't have hurt, and their mailing list likely also helped create a chance for the snowball effect. The game concept we put together had a great overlap with their fanbase.

Feedback Incorporation: With our game up and building momentum for eight months, we got a lot of feedback. Some positive, some negative. We focused on what people loved and basically ignored any »haters«. If we got criticism that showed fundamental misunderstanding of the game, or that was »neutral«, we took that more seriously as a marketing/messaging error, but our #1 goal was to identify what people liked about the idea (in our case, »mythology-building«), and bring it to the front while also making those features our top development priority.

Video Structure: One minute of minimal-text gameplay, followed by three minutes of team interviews. This structure allows viewers to immediately understand the game's possible appeal, while also giving a chance for viewers on the fence to be persuaded by getting to know the people behind the project. We paid a local (very talented) film-maker (thanks Mattias!) to do the team interviews/filming/editing part, and it was money well spent.

Animated gifs always win: Not much to say there. I think we picked a good assortment of quickly-loaded gifs that show different aspects of the game (some familiar, some intriguing), and animation catches the eye.

Lowest possible funding goal: Our goal covered even less than the game we wanted to make – we literally cut everything possible to cut and turned those into stretch goals, such as a fourth biome/environment and digital art book. It was primarily to not overcommit (a good producer always knows what's cuttable), but had the side-effect of making it possible to reach our funding goal that much more quickly.

Stretch goals revealed when emotionally relevant: By keeping our stretch goals mysterious at the start and revealing as we went, we could A) have something new to post an update about and rally the troops, and also B) take the time to double-check our math in how we arrived at those goals, before announcing them.

No playable demo: I know, I know, demos are ideal. But almost nobody expected one. We were fairly transparent in the fact that we were early in development and so we didn't have a demo yet, and people seemed content with that. So I'm glad we didn't spend extra effort hacking something together, and instead used that time to polish our video and page. Note that journalists would probably have been more likely to cover us, if we'd had a demo.

Previous (non-crap) shipped title a definite bonus: Nobody seemed to care that I used to work in MMOs (since Moon Hunters isn't an MMO), but they did care that Kitfox had worked together as a team and released Shattered Planet. Shattered Planet hasn't won any awards, but it has its own loyal following, which is good enough for someone to think, »Well, these Kitfox people have made a game before.« We got explicit comments from multiple backers that our previous title added confidence, even though they hadn't played it.

Few physical rewards: Out of 6.000+ backers, we only have to sort out 320 or so packages, and 400 or so postcards. It'll still be a chore, but more manageable than if we had been looser about giving away extra swag.

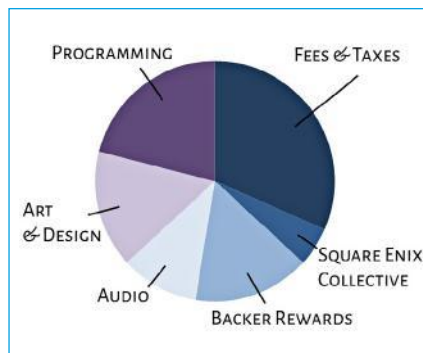
Press Approach: Square Enix generously covered a consultation from Agency, which was a great resource for us in getting a completely unbiased opinion from experts on the Kickstarter page. However, the best learning from them their suggestion to not mention the Kickstarter in our emails to press. We invited press to view our new (gameplay-only) trailer, come meet us at PAX, or talk about Collective if they'd covered it before, but not a single word about our campaign. Again, we didn't have massive press coverage and the coverage we did get didn't seem essential to our success, but every bit counts, and those bits likely needed a chance to get intrigued by the game itself before being possibly put off by the fact that the game was Kickstarting.

What Went Wrong

Promising PlayStation 3: The biggest, most obvious learning was my own ignorance of console development. Without talking to the programmers (due to sudden funding while I was out of town at PAX), I decided to offer »PlayStation 3 and 4 version« as one of our first stretch goals, at 75,000 dollars! When we achieved it, we said any tier with keys entitled you to either Steam or PlayStation keys.

But, as it turns out (as the programmers knew and would have told me if I'd asked) the PlayStation 3 is actually much harder to develop for than the PlayStation 4! Although both are technically supported by Unity, it's nowhere near as simple for 3.

So, two days after we had achieved the »PlayStation 3 and 4 version« stretch goal, we loudly amended it to be only »PlayStation 4 version«, with no PS3 version in the works at



Taxes and fees, backer reward costs and joining the Square Enix collective meant that the team had to set the funding goal at about twice the amount of money needed to fund development.

all. We only got one actual complaint, but a few days later we did get a slew of canceled pledges. Were they related? Probably. Did we look silly? Definitely.

Add-ons are confusing: We didn't want to give away extra T-shirts and posters and other physical rewards to people who didn't actually want them – that would be the height of unnecessary waste, of money and labor both. But many people don't know what an add-on is, and it's a weird process to manually adjust your pledge a certain amount to indicate you want an add-on. We anticipate spending a few weeks just sorting through add-ons, and we'll probably make a few mistakes when we compile all of our mailing lists as a result, but unless Kickstarter implements some kind of »shopping cart« system (which is probably illegal or something), I'm not sure there are any better options.

Undervalued beta access & soundtrack: We're pretty sure people would have paid more than we asked for beta access and copies of the soundtrack ... we erred on the side of having more beta players, but if we had priced them a bit higher, we probably could have achieved a higher average pledge?

Over-eagerness to fulfill rewards: I'm the kind of person that »gets stuff over with« rather than procrastinate ... so I decided I would send out our physical rewards ASAP, right now, before diving into deep production on the game. The problem with this approach is that you can only ever send one survey per backer tier through Kickstarter, ever. So I had to ask people ALL of their information, even choosing what kind of key they wanted, even though there is about a 99.999 per cent chance they will want to change their mind between now and when the game ships. I suspect this will cause way more headaches than it was worth.

Not enough stretch goal research: When I left for PAX on the second day of the campaign, I had no idea we'd be funded while I was gone. If I had, we probably would have done more than just make an initial wish-list of stretch goals with strategically chosen values. We would have, for example, had the programmers go through and sanity-check everything we were promising ... sorry, Mike and Jongwoo!

Platform choice could have been better-researched: We got a lot of questions about our choice of PlayStation 4 and PlayStation Vita, especially from fans of Xbox One and Wii-U. And when they asked why we had made our choices ... well, we didn't have statistics to back it up. We only had gut feelings that Moon Hunters felt like a game that would perform better on PS4, but no evidence or special deals with Sony, which would have been ways to support our position as developers.

Unsolved Mysteries

These things might have gone right or wrong. Without visiting all alternate dimensions, it's impossible to know.



The game was funded three days after the Kickstarter campaign was launched.

Why did we appeal to randos browsing Kickstarter?

We're super-grateful to Kickstarter and all of its channels as a commerce platform, because a full 37 per cent, 66,000 dollars, of our money came from their channels. That's a lot! Probably-maybe more than most! Something we did »clicked« with gamers on Kickstarter specifically – if I had any idea what it was exactly, I'd tell you.

Did running during PAX help? If so, how much?

We launched the day before I left to speak at PAX Dev and manage our booth at PAX Prime, and we were funded while we were there. I definitely met and hugged a few backers, and people who proclaimed proudly to me that they would back when they got home, and also met with a bunch of small press. I have a strong feeling that it's part of why our funding-slope on the chart doesn't flat line in days 4 and 5 ... but ... I have no idea how to quantify how our profile was raised during that time, and how much of it turned into Kickstarter dollars.

Did avoiding a 1 dollar/»goodwill« tier help?

I think it helped make the 10 dollar (actual lowest) tier seem more reasonable/appealing as a psychological »floor« of spending, but I can't be sure. It's worthwhile to note that people can always just fund for 1 dollar if they want to, and 101 did.

What was the effect of very few high-end rewards that max out at 1,500 dollars? Both of our top rewards were taken in the first week of our campaign. Which was sticky – we couldn't introduce any new higher rewards for fear of devaluing our two top supporters, and we didn't have much else to offer anyway. Maybe we would have gotten more if we'd made more room for super-funders?

Should we really have promised 4 platforms, including 2 consoles? As a new studio without

console experience, our promises of PlayStation 4 and PlayStation Vita are brave, if not arrogant. It definitely caught the attention of some fans, and allowed us to get featured on the PlayStation blog, but will we end up disappointing some people? I hope not. Only time will tell.

Did giving away copies of the soundtrack add or reduce its value? The trailer's song was one of its distinctive components. We currently have 4,000 copies of the soundtrack set to be given away, most of which are going to people who didn't explicitly ask for it. Will this raise the profile of our talented composers Ryan Roth and Halina Heron, or will it make the 10 dollar soundtrack price tag seem more outlandish when it releases?

Were T-shirts worth it? Comparing resources spent and received, our T-shirt tier is set to be our least profitable, which we anticipated when we created it. But maybe some of those people are more emotionally invested because they backed at a higher tier, and maybe their loyalty will keep growing as they wear the shirt around and it becomes part of their identity when they wear it with their friends.

What's the right frequency of updates? We updated every 3-5 days, resulting in eleven over the course of the campaign. We took the attitude that we wanted each update to be

meaningful and worth the effort of reading, offering new information. But some people think constant engagement with daily updates (or more!) is more effective, and maybe it is.

Did shout-outs to other Kickstarters help? After the first update, each one contained 2-3 links with pictures to other campaigns that we thought had good audience overlap – games that our backers might be interested in. No idea if it helped us at all, but we're glad we could kick it forward a bit.

Can a campaign page be too long/detailed?

When seeking feedback before the campaign's launch, some other devs commented that we had a LOT of information – lots of pictures, but also lots of text. But this is the kind of complaint that's hard to measure during a campaign, since it could just result in fatigue and not funding – but it's impossible to know why someone doesn't fund. I don't think we had too much, but that's a personal taste question. I'm not even sure too much is actually possible, but if it is ...?

716 people were «upsold» (increased their pledge). Is that a lot? When we reached the Familiars System stretch goal, we devised a new digital-reward goal, to upsell normal backers from 15 dollars (two copies of the game) up to a 20 dollar «Meow Hunter» pledge, in exchange for a limited-edition Black Cat pet and digital

Referrer	Type	# of Pledges	% of Pledged	Pledged
Direct traffic (no referrer information)	External	1868	32.79%	\$58,682.33 CAD
Video Games (Discover)	Kickstarter	985	15.30%	\$27,391 CAD
Search	Kickstarter	297	4.82%	\$8,622 CAD
reddit.com	External	274	4.31%	\$7,713 CAD
google.com	External	228	4.27%	\$7,636.01 CAD
Twitter	External	205	3.35%	\$5,979.50 CAD
Project of the Day	Kickstarter	202	3.84%	\$6,866 CAD
Kickstarter user profiles	Kickstarter	198	3.12%	\$5,580 CAD
48-hour reminder email ?	Kickstarter	193	2.53%	\$4,530 CAD
Facebook	External	191	3.85%	\$6,878.14 CAD
Popular (Discover)	Kickstarter	150	2.38%	\$4,257 CAD
rockpapershotgun.com	External	110	1.41%	\$2,515 CAD
kotaku.com	External	107	1.72%	\$3,077 CAD
Advanced Discovery	Kickstarter	90	1.63%	\$2,912.13 CAD
A project's backer confirmation page ?	Kickstarter	85	1.25%	\$2,244 CAD

The top 15 backer sources were social media driven whereas press coverage didn't have a big impact in comparison.

art book. It was a very good deal, and most of those 700 people were Meow Hunters. But no idea what the average is.

174 people (5,600+ dollars) canceled their pledge, with a high average pledge. Is that a lot? No idea what the average is. But even if all of those 5,600 dollars were PlayStation 3 die-hard fans, upset when we canceled our PS3 support, it was probably worth it for us. The PS3 version would definitely have cost us more than 6,000 dollars.

Backer Sources and Statistics

Our top 15 sources of backers were unexpectedly social media driven. We learned that Kitfox, unlike some other successful Kickstarters, has a primary strength via social media. Part of this is probably the art-driven nature of our game's appeal, but for whatever reason, coverage by major press didn't have as big an impact as our peer-to-peer word of mouth.

Note: Direct Traffic means we don't know where they came from – usually it means following a link from an external program such as an email client, or copying and pasting a link directly into your browser. Our newsletter and Square Enix Collective's email are both lumped in there and presumed to be major factors.

Now, if we re-arrange those top 15 by amount

pledged, we see something different. As you can see, Facebook fans supported us quite a bit more than Twitterers, even though we received more Twitter pledges than Facebook pledges.

In fact, if we divide the pledge-totals by their number-of-pledges, we can see who our most lucrative backers were. In those top 15 sources, Facebook backers contributed an average of 36 dollars, while RockPaperShotgun backers averaged 22.80 dollars – so a full 33 per cent less!

We could retroactively explain this to indicate that readers of major press just want to pre-order the game, while friends-of-friends or Facebook users at large are more likely to have an emotional investment in the team and take more of a personal interest in the project.

But if I'd gotten the opposite results, I'm sure I'd suddenly be explaining that RockPaperShotgun readers are more hardcore gamers opposed to Facebook casuals, which results in more money spent ... suffice to say that I'm generally distrustful of explaining data after the fact. Your guess is as good as mine.

Concrete Lessons Learned

People want to support success stories. The quicker you can talk about some way your campaign is successful, then the more successful your campaign will be. Being funded

Referrer	Type	# of Pledges	% of Pledged	Pledged	▼
Direct traffic (<i>no referrer information</i>)	External	1868	32.79%	\$58,682.33 CAD	
Video Games (Discover)	Kickstarter	985	15.30%	\$27,391 CAD	
Search	Kickstarter	297	4.82%	\$8,622 CAD	
reddit.com	External	274	4.31%	\$7,713 CAD	
google.com	External	228	4.27%	\$7,636.01 CAD	
Facebook	External	191	3.85%	\$6,878.14 CAD	
Project of the Day	Kickstarter	202	3.84%	\$6,866 CAD	
Twitter	External	205	3.35%	\$5,979.50 CAD	
Kickstarter user profiles	Kickstarter	198	3.12%	\$5,580 CAD	
48-hour reminder email 	Kickstarter	193	2.53%	\$4,530 CAD	
Popular (Discover)	Kickstarter	150	2.38%	\$4,257 CAD	
kotaku.com	External	107	1.72%	\$3,077 CAD	
Advanced Discovery	Kickstarter	90	1.63%	\$2,912.13 CAD	
rockpapershotgun.com	External	110	1.41%	\$2,515 CAD	
A project's backer confirmation page 	Kickstarter	85	1.25%	\$2,244 CAD	

Although the Twitter-pledges outnumber the backers that came from Facebook, the latter contributed more money in total.

early changes everything, but even getting to 50 per cent quickly or having a high number of backers quickly can also be brag-worthy, increasing people's excitement to support a success. It's a bit saddening to me as a designer, but it seems like a positive feedback loop – without fail, we were funded more the day after reaching a big stretch goal, not before! Most people want to cash in on high-value products, not help something barely exist.

As a result, Day 1 pledges are worth exponentially more than Day 30 pledges. Do whatever you can to encourage people to pledge as early in the campaign as possible. Feeding into the previous point, snowballs can only happen with backer pledges; potential or future backer pledges don't give you anything to brag about, and don't inspire anyone's confidence.

Kickstarter is a great platform – not just for money! In exchange for the per cent they take, Kickstarter not only automatically cross-promotes your campaign through their various site categories and mechanisms, but you can also use them to contact your backers for no additional fees! Most mailing list services, such as MailChimp or Sendy, charge money for mailing lists with thousands of recipients – but through Kickstarter, we can not only contact our 6.000 backers for free, it also gets automatically publicly archived and displayed! Great value.

Realistic development budgets interest very few people. Initial indie dev feedback on seeing our campaign page was that maybe we didn't have enough details of where the money goes, or that people would be worried 45,000 dollars is not enough money to develop Moon Hunters (and certainly not 22,000 dollars the estimated profit from such a campaign). But not a single one of our backers ever seemed to notice or

care. They trusted us, and our ability to do what we said we would do, as game developers, so that trust is the real win, I imagine.

People like words but don't read. Congrats if you're still reading this – you're not the average Kickstarter backer! We got lots of great comments from backers who were excited about different features, characters, or elements of the game, but they simultaneously betrayed that they hadn't really read most of it. They'd skimmed some parts or outright ignored pieces they did read. The copy on our Kickstarter page was around 2,000 words! The level of detail probably added confidence for backers, but more detail than most customers will engage with willingly ... even when their money is involved.

Summary

We don't yet know what of our experience was unique and what is just indicative of the trends overall. We were Kickstarter siblings with a few other »pixel revival« style games this summer, including the fantastic »Witchmarsh« and »Timespinner«, and were lucky enough to exchange shout-outs with the immensely talented Icepick Lodge rebooting »Pathologic«, as well as a brief overlap with Elegy for a »Dead World and Battle Chef Brigade«. Here's hoping that no matter when you launch your campaign, you're similarly surrounded by supportive, talented devs.

Part of success is being in the right place at the right time, but as we have no way of knowing which windows are closing, all we can do is try to do our best with what we have.

Tanya Short



The pre-alpha gameplay teaser of »Moon Hunters« on YouTube gives a first impression what the finished game will look like.



YOUR KEY TO THE GERMAN GAMES INDUSTRY

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

MAKINGGAMES.BIZ: 10,000 UNIQUE VISITORS
GERMANY'S BIGGEST WEBSITE ABOUT GAME DEVELOPMENT

FACEBOOK.COM/MAKINGGAMES: 9,000 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 80 COMPANY PORTRAITS

TALK TO US!

PROJECTS@MAKINGGAMES.DE

INTERACTIVE GAME MUSIC FOR GAME DESIGNERS

The music in games often consists of elaborate compositions these days, recorded by full orchestras just like in Hollywood. But the acoustics don't always adapt to what's happening in the game. The sound experts of Hamburg-based Periscope Studio have developed a simple solution for this.



Jan Werkmeister
Managing Director
at Periscope Studio

In the 90s, Jan was an active 68000 Assembler programmer. He's been working in music production since 1995, and in 2007 eventually founded Periscope Studios together with two partners. The Hamburg-based studio provides music for computer and console games and develops an audio technology and middleware for the intelligent control of music in games, called *psai*.



Andreas Pongs
Lead Developer of *psai*
at Periscope Studio

Andreas gained first professional experience as a developer in the games industry in 1994 and has been engaging in game audio, multiplayer and artificial intelligence for many years. He's been Lead Developer of *psai* at Periscope Studio since 2010.

On the one hand, interactive music gives the player decisive feedback about the current game situation, such as »Danger ahead«, »Be careful!« or »Well done, all enemies defeated!«, making it an important game-play element. On the other hand, it lets players immerse themselves deeper into the game. If the music is implemented well, it controls the player's emotions like in a thrilling movie, only better since the music is directly linked to the player's actions. Enhanced by the interactive music, players get to experience their »personal movie« in the game.

Let's summarize the goal as »Wouldn't it be great to implement interactive music and thus increase the quality of the game«. But what does it take to reach this goal?

How hard can it be?

When you look closely, you realize just how little the potential of interactive music is actually being tapped even in the latest games. While there is a lot of very well composed game music out there, sadly it is limited to just being tuned in and out, washing away the opportunity of actual interaction – the adaption to the gameplay and the player's actions. How come?

At the end of the day, the simple answer is probably: »Because it's too complicated«. At Periscope Studio, we have worked on interactive music in games since 2007, and the first question a developer would usually ask was: »What's the benefit of interactive music and what does it take to implement it?« We answered the first part of the question above, which most developers agree with; they would love to have interactive music in their game. However, when it comes to implementation, the initial enthusiasm is usually followed by a reality check.

First of all, the benefits of interactive music are hard to demonstrate. Unlike with new graphics technologies, you can't just quickly

look at sound, and it's even more difficult when it comes to interactive music since it only comes to life during gameplay. Thus, the actual benefit is not instantly obvious, only that it's somehow complicated to implement. So, conditions aren't exactly great when it comes to convincing somebody to spend a small amount of the development budget on interactive music. Why invest money in something that basically makes everything just more complicated?

Then again, what does »complicated« even mean? In order to explain that, we need to take a little trip to the production of the very first soundtrack which our then young studio was trying to implement interactively. The first assignment came from Daedalic Entertainment who are known for their lovingly narrated adventures. Point&Click adventures don't really feature the greatest dynamics, and yet the implementation of our interactive music via traditional middleware turned out to be incredibly complex. We wanted to change the mood of the music based on the game progress in just a few scenes within a game prototype, but we had to program a seemingly never ending series of logical operations in order to avoid causing absurd music changes and wrong moods.

The reason for this is that by using traditional tools you basically have to start from scratch and manually program and adapt the music to the gameplay. A simple example (characteristic for an RPG) makes clear what factors need to be taken into account: The player is exploring a vast steppe and eventually gets to a forest where he is attacked by bandits and engaged in a brawl. The player is forced back and the fight is continued in the steppe where the player defeats the bandits.

An appropriate musical implementation could look like this: The background to the steppe is lyrical landscape music while the menacing forest is supported by a slightly darker sound, and as soon as the fighting starts, it turns into action music. With these three

sounds alone you already have to program quite a bit of playback logic. For one, you want the location sounds to smoothly merge since it would surely be regarded as inadequate if the music changed abruptly once the player enters the forest and then again on the way back out of it. In terms of action music, it's a very different story: The music needs to start immediately at the beginning of the fight in order to warn the player of the imminent danger. If the player changes location during the fight, it should be avoided in any case for the steppe music to somehow be triggered again by an event, even though the fight is still in full swing. As soon as it is over, the program should detect that the location has changed in the meantime («Steppe») and not just resume the last piece of music («Dark Forest»). In addition, if you work with regular MP3 pieces, you need to be very careful when creating transitions in order to avoid for the pieces to sound disharmonious or like merged staccato sounds. If you now try to create a musical logic for the entire game by using traditional tools, you will either give up and implement less interactive music, or put an audio team in charge to take care of it. The gist of what Tim Schafer, founder of Double Fine Productions, says in the developer commentary to «Monkey Island 2» – for which they created an interactive soundtrack with their iMuse music system – is: «We had a team of two full-time programmers who were solely working on the changes in music. The result was great, but we wouldn't do it again this way.»

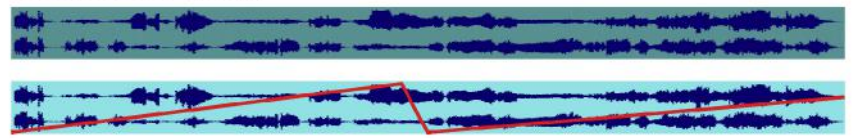
Starting the development process

The situation presented a challenge for us, and based on our idea to create highly adaptive game music like in interactive movies, we began to develop our own technology for interactive music in 2008.

To begin with, we had to ask ourselves which one of the three processes for creating interactive music we wanted to choose for our solution. If you want music to change automatically, you can use the following three processes: interactive mixing, segmentation or generative music.

Interactive mixing plays two or more parallel tracks synched to tempo, an input variable changes the mixing ratio (the volume of the individual tracks). You can, for example, add a percussive action track to a quiet background track, according to what kind of action is happening in the game. The benefit here is obviously the ability to respond quickly: Since both tracks are already synchronized, the »transition« (the changing of the volume levels) can be carried out at any time and promptly. The disadvantage is the very limited possibility of real mood changes. Since all tracks of an interactive music set have to be set up similarly in order to harmonize with each other, no real mood change is possible, only a more intense version of the ones that exist already.

Interactive Mixing



Segmentierung & Rekombination



With interactive mixing the music changes through adding differently sounding, synchronized music tracks while segmentation and recombination create a series of music segments.

The segmentation separates the source music into segments before its interactive use and then recombines it dynamically. The advantage here is a complete freedom of sound; the music can – on demand – change completely from one moment to the other. A drawback is the latency resulting from the fact that a change can usually only be made at the end of the current section (by changing into a differently sounding section).

The third option is the use of generative music. According to predefined musical and formal rules, the music is »recomposed« in real-time using an algorithm. This can be done with sound synthesis or sample-based. While the benefit is maximum interactivity, the downside is high processor strain and the current state of technology: It's not possible to use the generative procedure for all sorts of music yet – the resulting sound is not sufficient enough. This may change in the future and would then open up a whole new chapter for generative music.

How does psai work?

For psai, we decided to use the segmentation procedure, however, we considerably »pimped« this procedure in order to significantly improve the flexibility and adaption to typical gameplay situations.

»How can we keep time and costs low?« has always been the core question while we were developing psai after – as described earlier – we had made very bad experiences conceptualizing and integrating interactive music with traditional tools. For this reason we wanted to make it as easy as possible for developers and game designers by putting the whole complexity of interactive music »under the hood«, so they would only have »to put the pedal to the metal« to get going.

A psai soundtrack consists of psai themes, and each psai theme consists of music separated into individual segments. During gameplay, the psai themes are controlled via very simple trigger scripts. A trigger basically contains two pieces of information; the theme that's to be triggered and the level of intensity (0-100 percent).

Trying out psai for yourself

Webplayer demo

The demo in the example given can be downloaded here and played in the browser (Unity plugin required): www.makinggames.biz/angrybots (or: www.makinggames.biz/psai under »Demos«)

Testing psai in Unity

For your own test trials, you can download psai's Music Engine and the psai Demo Soundtrack used here from the Unity Asset Store and try them out in your own projects.

These two parameters are principally sufficient in order to be able to portray everything the interactive soundtrack needs to »know« about the runtime since the choice of the theme determines the mood while the intensity – as the parameter says – determines its intensity. So, when I trigger a »Dark Menace« theme with 10 percent intensity, you will notice that something's not right. If I trigger the same theme with 100% intensity, ideally fear will be written all over your face. Just to avoid confusion: Obviously, high intensity usually means a fuller music; however, it doesn't always necessarily mean »Action« or »Danger«. When I choose a »Romance« theme, I won't be met with danger at 100%, but rather with an epic and exuberant romantic theme.

Fire & Forget

We designed psai in a way that it resembles the human senses, which results in a unique attribute: When there are no triggers, the intensity level automatically drops until it reaches zero. In this case, psai will only play segments with very low intensity and maybe pause here and there in order to reflect the »There's nothing going on in the game right now« situation. This has the additional advantage that the developers basically don't have to worry about actively »cutting off« a theme. Instead, they simply place triggers in their game world according to the »fire & forget« principle without having to worry about getting on the player's nerves with constant muzak.

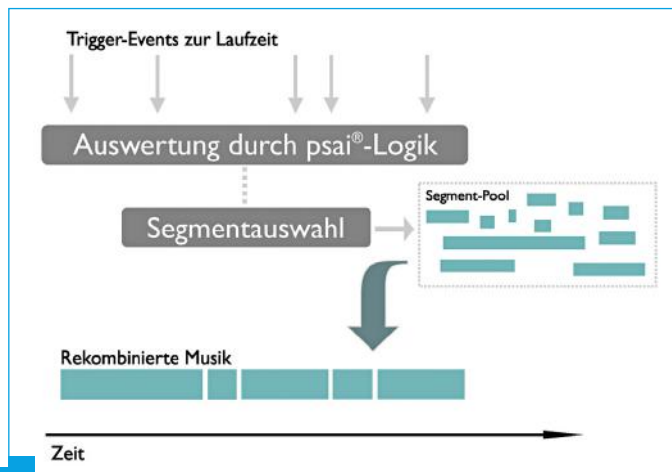
If we were to use psai to add sound for our example above, psai would offer an additional central way to make things easier: Each theme is assigned a certain theme type. We call background music »Basic Mood«, and battle music which is usually supposed to start immediately is assigned the »Action« type. With these simple assignments, psai can decide very easily which theme has priority. We will go into further detail about the theme types in our »Angry Bots« example that's coming up.

First, I would enliven the steppe with triggers of the background theme »Steppe«. I would then place triggers in the forest for the background theme »Dark Forest«. Since the forest is getting more menacing the further the player advances into the forest, the intensity of this theme could be linked to this parameter. The attack by the bandits would be triggered by an action theme, the intensity of which could be linked to the number of bandits, the player's health or something like that.

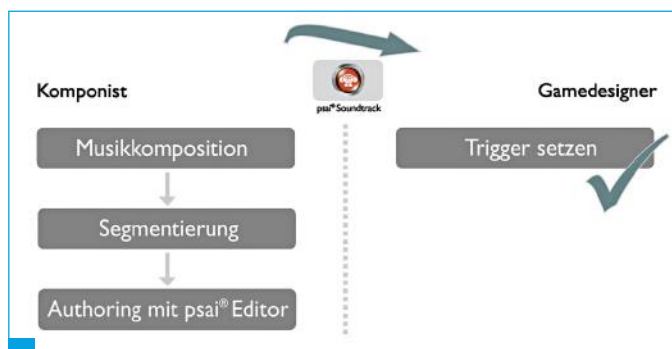
This would be all that game designers would have to do in order to »set up« the soundtrack. Now they could fully focus on their game and do some fine-tuning.

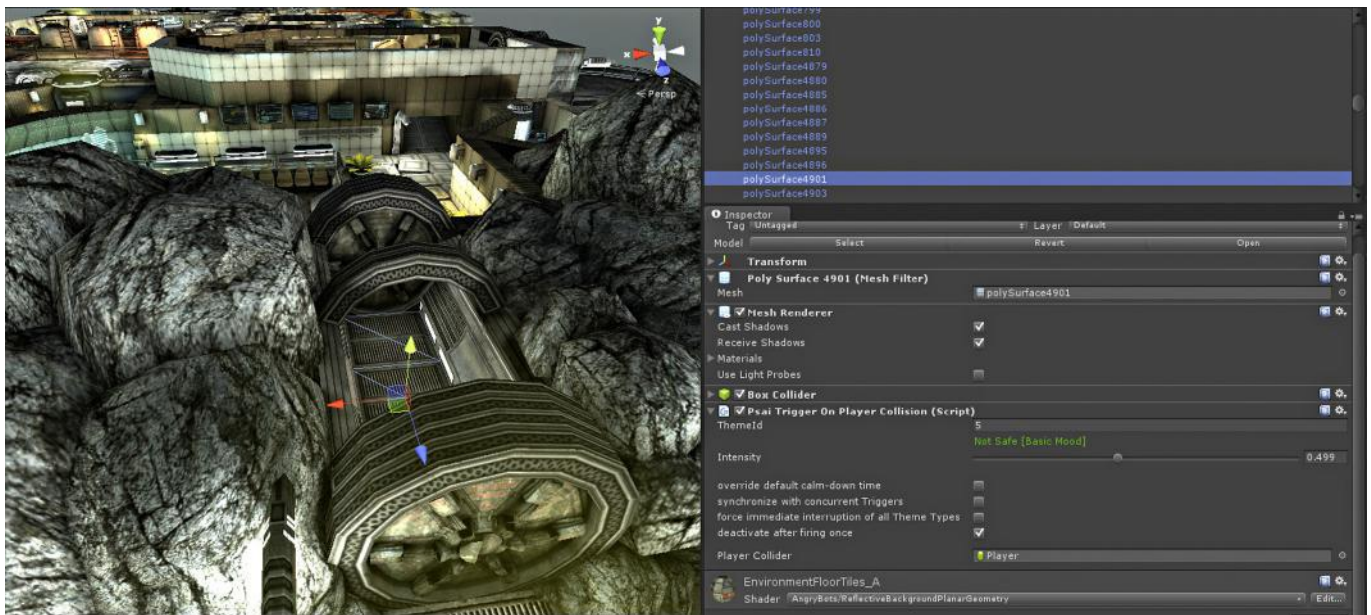
While the game is running, the incoming trigger signals are evaluated by the internal logic and the next adequate segment will be selected for playback. To stick with our example: As long as the player remains in the steppe, psai will keep playing the »Steppe« background music and those kinds of segments that make this music sound as diverse as possible. When there are no more triggers, psai will automatically insert a pause and after a short break start playback again with low intensity, thus enliven the steppe as adequately as possible. As the player enters and advances deeper into the forest, psai will start fading to the dark theme in order to perfectly reflect the situation. The transition from »Steppe« to »Dark Forest« will be soft and musically appropriate, sounding as if everything had been composed just like that. As soon as the player gets attacked, psai will promptly switch to action music; in this case the decisive factor is not the latency or soft transition, but the quick signal that says »Danger«. If the player changes location during the fight, psai will notice the location change due to the triggers placed for »Steppe«; however, it will grant priority to the ongoing fight and only start to softly fade to the steppe theme again once the fight is over. Psai prioritizes and puts things in order. On the one hand, this behavior by psai is due to the theme types which we have designed with regard to characteristic gameplay situations and the human perception. On the other hand, it is due to the fact that psai »intelligently« finds the best way through the number of available audio segments so to speak to reach the musical goal as adequately as possible. There is no manual programming of the sound behavior, even

Psai recombines the music segments via incoming trigger events (simplified illustration).



The chart shows the task split between composer and game designer.





As the player enters the inner part of the station, the basic-mood theme »Not Safe« will be triggered via the TriggerOnPlayerCollision script.

when more complex games and countless triggers are involved, although it wasn't easy during development to define the appropriate behavior of the music for all kinds of »trigger banging«. Through the use of psai in actual projects like »Black Mirror 3« (dtp) or »Mount & Blade 2 – Bannerlord« (Taleworlds), we were able to optimize psai to near perfection.

How do we create the music?

Of course the music needs to be composed and prepared for psai in order to be able to react as described. We realized early during development that we needed to allocate the tasks according to the »Leave it to the pros« principle, so that developers or game designers didn't have to worry about how to best merge these two pieces of music. This is up to the composers who know exactly what their music is supposed to sound like. Likewise it doesn't make sense for a composer to fiddle with the game engine during development; that's up to the developer or the game designer. That's why we have completely separated the creation of interactive music from the implementation of said interactive music into the game.

Before even the first line of code is written, a composer can already create the complete interactive soundtrack based on the story and gameplay, using the psai editor, and test and tweak it without the slightest knowledge of programming. The composer then hands the psai soundtrack to the developer who can implement it into the game without needing any musical knowledge. The soundtrack can now be controlled very easily; the game designer in charge only needs to place triggers at the desired locations in the game and can fully focus on the gameplay without having to worry about any audio programming or the music itself.

Let's recap: We spare the user a big part of the complexity of interactive music and let psai take care of it. Everything else we keep very simple.

Setting the Unity demo »Angry Bots«

In the following, we will pass from theory to practice and set the Unity demo »Angry Bots« to the psai demo soundtrack. This demo soundtrack is already a few years old and at the time was composed for a completely different project. So it isn't a tailored music; it simply seems a bit too »big« for »Angry Bots« with its isometric appearance and its not really threatening looking bots. It's just about explaining the basic principle of scoring with psai and about encouraging developers to try it out with their own project. The psai demo soundtrack and the »8 FREE Music Themes« available at the Unity Asset Store together make 17 themes which you can experiment with just as you wish.

In this example, we have chosen the music changes and intensities in a way that they are particularly noticeable, and it can easily be identified what psai is doing there. If this was a real project, you would work with different music and a more subtle approach in order to benefit the game in the best way possible.

What exactly do we want to add sound to?

We came up with the following crucial features which are also very well suited to be applied to other projects:

- Background music when starting the level
- Action music when the player is attacked by the little bots
- The intensity of the action music adjusts to the threat level and increases the closer an enemy gets to the player; however, with

- weak enemies it won't rise infinitely, only if they attack in groups (intensities add up).
- Dark background music as the player enters the station
- Increasingly dramatic background music when approaching the »Final Boss Bot«, and a powerful effect as the final boss discovers the player for the first time, which, however, won't be constantly repeated when the player sometimes moves out of reach. High to very high intensity action music during the battle with the final boss (depending on the distance and whether the player takes damage).
- Heroic music when the final boss explodes

Which themes and theme types are used?

Even though psai offers a whole range of theme types such as »Dramatic Event«, »Highlight« and more with special behavior, we came to the conclusion that most of the usual game scenes can be reflected with the theme types »Basic Mood« and »Action Event«. »Basic Mood« is suitable for all kinds of background music with a longer playback time, reflecting a basic mood of a region or a setting. »Basic Moods« change in a musically harmonic and soft fashion, providing a breath of fresh air, and also take a break every now and then when nothing is happening for a longer period (no triggers). Battle scenes are best reflected by an »Action Event« which instantly interrupts a »Basic Mood« playback and, with decreasing intensity (no more triggers), returns automatically and in musically harmonic fashion to the last »Basic Mood«. An »Action Event« theme playback, however, will never be interrupted by a »Basic Mood«. In a case where during a battle situation an even stronger opponent breaks through the wall unexpectedly and a further

dramatic surge is required, an »Action Event« may be interrupted by a »Shock Event«. To keep it simple, for this example we won't go into further detail regarding this and other theme types. Just as another reminder: The basic principle of triggering theme and intensity applies, of course, equally to all theme types mentioned.

Psai scripts in Unity

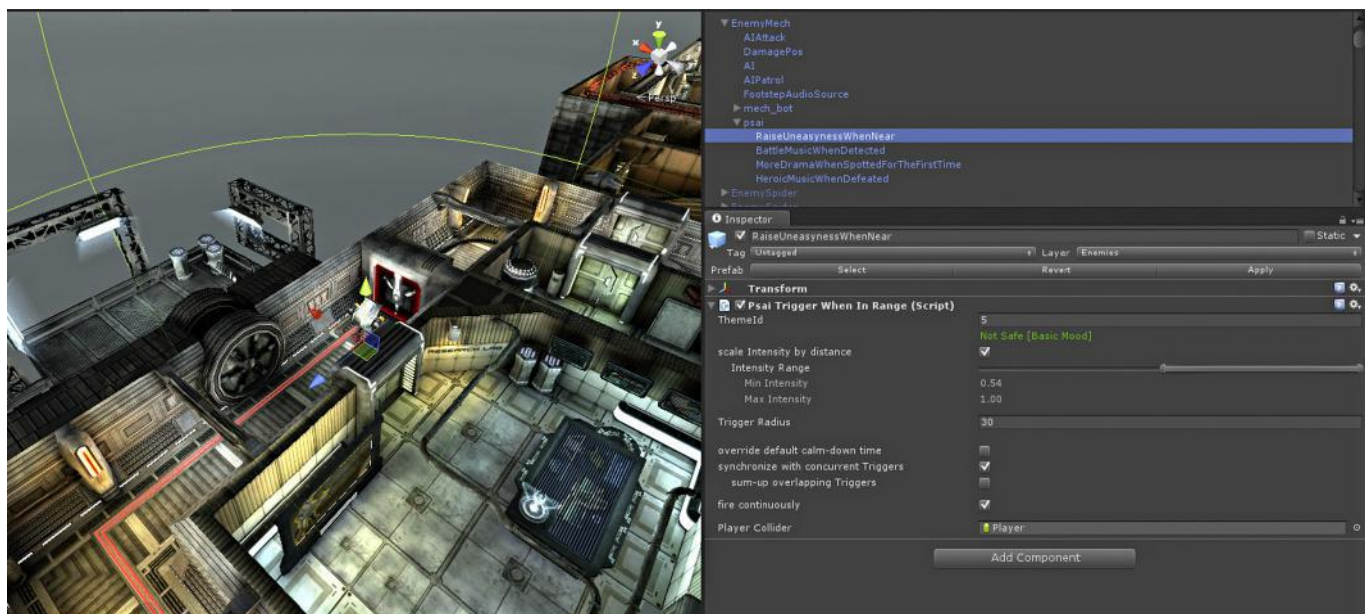
In the last paragraphs we explained that you have to trigger psai themes which the API provides the respective »TriggerMusicTheme« command for. This command could now be used in order to trigger the music via script, however, in Unity, it is much easier than that.

The component-based structure of Unity makes it possible that psai can now be triggered without having to program a single additional line of code. The Unity version of psai contains a whole bunch of scripts which cover the most common application cases of interactive scoring. Via drag & drop, these scripts are simply assigned to the respective game object which is supposed to affect the music.

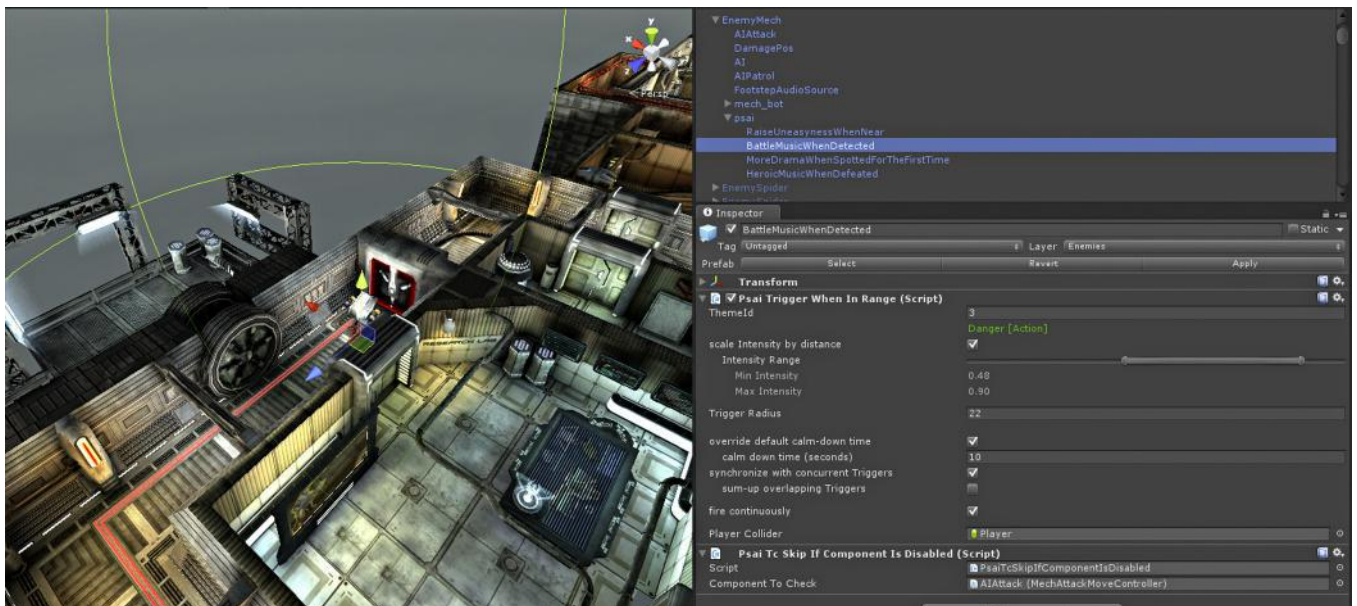
Background music and the attack of the little bots

In order to define the general background music which is to be played right at the start of the level, for example, we create a new empty game object and, via drag & drop, assign it the component »TriggerOnSceneStart«. For each theme there are default settings for the average game duration or for the lengths of the automatic pauses, so that we usually only have to choose the desired theme and the initial intensity.

The same goes for the component »TriggerOnPlayerCollision« which we can always use when the player enters a specific part of



TriggerWhenInRange script: As the player approaches the boss, the intensity of the »Not Safe« theme rises.



»TriggerWhenInRange« script: »Action Event« theme »Danger« is played when the player is near the boss and the condition of having been spotted by said boss is fulfilled.

the game world. The script will then simply be dragged onto the game object with the respective collider like, for example, the floor mesh or a custom-designed trigger plate.

In order to link the intensity of the music to the current threat level in action-packed situations, psai script »TriggerWhenInRange« has proven efficient. It lets you set a trigger radius for each opponent, which – once entered – triggers the script.

The desired intensity of the music can also be controlled via a slider as the opponent enters the outer border of the radius or when the opponent is standing right in front of the player. Since a single opponent on the horizon is only a relatively minor threat as opposed to a bunch of around 20 opponents, we can also initiate via checkbox that the intensities of their trigger events add up.

We also always have the option to set additional trigger impulses for more intense music, for example, when the player takes damage. This can be easily done with the »PsaiOne-ShotTrigger« component; all the game logic has to do is link it to the damage event. The Unity sample project »Angry Bots« uses a simple signal system which activates various other scripts for certain events (»OnDamage«, »OnDie« etc.) which then animate explosions and the like. We simply drag & drop our one-shot trigger onto the list, so that no scripting is required here either.

That way, we gradually equip all suitable game objects with respective trigger components. Especially for battles in real-time games it is normal that there are several trigger scripts going off simultaneously while the game is running. Psai automatically synchronizes, filters and evaluates these trigger impulses through the theme types and trigger component settings. It's also no problem to

have several trigger components per game object. In case of a final boss, for example, we therefore have quite something to offer in terms of scoring.

Enemy bosses

We would like to portray the following behavior:

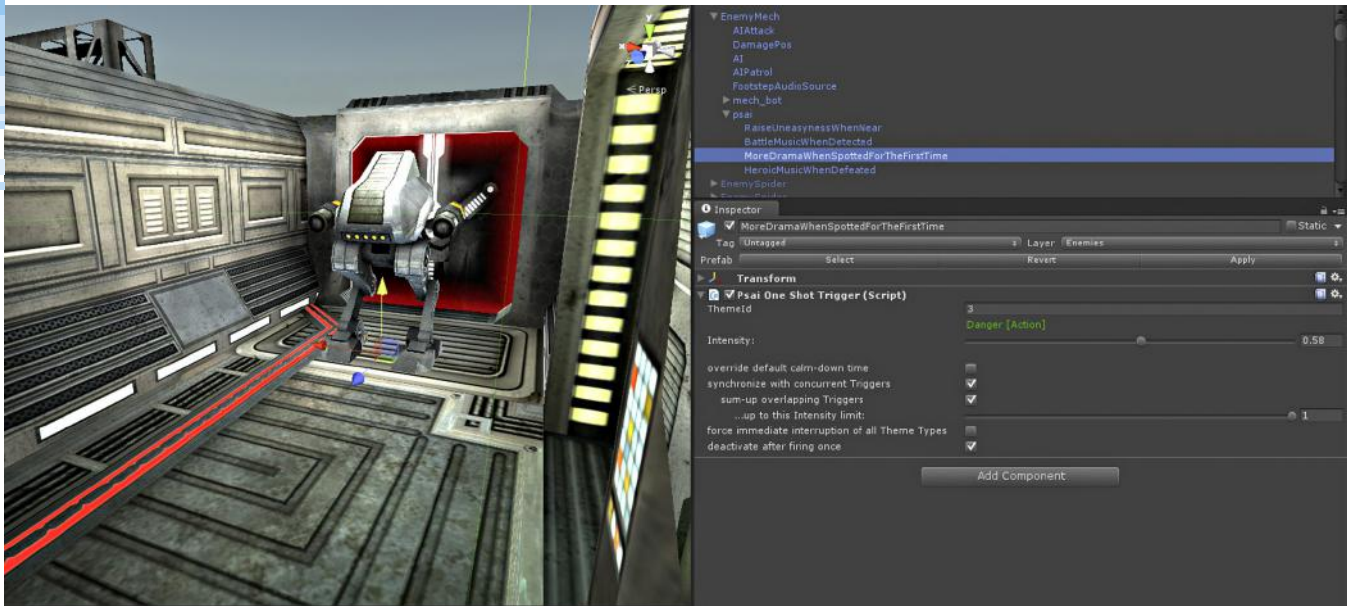
1. Foreboding: The background music slowly gets more dramatic as the player approaches the boss.

First of all, we come up with a name which clearly describes this musical element, like for example »RaiseUneasinessWhenNear«. Below this name we create a new empty game object, which will be in charge of this feature, and attach it to the enemy boss object. That way we keep it clear and flexible, so that in case our soundtrack suddenly doesn't play back as expected, it's easier for us to find the cause through the psai-integrated logging feature.

The actual musical feature is reached via the »TriggerWhenInRange« script which we already mentioned and which we can drag & drop onto our »RaiseUneasinessWhenNear« object. We set our background music as the theme, choose the desired radius and a high to very high trigger intensity.

2. Battle: Battle music sets in as soon as the enemy has spotted us.

We first set a new child node again; this time a good name would be »BattleMusicWhenDetected«. Again, we apply the »TriggerWhenInRange« script which now triggers the battle theme, and adjust the radius roughly to the enemy's field of vision. In order to prevent the battle music from setting in early when the enemy hasn't even spotted us yet, we use a psai component from the »TriggerConditions« folder. »TriggerConditions« can limit the



Drama, Baby! If you're spotted by the boss bot for the very first time, the intensity of the »Danger« action music increases dramatically via the »One-Shot-Trigger«.

conditions at which a trigger is to go off. In our case, we only want to trigger the music when the attack script of the enemy is activated, which we can simply do again via drag & drop: We first drag the »TriggerCondition« called »SkipIfScriptIsDisabled« onto our »BattleMusicWhenDetected« object and then refer the »TriggerCondition« to the attack script of the final boss.

3. Shocker: As an additional shocker we also add a powerful musical effect which is to set in when the player is discovered by the final boss for the very first time.

For this, we use our already mentioned one-shot trigger, choose our battle theme with ultra-high intensity and now activate the »disable after firing once« checkbox. This makes the script go off only once and not, for example, every time we face the enemy again after having escaped earlier. Ideally, we might also add a »TriggerCondition« called »SkipIfNotVisible« if we don't want the effect to be played when the final boss is not even in the picture at the time.

4. Reward: As the enemy boss explodes, there's an immediate change to heroic music.

Another »One-Shot-Trigger« is applied here, which we enter into the list of the scripts that will be activated for the enemy boss' »OnDie« event. Via a checkbox we decide whether we want to start the heroic theme at the same time as the explosion of the final boss, or whether we prefer a musical transition at the end of the current segment.

Conclusion

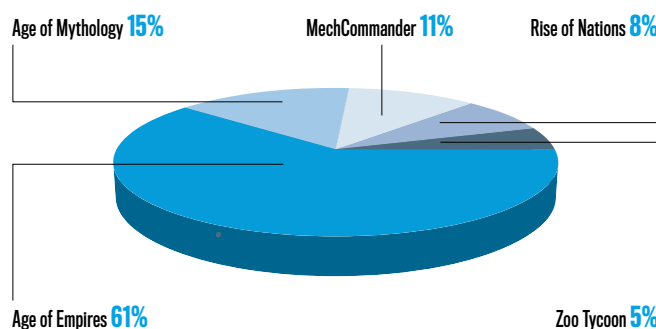
The simplified use of interactive music through psai in combination with Unity's component-based, graphical approach makes it possible to create an interactive music score even without

any programming knowledge. But an even bigger advantage is probably the fact that the parameter tweaking and even the implementation of entire musical elements can be done while the game is running. Some of the psai scripts could also be implemented by programmers using a few lines of code within their own game logic. The question is, however, whether they would have tested the respective elements at all since nobody likes messing around in their source code when there's no need to, not even for the smallest of changes. So, if you can tweak the parameters by a simple mouse click during runtime instead without having to worry about messing up the code, it can lead to an unforeseen eagerness to experiment. **Jan Werkmeister, Andreas Pongs**

DATAFLOW

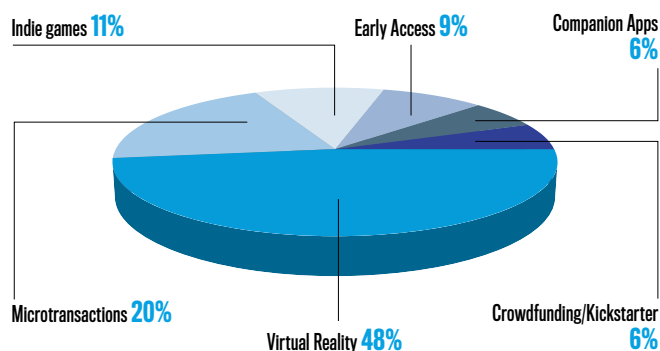
Our consumer magazines GameStar and GamePro are regularly asking their readers and website users about current topics of the games industry. Here's a small excerpt.

Of which popular strategy game by Microsoft would you like to see a sequel?



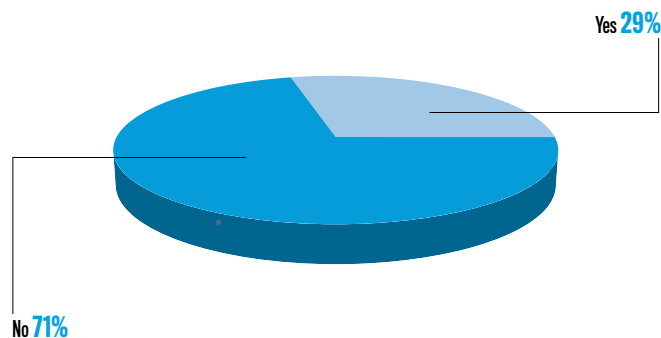
Source: Poll on GameStar.de, 8,680 participants

What do you think is the most important trend in 2015?



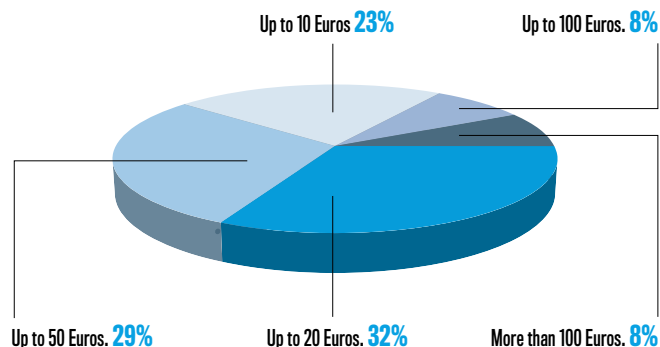
Source: Poll on GamePro.de, 2,056 participants

Do you give away games on christmas?



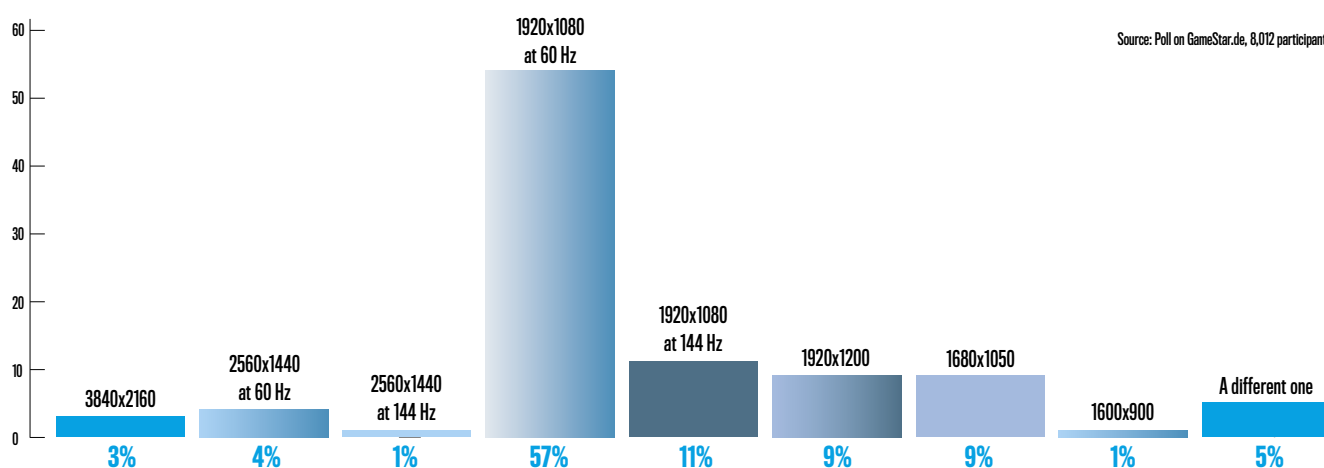
Source: Poll on GameStar.de, 11,918 participants

How much money do you spend during a Steam-Sale?



Source: Poll on GameStar.de, 13,948 participants

Which resolution do you mainly use for games?

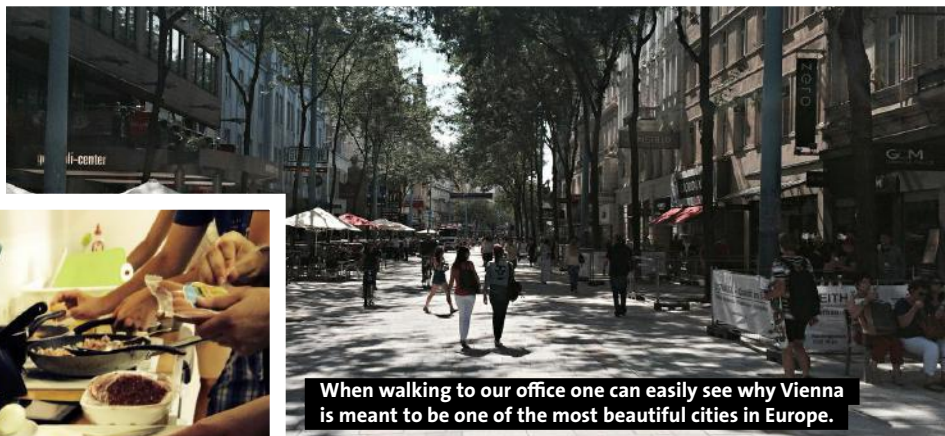


Source: Poll on GameStar.de, 8,012 participants

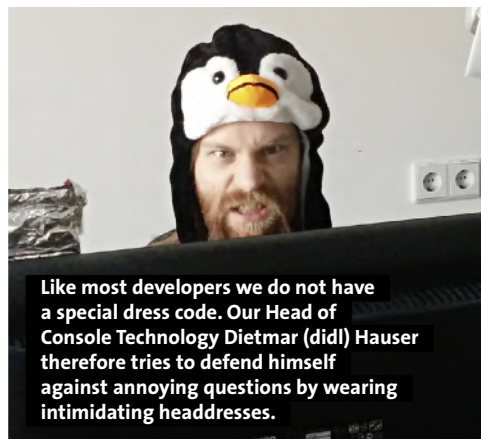
A day at ... SPROING



Our by now well-known logo showing a frog is accompanying us for about 13 years and decorates our entrance.



When walking to our office one can easily see why Vienna is meant to be one of the most beautiful cities in Europe.



Like most developers we do not have a special dress code. Our Head of Console Technology Dietmar (didl) Hauser therefore tries to defend himself against annoying questions by wearing intimidating headdresses.



We can prove that the mythos of devs being used to an unhealthy lifestyle is wrong: In our office we regularly do our own cooking.



Only in big meetings you really realize how many people are working at Sproing (currently about 90).

During the week we have a lot of meetings – it feels like a thousand. This example shows our biggest department – our programmers!



The most recent project we are working on is the RPG-MOBA Sigils: Battle for Raio that will be released for iPad shortly.



That is what our restrooms are called in Viennese dialect – which literally means something like very small house.

HEISL

Before going to work, some of our employees like to do some exercise at the nearby sports ground.



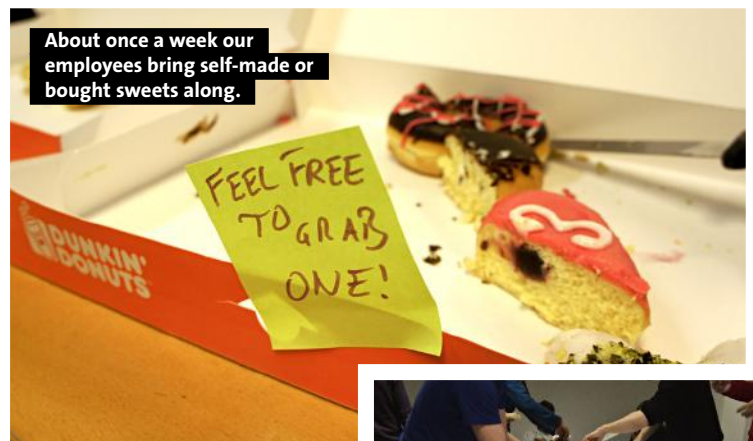
Lead Artist Andrea Mannel is doing some final enhancements on one of the champions from our game Sigils.



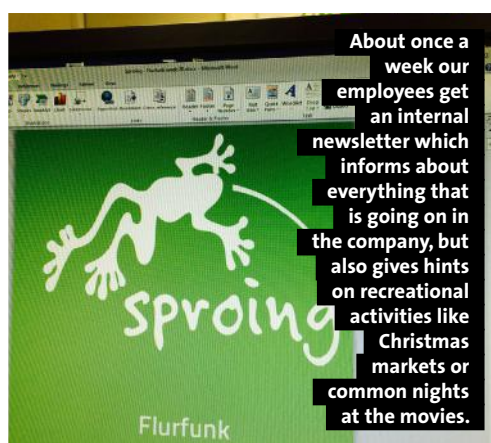
Interesting projects, a lot of mutual activities, a continuously growing team and one of the most beautiful cities in the world - Vienna - are the ingredients for working at Sproing.



Beside of constant visits of journalists, we also take part in campaigns like the so called Daughter's Day, where young girls can get a taste of professions that are mainly dominated by males.



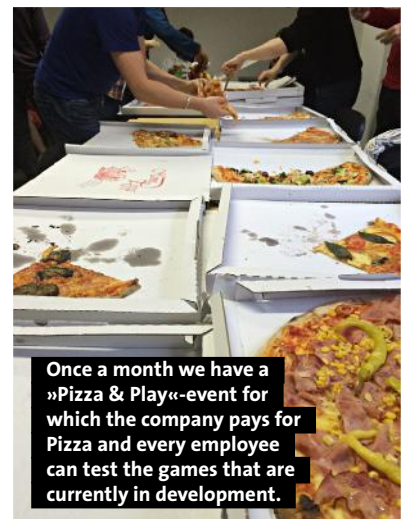
About once a week our employees bring self-made or bought sweets along.



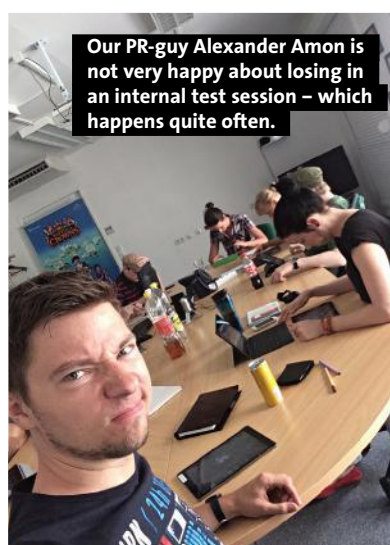
About once a week our employees get an internal newsletter which informs about everything that is going on in the company, but also gives hints on recreational activities like Christmas markets or common nights at the movies.



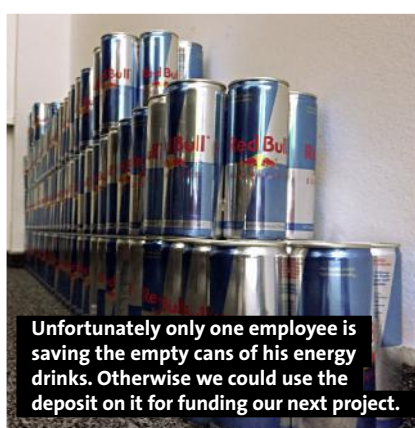
One cliché is right: we all love videogames! Therefore we organize tournaments every once in a while, which we call »Playfrögs«.



Once a month we have a »Pizza & Play«-event for which the company pays for Pizza and every employee can test the games that are currently in development.



Our PR-guy Alexander Amon is not very happy about losing in an internal test session - which happens quite often.



Unfortunately only one employee is saving the empty cans of his energy drinks. Otherwise we could use the deposit on it for funding our next project.



Being the biggest game company in Austria means that we try to inform politicians about what we do. In this example our CEO Harald Riegler shows Federal Minister of Labor Rudolf Hundstorfer (standing right) around our office.

From the roof-deck of our office, which is especially popular amongst smokers, one can enjoy the view across Vienna.



MAKING GAMES

THE NEXT ISSUE

03/2015
MAKING GAMES
WILL BE AVAILABLE ON APRIL 17TH 2015

The Next Generation of Mobile

Games for smartphones and tablets is a profitable market that is growing more and more and that seems almost flooded when it comes to specific game genres. Our experts will explain how they want to be successful by introducing new ways of gameplay, by focusing on specific players or how user generated content will give mobile games a new boost. By means of one example we also will show how updates with fresh content will keep players interested in a game in the long run.

Other Topics

How to write an RPG

Chris Avellone from Obsidian Entertainment shows what is important when designing an immersive role-playing game.

Lords of the Fallen

About six months after the release of their hardcore-role-playing game Deck13 talks about the development and about what went right and what went wrong.

Middle-Earth: Shadow of Mordor

Reviewing the three years lasting development of the Action-RPG in Tolkien's legendary world of Lord of the Rings exposes the mistakes that were taken during this journey.

IMPRINT

Editor-in-chief
Heiko Klinge
(Address see Publisher)

Managing Editor
Sebastian Weber

Editors
Yassin Chakhchoukh
Patricia Geiger
Markus Schwerdtel

Art Director
Sigrun Rüb (Ltd.)

Layout

Manfred Aumaier
Alexander Wagner
Eva Zechmeister
Anita Blockinger
(Freelancer)

Editing (Freelancer)
Marion Schneider

Translation
Bettina Wilding

Contributors

Michael Benda, David DiAngelo, Tiare Feuchtnr, Fabian Fischer,
Thomas Grip, Kevin Maloney, Andreas Pongs, Tanya Short,
Sheridan Thirsk, Jan Werkmeister, Benjamin Wojkuvka

We thank our interview partners:

Jason Chayes, Mike Donats, Victor Kislyi, Karol Zajackowski

Office Management:

Isa Stamp, Anita Thiel, Annie Weissenberger

CONTACT US

Advertising Sales:

Dustin Reichl,
dreichl@idg.de

Address Changes and Subscriptions:
kundenservice@makinggames.de

Article Suggestions and Comments:
editor@makinggames.de

Recruiting- & Projektanfragen:
projects@makinggames.de

Address changes and Subscriptions:
kundenservice@makinggames.de

Company register changes:
info@makinggames.de

Subject: Company register

Ad approval for print issue:
Manfred Aumaier, +49 89 / 360 86-602
anzeigendisoprint@gamestar.de

Ad approval Online:
banner@idg.de

Media rates:
www.idg-eas.de

CUSTOMER SUPPORT

Subscription and single issue orders:
Making Games Kundenservice
ZENIT Pressevertrieb GmbH
Postfach 81 05 80, 70522 Stuttgart, Germany
Tel.: +49 711 / 7252-275
Fax: +49 711 / 7252-377
Austria: Tel.: 01 / 219 55 60
Switzerland: Tel.: 071 / 314 06-15
E-Mail: kundenservice@makinggames.de

Web: www.makinggames.biz/shop

Prices (print):
Single issue: 6,90 Euro
Subscription: 35,40 Euro
(includes 6 issues)
Please contact us for foreign prices and bulk discounts

Payment option:
Postbank Stuttgart, BLZ 600 100 70
Konto-Nr.: 311 704

How to reach the team of Making Games:
IDG Entertainment Media GmbH
Making Games
Lyonel-Feininger-Straße 26
80807 München, Germany
Tel.: +49 89 / 360 86-660
Fax: +49 89 / 360 86-652

DISTRIBUTION TRADE CIRCULATION

MZV GmbH & Co. KG
Ohmstraße 1, 85716 Unterschleißheim, Germany
Telefon: +49 89 / 319 06-0, Fax: -113
E-Mail: info@mzv.de
Web: www.mzv.de

PUBLISHER

IDG Entertainment Media GmbH
Lyonel-Feininger-Straße 26
80807 Munich, Germany
Phone: +49 89 / 360 86-0, Fax: +49 89 / 360 86-501
www.idgmedia.de
Founder: Patrick J. McGovern (1937 - 2014)
Authorized Representative: York von Heimburg
Publishing Director: Frank Maier
Director Sales: Ralf Sattelberger (-730)
Head of CRM/Marketing: Matthias Weber (-154)
Board:
York von Heimburg, Keith Arnot, David Hill
Production Management: Jutta Eckbrecht
Print:
SDV Direct World GmbH
01159 Dresden, Germany

LIABILITY INFORMATION

© Copyright IDG Entertainment Media GmbH

Liability: Despite review neither the editorial team nor the publisher are liable for the validity of any publication. Any paper in Making Games is published without consideration of possible patent protections. Any names of products are used without warranty of free utilization.



QUO VADIS

create . game . business

21-23 April 2015, Berlin

www.qvconf.com



TICKETS ON SALE NOW!

- Quo Vadis Games Industry? Germany's biggest expert convention on games industry topics
- Keynotes about Call of Duty, Mass Effect, Alan Wake and over 110 Workshops
- Over 2.500 Attendees in 2014
- 900 qm Matchmaking Area and Business Center
- New locations: Cafe Moskau and Kino International

15% special Early Bird Discount:

MGP15



All about Quo Vadis 2015: www.qvconf.com

INTERNATIONAL
Games Week
BERLIN

compuTEC
MEDIA

ARUBA
EVENTS



JAN DAVID

Staff Designer
started as
Intern



YAGER

More than 130 people from
20 different countries
working together on the
banks of the river Spree.

Creating awesome games
since 1999.

Join Our Team!

**WWW.YAGER.DE
/CAREER**

LISA
Design
Intern

