03/2016

OFFICIAL PARTNER OF GAME Bundesverband e.V.

making games

DESIGN | BUSINESS | ART | TECHNOLOGY

4 197050 406909 03



THE UBISOFT NETWORK

HOW UBISOFT COORDINATED THE COOPERATION BETWEEN ITS STUDIOS, MAKES INNOVATION PREDICTABLE AND DRIVES TECHNOLOGIES FORWARD

EARLY ACCESS BEST PRACTICE
HOW PLAYER FEEDBACK TURNED THE RPG VICTOR VRAN INSIDE OUT

LEVEL DESIGN ANALYSE
HOW TRAVERSAL LEVELS IN TOMB RAIDER AND CO. ARE ARRANGED

POST MORTEM ON BASTION
WHY PORTING THE ACTION-RPG TO PS VITA FAILED BY A SNATCH

European High Score



„Die gamescom war wie immer großartig und für uns sehr erfolgreich!“

(Category Manager, TEKNOSA, Türkei)

„Die gamescom zeigt weltweite Entertainment-Trends und ist das Tor zum europäischen Markt.“

(CEO, Koch Media, Deutschland)

„Perfekt, großartig, lehrreich – ich kann die gamescom 2016 nicht abwarten“

(Purchasing Direktor, Gaming, Maxitoys, Belgien)

YouTube

17.-21.08.2016 Köln

Jetzt Tickets sichern!

gamescom.de

gamescom

Celebrate the games!

Koelnmesse GmbH
Messeplatz 1, 50679 Köln, Deutschland
Telefon: +49 1806 016 015*, gamescom@visitor.koelnmesse.de

(*0,20 Euro/Anruf aus dem dt. Festnetz; max.0,60 Euro/Anruf aus dem Mobilfunknetz)

BIU Bundesverband Interaktive Unterhaltungssoftware

koelnmesse

THE INDIE AMONG THE BIG PLAYERS?

If we look back, we realize that the indie scene that's so flourishing today is still comparatively young. Back in the days, smaller studios weren't able to publish a game just like that. Usually, they would have to do it via publishers who were very picky, but also supportive in terms of sales and financing, but in return wanted a share of the revenue pie and often interfered with the developer. Then there was Steam with Greenlight and Early Access; crowd-funding became a big hit thanks to Kickstarter; Sony and Microsoft opened up their platforms for more or less everyone; Apple and Google won a huge audience through smartphones and app stores, and ever since then pretty much everyone can try their luck developing games. But we've also seen some kind of prejudice since then: Big publishers only throw big brands like »Call of Duty«, »Battlefield« or »Assassin's Creed« on the market because they know that they will bring in the big bucks, and only independent developers deliver creative and innovative content – to put it briefly and provocatively.

Striving for innovation

That's the thing with prejudice: There may be a grain of truth in it, but this doesn't apply to everybody. The big publishers, too, don't want to follow the same pattern year after year with their studios, but strive for innovation, even if it often only happens in the background, invisible to the public. At least, this is what we learned first-hand from Ubisoft in their Montreal studio, which is why we dedicated this issue to this topic. Obviously, Ubisoft services its big and successful brands, with thousands of people working around the clock, which has its pros and cons (page 34). But at the same time, there are departments like Fun House (page 47) where every employee can put forward game ideas, and »indie teams« then build prototypes – all without any pressure, but with all the freedom it takes for innovative ideas as demonstrated with »Eagle Flight« (page 38). And although everything from game content (page 16) to applied technologies (page 22) is somehow regulated, the French company encourages its employees strongly to share their knowledge, provide feedback and support each other, and for that they even organize an annual internal developer conference (page 14). Of course, this is far from being a classic indie, but in terms of the fundamental attitude and the enthusiasm of the developers there actually isn't too much of a difference.

> »The big publishers, too, don't want to follow the same pattern year after year with their studios.«

Enjoy the new issue!

Your Making Games Team

Sebastian Weber
is Managing Editor of Making Games Magazin.



Good-bye, Patricia!



We sent the issue 02/16 to the printer in early February, so that it would reach our readers in time. And almost at the same time we learned about the rare occasion of one of the Making Games people leaving the team. Patricia announced her good-bye after she had uploaded the last PDF to the print house server. Patricia had been a member of the small Making Games family since December 2013. She worked on magazine articles, kept contacting many of you to make sure we were able to introduce exciting tools every month, did a lot of organizational work in the background at every Making Games Talents, and always lent an ear to each of our authors.

Patricia stays with the company, but is now looking after a different project. So we say thank you for two years of strong commitment, and all the best for the new job!





ON-SITE UBISOFT DEVELOPERS CONFERENCE

For six years now Ubisoft has been hosting an annual internal GDC-like conference. In 2016, external visitors were allowed for the first time – and we didn't want to miss the chance to look behind the scenes at Ubisoft and their Montreal studio.

Cover story content

The cover story revolving around UDC differs a little from the typical cover stories we usually feature in our magazine. Since we had the opportunity to talk to a lot of developers in Montreal, this issue mainly consists of interviews covering the following topics:

- **Page 06**: Interview with the head of the Editorial Team managing all of Ubisoft's productions.
- **Page 12**: Interview with the head of the Technology Group who are the masters of all of Ubisoft's in-house tools and programs.
- **Page 16**: Interview with the Director of User Experience who makes sure that the teams can work more effectively.
- **Page 20**: Interview about Artificial Intelligence.
- **Page 24**: Interview with an employee at Ubisoft Singapore who explains how a AAA production developed around the globe works.
- **Page 28**: Interview about Eagle Flight and the new medium VR.
- **Page 32**: Interview with Alex Parizeau, Managing Director at Ubisoft Toronto.
- **Page 37**: Fun House – Ubisoft's laboratory for game concepts.

You may be wondering why our cover story isn't about how to make it in upcoming markets like Russia, South America or China as previously announced, and what you get instead now is a long series of interviews with Ubisoft developers. But when we finalized the last issue in early February, we didn't know yet if we would be travelling to Montreal to attend Ubisoft's developer conference »UDC«. But since the French publisher now, on the occasion of the 7th anniversary of the event, decided to welcome press for the first time, we didn't want to miss the chance to give you insight into the daily work procedures of a global industry giant.

UDC? What's that?

So it happened that in late February we travelled to Ubisoft's biggest studio in Montreal where the Ubisoft Developers Conference, short UDC, took place. The internal conference was first hosted in 2010 and has grown significantly ever since. While there were only a handful of presentations six years ago, the number increased to 60 in 2016. The number of submitted presentations, too, increased from 76 to over 170. The basic procedure of UDC can be compared with the Game Developers Conference. A Ubisoft-internal committee collects speech submissions, evaluates and selects them, making sure there is a good mixture of all areas of operation. On top of the presentations there are seminars, discussion panels and an exhibition floor where new in-house tools are presented.

The focus of the event is on sharing knowledge between the teams – something that's deeply rooted in Ubisoft's DNA as we're told over and over again when talking to Ubisoft employees. Although Ubisoft owns dozens of studios all over the world, the various teams are working together, learning from one another, inspiring each other and – in case of the people working in Montreal – taking the new experience home and discussing them with their office peers.

On-site, us press representatives were allowed to attend two special presentations. One

Photos: © Photo Thierry du BosCosmos ImageUbisoft





Ubisoft puts great emphasis on motivating its employees. Every game that's released officially gets a place on the Wall of Fame, and awards are displayed prominently.

was about systemic AI models (see interview on page 20), encouraging the various teams across the globe to reflect upon the problems that open-world games create for AI programming. The other presentation was about a new animation technique. Just like at GDC, people in the audience can ask questions after every presentation or discuss it with the speaker – and not just the people on-site. Since all presentations are streamed live to all internal locations across the globe, Ubisoft employees in Germany, France, China and everywhere else can give their input via chat. Employees also have the possibility to access the recorded presentations through an internal portal anytime they like.

Ubisoft Montreal – the game factory

The fact that Ubisoft has always hosted UDC in Montreal and not in the Paris headquarters is only logical. The biggest studio in the world has approximately 2,700 employees working on AAA projects and experimental titles across four floors, which the outside world is sometimes not aware of. So without much travel expenses, UDC is likely to have a lot of attendees and experts for presentations on-site.

During an extensive studio tour we also learned that on each of the four floors one team works on a big project, with each team consisting of at least 300 to 400 people – with the addition of the other global locations that work with them. In addition to these development groups who are fully focused on one title only, there are cross-project departments like sound, playtesting or motion capturing which support all projects. The sound department, e.g., includes 50 people who are in charge of speech recordings for all planned localizations and the sound effects as well as mixing for the final game. For example, the mixing of the intro sequence for »Far Cry Primal« alone took three working days. Speaking of sound effects: The team has its own Foley Artist whose workplace looks like a crazy workshop stuffed with all sorts of props which he sometimes uses for completely different purposes in order to record any sound he wants. Again, the Primal intro serves as an example, for which the artist recorded and mixed about 50 to 60 different sounds. It's impressive to look at, but it's even more impressive to see the creativity behind it.

Obligation of confidentiality

Ubisoft Montreal is a fascinating place. The big old factory building is extremely charming and this hive of developers demonstrates within minutes how much effort it takes these days to make a AAA game. At the same time it was interesting to see how much focus Ubisoft puts on the collaboration of their international teams instead of having them do their own thing like at a lot of other publishers. At UDC you constantly see groups of people everywhere, engaging in discussions or listening to presentations held by their peers. The idea of an internal developer conference works out perfectly since it encourages people to share their experience in an easy, informal way. After all, UDC's motto is »What happens at UDC, stays at UDC«. It's prohibited for the attendees to write anything about it on Facebook, Twitter or anywhere else. Hence, we aren't allowed either to write about some things we learned about, saw or overheard because people at UDC talk about a lot of subjects that haven't been announced in public. And yet we're sure that the following interviews will give you fascinating insight into the way Ubisoft thinks and works.

Sebastian Weber







In terms of the event organization and content, UDC is following GDC, so that the internal conference is packed with presentations, seminars, discussion panels and exhibition floors.



The sound department uses this console to mix every game on-site. Sometimes an intro sequence alone can take up to three days of work.



Foley Artist Tchae Measroch records all kinds of sound effects for Ubisoft's games, from swords to flames, mixing them as needed for each game.



Ubisoft Montreal has several playtesting labs where external, supervised players test new games during development. Ubisoft has registered more than 15,000 potential testers in Greater Montreal.



»WE ALWAYS TRY TO ADOPT THE PLAYER'S PERSPECTIVE«

The Editorial Team is one of Ubisoft's internal departments, making sure all productions follow a certain procedure. Team lead Olivier Daube explains which rules they have to follow, how these rules are set up and how the collaboration with the studios works.



Olivier Dauba
VP Editorial
at Ubisoft

Olivier joined Ubisoft in 1998 after having built simulations for the European Space Agency. He worked as a Gameplay Programmer, Producer and Creative Director on several titles including Rayman 2 and 3, XIII, Splinter Cell Pandora Tomorrow, and all previous gen Ghost Recon games. After a break teaching and researching in academia, Olivier joined the Editorial Team at Ubisoft's international HQ in 2012 where he is now a VP.

Making Games Can you explain in brief what an Editorial Team does?
Olivier Dauba We have a Chief Creative Officer, Serge Hascoet, and the Editorial Team which creates a certain set of guidelines for every single project we do, making sure that the teams stick to these guidelines. This is a very brief description of what we do, and I don't think other developers have a department of this kind. Publishers usually do this kind of thing on a studio level, but our company does it on a global level across all studios.

Making Games In what way are you separated from departments like marketing? Usually, the word »editorial« has a different meaning.
Olivier Dauba We're actually completely separated. We work together on some topics, but our competence lies mostly in the content of the games. For example, just this morning I discussed our vision of open world with some people as this is a set of guidelines that we have defined: that we communicate with regard to what we expect from our open-world games. Usually, this includes different presentations, analyses and things like that. We also travel a lot to the various studios to communicate these guidelines to the teams. With regard to our day-to-day work, it's mostly the game designers from our team who are in touch a lot with the game directors and designers from the individual studios to give them feedback on the content of their games.

Making Games So your job is mainly focused on internal affairs and not so much meant for external parties?
Olivier Dauba That's right, we don't have much contact to external parties. Our job is strictly internal.

Making Games Is the evaluation of games, for example, also part of your job?
Olivier Dauba Yes, this is also part of our job. And we also do content reviews and help the teams to set milestones. Every single project goes through a so-called stage-gate process where the project is being reviewed at various stages of development. We look at the initial concept or later at the first prototype. In the follow up, we help the teams to define their target; we basically assist in planning every step of the development.

Photos: © Photo Thierry du Bois/Cosmos Image/Ubisoft



Ubisoft puts great emphasis on authenticity in their games. The team behind Ghost Recon: Wildlands traveled all the way to Bolivia in order to get to know the country and its people.

Making Games Can you give a specific example of what kind of guidelines you set for the studios?

Olivier Dauba These guidelines can include, for example, the way the game world is supposed to be set up in an open-world project or – a bit more specific – how a user interface should work, but also things like which fonts may be used, etc. So our guidelines basically refer to everything, from the vision of a game all the way down to the smallest details and practical recommendations.

Making Games How do you actually set these guidelines? Are they based on what the players want, or do you think more in terms of marketing and what the teams are actually able to implement?

Olivier Dauba We always try to look at things from the player's perspective, so the main focus is on accessibility. We want the players to master every aspect of the game, so they should be able to easily pick it up, and we want to make it as easy as possible for them to learn the rules of the game or of the virtual world. Today, we're moving away from the era of very linear single-player games to games where the player is more or less free to experiment and to try out things, and that's really something we try to push and promote in our games. Our aim is for the players to always have an understanding of what they can achieve in the game, to make sure they have an understanding of the usability of the different tools that we give them and to make sure that with this understanding they can come up with their own way to play.

If we go back 15 years, we made games where there was only one way to play the games. Today, we are used to making games that are much more intelligent and ultimately the players should feel intelligent by what they do; it should not feel like they're just doing what a designer intended them to do.

> »Today, we are used to making games that are much more intelligent than 10 to 15 years ago.«

Making Games How do you make sure you get the balance right between pleasing your existing audience and newcomers who need to be taught how to play a game?

Olivier Dauba That's a very hard task for sure, especially as we reach more and more players.



Ubisoft is constantly analyzing if game content has already been there before in order to avoid having to use tutorials over and over again.

We are always looking at ways of using data to analyze what the player did in previous games. So if you already know how to ride a bicycle and I would want to teach you how to ride it again, you may not be very happy about that. That's why we use reminders, things like setting up a fairly easy biking challenge just to see if you actually remember it. I think that's a more elegant way than just to use tutorials over and over again.

Making Games So you are using a rating system where players can rate missions in order to collect data for future projects?

Olivier Dauba When a project is in development, we use a lot of data sources. For example, we are tracking data regarding how the player is actually playing the game. We get this kind of ratings and feedback you mentioned, whether it is included in the game itself or through external interviews with players. Then we try to use the data from a game to improve the next one. For example, when a specific type of mission gets a lower rating than another one, you will probably find less of those in the next game, or depending on what our own analysis is telling us, we may want to tweak it differently. The fact that players don't like something doesn't necessarily mean that they don't want it; it may just mean that it isn't tweaked well enough, and we may want to improve it.

»We're moving away from the era of very linear single-player games.«

Making Games You mentioned earlier that you try to look at things from the player's perspective. How do you proceed if your approach based on this perspective is contradictory to what the development teams have in mind?

Olivier Dauba A lot of the work we do with the producers is about setting up a road map and trying to prioritize things as we're well aware that every team has to live with constraints and deadlines and that we therefore can't allocate twenty different tasks.

Making Games Do you sometimes have disputes with a team planning to do something that doesn't comply with your guidelines, e.g., when it comes to nudity in games or things like that?

Olivier Dauba Of course we sometimes have disputes. They're not really about things like nudity or that kind of content; here we try to follow the guidelines of PEGI, etc. We always try not to censor anything in games, but we don't want the teams to do anything crazy. Instead we want them to do things that make sense in the context of the world they are creating. So if a team is making a game about World War II, for example, there are certain things during that period of time that may make some people uncomfortable. But if it happened in the story's period of time they are recreating, it has to be in the game. Assassin's Creed Unity was an interesting example. It's basically about the French Revolution, and especially in France people prefer to downplay or tweak that part of their history. So the game was probably a bit more historically accurate than what the French learn in school. But that was what the world was like back then, so we didn't try to censor that, but rather tried to depict it as what it was.

Making Games In an example like this, does the Editorial Team encourage the studios to be historically accurate?

Olivier Dauba Typically, the ambition of being historically accurate is pretty well engrained in the teams. I think the teams have now absorbed everything they needed to know to be able to achieve the best result from a historical point of view. If you take a team like the one around Assassin's Creed, they have their own pool of historians on the production floor, so



In Assassin's Creed: Black Flag, Ubisoft used a mission rating system for the first time to give the developers feedback on which tasks in the game needed to be tweaked or even removed if the players didn't like them.

those guys are actually the ones making sure that everything is extremely accurate.

A part of our Editorial team is in fact in charge of making sure the team has every input they can possibly have in order to be super accurate. They gather every type of expert you can imagine for a production team and they also organize field trips. For Far Cry 4, for example, they organized such a field trip. Surprisingly, they managed to meet somebody from the French embassy in Nepal who was very much into ornithology. During weekends, this guy would be traveling around the countryside and record every possible bird sound you can find in the area. Because of him we got access to tons of bird tweets from Nepal, which we integrated into the game.

The team behind Ghost Recon, to use a more recent example, traveled to Bolivia to get to know the country and its people. When said team then returned from the trip, they had a much better understanding of what was happening in Bolivia and the way of life there. These assets are then taken and fed into the so-called world-texture tool to make all the information they gathered available not only to the maybe 15 people who took the trip, but to the entire team.





Although the Editorial Team presents the team with a lot of guidelines, they try to avoid censorship at all cost. Nudity or sex scenes like in Far Cry 4 (top) or historical accuracy like in Assassin's Creed Unity (bottom) are even requested (even if they may be unpleasant from the French point of view) as long as it makes sense for the game world.

Making Games Who do you talk to when organizing a field trip? Do you have contacts to the embassies or the foreign governments?
Olivier Dauba Basically, we can talk to anybody. The job of the team who deals with that is to build a network of experts. Some of the guys on the team are spending their lives on the phone making contacts, but at the same time they are trying to organize conferences or seminars as well, not only field trips. So if a game is dealing with a specific topic, we try to identify the best specialists and invite them, and those guys are usually eager to share their knowledge. In some respect, we regard our creative people a bit as embedded journalists who go to some place, grab everything that is relevant for the game world, prepare it and give it back to the team and ultimately to the fans and players. It's not necessarily about a big vision behind the game; it's more about identifying everything that makes sense for the game and creates a memorable experience.

Making Games The guidelines you have for the studios – is there already a set of guidelines in terms of VR?
Olivier Dauba VR is giving us the opportunity to be pioneers again. It's a bit like doing things again for the first time and re-learning everything that we know, like starting from scratch. We have to learn a lot about how people feel in VR. There is a lot of inconsistency in VR regarding what the various senses feel and what the experience can make you feel. VR right now is about doing a lot of user testing, getting tons of user feedback regarding what works and what doesn't, and that's in some respect what makes it so exciting. What we did in the past on traditional games doesn't automatically work in VR – everything we know about camera management, for example, doesn't work the same in VR. In VR, the virtual camera is basically the head of the user, so if you try to move it against the user's will, it will turn out to be extremely unpleasant for the user. So a lot of what we do regarding cinematography doesn't work; we have to re-learn and find a new way to narrate something in VR.

> »VR is giving us the opportunity to be pioneers again.«

Making Games How much more testing do you have to do for VR compared to traditional games?
Olivier Dauba Comparing the amount of testing might not work very well because the



For Far Cry 4, members of the team traveled to Nepal to gain some impressions. In the end, an employee of the French embassy delivered tons of bird tweets for the game, which he had recorded himself.

scale of the games is different. What we do in VR is a bit simpler than what we do in an Assassin's Creed, for example. When it comes to the frequency of the tests though, it is something that we do a lot more. Every single developer is testing something in VR as well. The first people who tried VR at Ubisoft actually had a pretty painful experience – I did some of that testing myself. Even when it comes to being able to work on a VR project, we set ourselves guidelines regarding the amount of time you should be using your headset, how you should switch from your headset to your workstation, etc. We also work with experts from universities as well. VR may be new for the mass market, but it has been around in other industries for about twenty years.

Making Games Why is Ubisoft taking the risk to bet on VR? There is no guarantee of how the users will adopt it, so surely it would be easier to wait a few years until there is a real market for it and then refer to the experience of other developers?
Olivier Dauba We are taking very measured steps. We want to grab this kind of knowledge while the opportunity is still right in front of us.

Making Games So it's important for you to show that Ubisoft still takes risks and doesn't rely solely on the big brands?
Olivier Dauba Indeed, we are still a creative company. And we still want to show that people who want to be creative should come and work with us. We believe that VR and AR will have a significant impact on entertainment.

> »We are still a creative company, and we always want to show that.«

Making Games What is causing the biggest problems for you regarding VR?
Olivier Dauba Movement is still a bit difficult, for example. I think VR can be compared to what the situation was like with mobile games. When smartphones were initially released, people tried to recreate the same types of games they were used to make on other platforms. So initially on iOS you had games with a virtual joystick which didn't work properly, and the same thing could now happen with VR. With VR you have to understand what the system lets you do, e.g. that movement is limited as you are tracked in a place that's smaller than your living room and this is, of course, something you have to take into account for your game design. That's why being able to learn about VR early on is so crucial for us. A game with free-roaming movement and a third-person camera might not be the best idea for VR; it may be fairly easy to do actually, but we don't know yet. On the other hand, we're seeing really interesting experiments with exciting concepts like »Werewolves Within« which we announced at GDC. It's a game that's very social and something that benefits heavily from VR. Something that's innovative and pushing VR, but also AR, significantly. At the moment, engaging in VR feels a bit like the Wild West of game development.

Making Games This sounds as if traditional games are going to be more and more open-world games, while VR is still very limited in this respect.
Olivier Dauba There are definitely limitations for VR, but when we talk about limitations, it's

more about finding solutions for that, which can be very exciting. Direct movement may not be the right solution, so some teams come up with ways of teleporting the player from one place to another. Systems like this could work very well. If you have an open world, you probably have a map, too, on which you could have the player to decide which place to go to. A bit like Google Maps and Street View. The challenge might not be movement itself. In Assassin's Creed or Far Cry there is usually a challenge connected to moving from one place to another, but in VR it may not be about moving, but more about where you stop or teleport to and have a challenge there.

Making Games To come back to the guidelines again you set for the teams: How do you proceed with a new medium like VR?
Olivier Dauba We're usually among the first ones to try out stuff like this. We do some of those experiments ourselves, and for VR we did some of the first prototypes as we have some programmers on the team. That way we can take some of the risk away from our production teams by showcasing them our prototypes and sharing our experience with them.

Making Games How do these guidelines look like in the end? Is it a kind of checklist for the teams to go over?
Olivier Dauba Yes, basically. There are many different guidelines regarding the type of topic we are discussing. For example, when it comes to PC we have a set of guidelines that are more about the types of features players are expecting in a PC game. We compile this list and use it as a review tool. Of course we talk to the producer of the project and together we try to assess what makes sense for the game and what doesn't, or what else we could include from our guidelines. This way to do it has really paid off in the last few months, especially for our PC games.

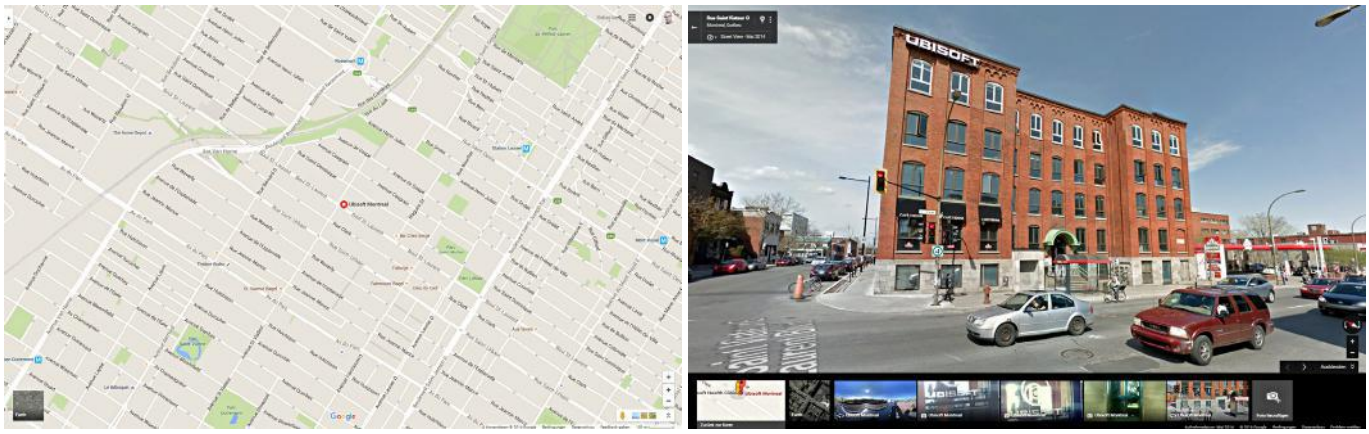
When it comes to content we have guidelines regarding animation and inputs, for example. We believe that a player input should always overrule an animation. So if I am pressing a button, I'm expecting a reaction and that reaction should happen; I shouldn't have to expect an animation to finish first. We may have guidelines related to how soon I should be in the core of the experience. If I was making a game about pirates, for example, the player should be able to meet the first pirates within the first 15 minutes of the game or so. Otherwise the player might get the feeling of not playing the expected fantasy and that could be frustrating. The same thing may be about how long the matchmaking etc. should take for a multiplayer game.

> »The fact that players don't like something doesn't necessarily mean that they don't want it.«

Those are the types of rules we have and some of them are based on experience, some of them come from best practices, either from some of our own games or even from competitors. Not everything is based on our own imagination, we do a lot of analytical work.

Making Games Does that mean that basically any team can provide input for those guidelines?
Olivier Dauba Exactly. Every single team usually tries to push the envelope one way or the other, so they're coming up with new procedures all the time. The benefit of our position is that we learn about something new, and since we're in touch with everybody, we can broadcast it to all the other teams around the globe. We have our own internal social network and sharing platform where we can promote presentations.

Interview: Sebastian Weber



A lot is going on at Ubisoft in terms of VR at the moment, with movement in the game world being one of the trickiest issues. Creating a map of the virtual environment for teleporting like Google Maps and Street View may be one way to solve this problem.



»WE HAVE TO ACCEPT CHANGE, OR ELSE WE'RE DONE«

The so-called Technology Group is the guardian of all technology within the Ubisoft group. The head of the group, Chadi Lebbos, explains which benefits a centralized system like this brings to the company.



Chadi Lebbos
Technology Group Director at Ubisoft Montreal

Chadi joined Ubisoft Montreal in 1997 when the studio was home to just 100 team members. He started his career as Localization Manager, before spending over a decade as Producer and Executive Producer working particularly on the Tom Clancy's Rainbow 6 franchise. Since 2013, Chadi has been working as Technology Group Director, ensuring the sharing, development and optimization of internal tools and middleware.

Making Games Can you explain briefly what exactly the Technology Group is?
Chadi Lebbos The Technology Group is mainly responsible for providing technical solutions for our productions, be it AAA titles, mobile or externally developed games. We cover almost all areas, so the teams can focus on what they do best: making the game. And we're taking care of what's underneath. This may include, for example, AI, animation or physics modules.

Making Games How has your job developed or changed over the last few years?
Chadi Lebbos We're always changing and evolving, and we have no choice than to adapt, or else we're done. So the Technology Group's mandate is first to collaborate. We're not just a supplier of technical solutions which we provide and then our job is done. Actually, we co-develop the programs and tools together with the production teams. Once we're finished with a new technology, we add it to our catalogue which every studio has access to. So basically we're in touch with everybody in the Ubisoft universe, pay a lot of visits to the studios and help out here and there, sometimes even for months at a time.

Making Games Do you also advise the teams or even sell them a certain tool or so when you think it would benefit the production?
Chadi Lebbos Selling is not really the word for it. What we have is a process for every project we call the »stage gate«. At the beginning of a production we sit down with the team to see what challenges they might be facing. So if it's a technological challenge or a productivity challenge, we see if we have solutions for that. Sometimes the teams are also interested in certain technologies that already exist, which they want to use and push further. In that case we help out as well, and sometimes we even work on prototypes together. That's as far as the collaboration between the Technology Group and the teams goes, so we can't really call it »selling«.

> »We can't just add more and more people to a project.«

Making Games But the final decision about which technology a team wants to use rests with the team itself, correct?
Chadi Lebbos That's correct, the final decision is left with the studio. And I think that's what gives Ubisoft the edge. Although there is a set of rules for the productions to follow, you are still free to innovate within those rules, and we have the means to support them in that.

Making Games Speaking of rules, can you explain where the rules stop and autonomy for the teams begins?
Chadi Lebbos It's a very organic transition. We always look for a win-win solution. There is no hard set of rules to use this or that tool. In a

Photos: © Photo Thierry du Bois/Cosmos Image/Ubisoft



Ubisoft puts great focus on procedurally generated game worlds in order to avoid having to add more and more people to ever-increasing projects.

way it's a no-brainer. Why should you develop something that already exists and that has been used by many of our games – that would just be a waste of money. In the end it's based on common sense and also what you want to spend your resources on in your project. Is it on re-developing a technology that already exists or rather on spending my resources on a game feature to improve the game and maybe help to sell more copies of it.

Making Games When you provide a team with a new technology, do you have resources to provide training or to connect them to experts who can help?

Chadi Lebbos The Technology Group consists of developers, but also product specialists, and we also have what we call User Experience (UX) experts. When we create a new technology, we also provide comprehensive documentation, tutorials, and we also have product specialists who go into the teams to teach them. All documentations and tutorials can be accessed at any time through an internal web platform. The UX experts sit down with the users to see how the tools are being used and what may be needed to be improved. We may also adjust the layout of buttons etc. in order to make the tools more efficient to use.

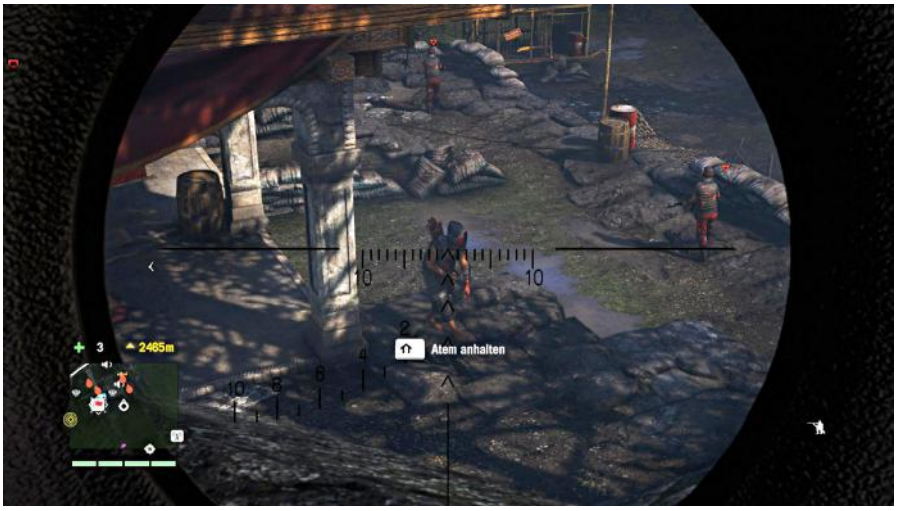
Making Games Speaking of web platforms: Is there a database for the teams to look up which tools and programs you can provide, or do the teams have to contact you and talk it over with you?

Chadi Lebbos We do have a database for our tools, and we're currently working on a huge database with all the tools and scripts that are available in the Ubisoft group. Of course that's a bit of a challenge because technology moves so rapidly, so we need to be able to maintain that over a long period of time.

Making Games Talking about new technologies, procedural generation is becoming more and more popular in game development. Can you talk about how Ubisoft is approaching this technology?

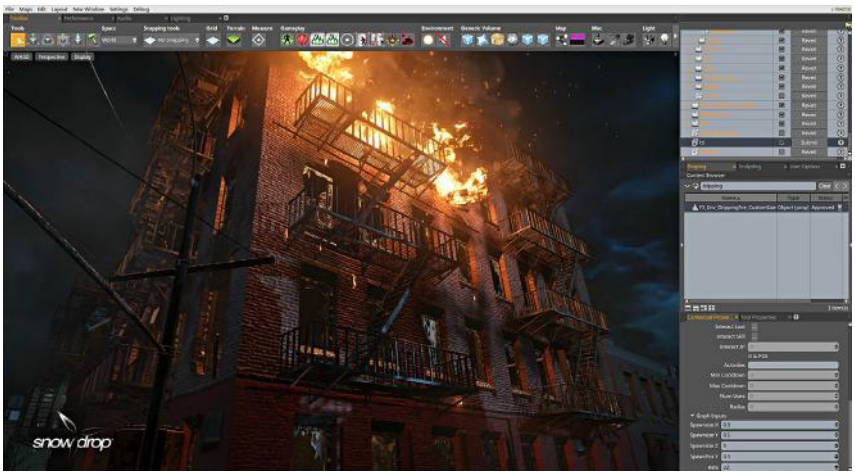
Chadi Lebbos Right now we're experiencing a change within the gaming industry, and not just here at Ubisoft because the games are becoming bigger and bigger. Just to give you an example: When we started doing open-world games, the game worlds had a size of maybe five square kilometers, while today it's more like 25. To create five square kilometers of a virtual world, you need about ten level artists. So it's obvious that we can't just add more and more people to a project as our games are becoming bigger and bigger as quality and depth of gameplay need to be maintained. And this is where procedural tools come in and help us generate content by itself, like the layouts of the game, the trees, the buildings, etc., all of which are actually based on real laws of nature. If there is a river, the system will place brushes there; you won't have vegetation that doesn't fit. If the level designer wants to move the river, back in the days he had to erase everything and re-do it. Today he just moves the river by point and

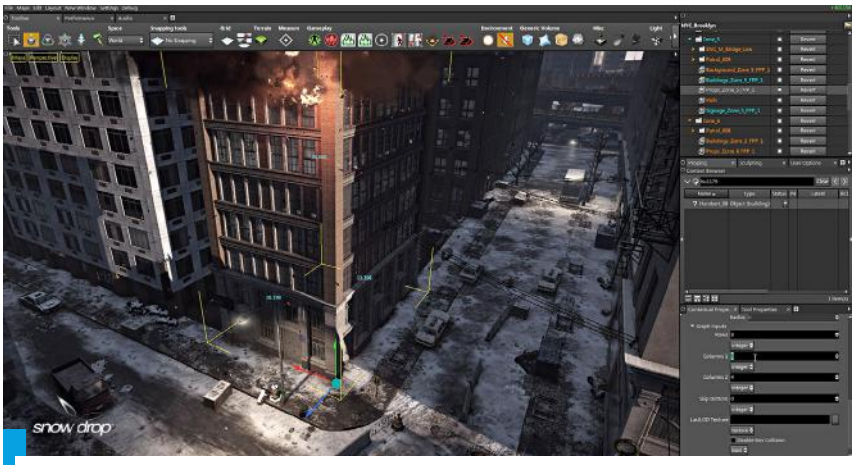
> »Having people in your company who are building an engine gives us expertise that we can build on.«



Just like the game worlds, the nav meshes the characters move on are generated by the game itself.







For The Division, Ubisoft Massive developed its own engine called Snowdrop. The engine is maintained directly by the team that also supervises any changes other Ubisoft studios might want to make to the engine.

click, and everything follows automatically, so there is a lot less manual work to do.

It's a similar thing with what we call »nav meshes«. You can imagine them as the veins and roads within the game on which the AI walks. Years ago, a level designer had to place those paths around each object in the game world. Today there are systems which use a lot of maths, physics, etc. to spawn those nav meshes procedurally across the world. If there is an explosion within the game and the objects move, the nav meshes will automatically adjust their place. This gives us more space in general to focus on innovation than on dull manual adjustments.

Making Games Do you also consider third-party technologies?
Chadi Lebbos Absolutely. If there is a technology that already exists and already does what we need, we won't invest money to develop it ourselves. The Technology Group is always analyzing what is being done externally in order to leverage it or push it forward.

Making Games Would your ultimate goal be to be independent of third-party technologies?
Chadi Lebbos No, not really. If there are solutions we think are good, we see no problem in using them. We're always looking to be independent in terms of our engines though because that way we don't have to wait for somebody to implement a certain technology if we need it; we're in control of it ourselves. Besides, having people in your company who are building an engine gives us expertise that we can build on.

Making Games Speaking of engines: Your newest engine is Snowdrop which is used in »The Division«. How exactly do you develop such a technology? And do different studios collaborate on it like they do on the games?
Chadi Lebbos Basically, the teams analyze what already exists in terms of engines and that is, of course, also connected to what type of game you make. A game like The Division had certain requirements concerning the engine, and that was the reason for the team to build Snowdrop. But to be honest, the way Snowdrop was built to be more productive is very much on par with what the new industry standards are, so the other engines automatically look on how to adapt to also focus on that trend. So it's good to have a couple of engines internally to be able to challenge each other.

Making Games How is an engine like Snowdrop distributed among the teams? Is it something the Technology Group manages?
Chadi Lebbos No, every team that develops an engine is responsible for maintaining it. In case of Snowdrop we're actually setting up a team at Massive that will be monitoring all changes that are being made to the engine and then analyze how and if they can bring it back to the – what we call – main branch of the engine in order to make sure that there aren't too many branches in the end.

»If there is a technology that already exists and already does what we need, we won't invest money to develop it ourselves.«

Making Games How do you determine what technologies the Technology Group works on? Or do you get requests by

the teams if they need a certain technology developed?

Chadi Lebbos This depends on two aspects: On the one hand, we look at the production needs, and on the other, we look at market trends. For example, as I said earlier, procedural generation is something very big at the moment, but that doesn't just concern building the game world, but also things like object modelling, animations, etc. So there are a lot of ideas that we need to spend more time and R&D on to make sure we're moving in the right direction.

Making Games While your group is working on something, do you also contact experts in the studios to ask for their help or feedback?

Chadi Lebbos Of course, this is part of the development process. Every three months we do a presentation update about what we work on, and then we get feedback on it. So if we're developing something in 3D, we're contacting people from the 3D community in all productions in order to get their feedback.

Making Games How do you find the balance between working on required technologies for current projects and R&D projects that may be needed in a few years' time?

Chadi Lebbos To begin with, we make sure that we have people to work on R&D projects, so we actually put people aside in order to be able to do that instead of being fully involved in the production. In terms of VR, for example, we're currently trying to get an idea of what it actually means to be »VR-ready«. When there is a new generation of consoles, there are also challenges, but they're comparatively simple. VR, however, is a completely different beast because we can't just port a game. It's exciting since VR is bringing a new way for us to address entertainment.

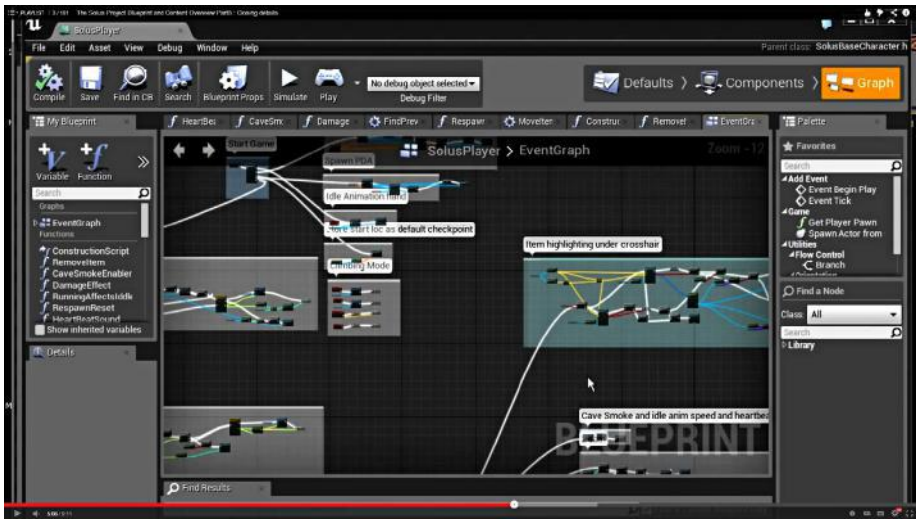
In the »stage gate« process I was talking about earlier we also have a period where we invest in technology in order to see how it can push creativity. Based on that we often find new ideas on how to push online functions, VR, procedural generation of content, etc. even further. And that's what we love about it since there is always something that somebody has a new idea about. A lot of colleagues read tech papers, and we collaborate with universities. We have master students and PhD students, all of whom provide new input. Some ideas may seem like sci-fi, but who knows – maybe one day we'll get there.

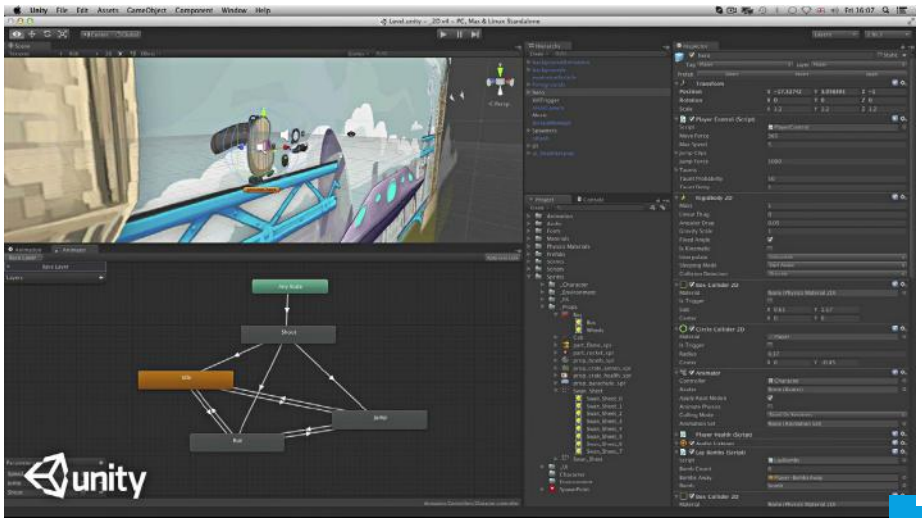
Interview: Sebastian Weber



VR is currently a very important topic for Ubisoft, which is why the Technology Group is working with the development teams in order to get an idea of what it actually means to be »VR ready«.

»VR is a challenge for us because we can't just port a game.«





Although the Technology Group is keeping an eye on external programs and engines like the Unreal Engine or Unity in order to monitor trends, the goal for Ubisoft is to be independent of external engines.



»MY JOB IS STILL A NICHE TOPIC«

As User Experience Director, David Lightbown's job is not to make sure that players get the most accessible products possible. Instead he meticulously analyzes the tools used by the developers in order to make them more effective.



David Lightbown
User Experience Director
at Ubisoft Montreal

David has been working professionally in computer graphics for over 20 years. The majority of his career has been focused on helping game developers find ways to be more productive. David is an Autodesk Master Award recipient, participates in mentoring programs for game developers, and recently published a book: »Designing the User Experience of Game Development Tools«. David currently holds the position of User Experience Director for the Technology Group at Ubisoft Montreal.

»Fast iteration times and efficient tools and pipelines are key to delivering top-quality games.«

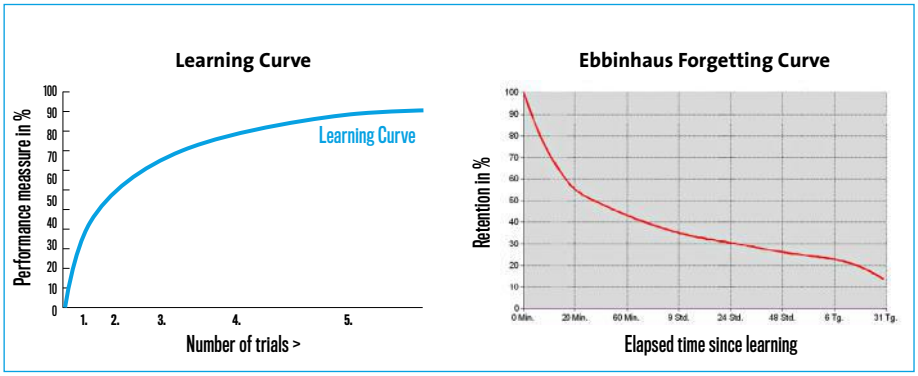
Making Games When we think of »user experience« we usually think of adjustments for the player. But your role is a much more internal one. Can you describe what exactly your job is about?

David Lightbown People usually don't hear a lot about my role as User Experience Director; the job is still quite new, and my position probably only works in a company like Ubisoft that doesn't hesitate to break new grounds when it comes to making game productions more productive. Previously, I worked as a Technical Director for 12 years, and I've been a part of the games industry for about 15 years, and at some point I had this idea to apply the user experience design which you see at other companies like consumer electronics from Apple in a productivity sense on game development. I started reading all sorts of books about user experience and put together a presentation which eventually led to my job at Ubisoft. I came here, presented those slides and explained how my idea could increase productivity. The whole idea was eventually applied to »Rainbow Six: Siege« for which an internal UX team was built. The game features destructible environment, so the player can never count on that wall next to them still being there five minutes later. That's cool, but also complicated from a game development point of view. So we had to make sure to use tools that everybody on the team knew how to use, not just super specialized people. So the UX team took the ideas and techniques that I had put into place and built the required tools for a breakable environment. At the end of the project, there were a lot of team members who thought that due to the complexity of the environment it would have been hard to make the game without these tools.

Making Games Can you tell us a specific example of the effects your work and your approach had?

David Lightbown There are different ways to measure improvement. On the one hand,





The concept of the learning curve (left) is widely known and shows that when learning something new it's especially hard at the beginning to memorize it. But there is also such a thing as a forgetting curve (right) which shows how you forget things over time when you don't use them.

we get lots of feedback simply by talking to people about how they think our UX work helps them to be more productive. In order to optimize the tools, you must understand people's cognitive and visible limitations, and then we can see that they can feel the difference it makes. Part of what I do in my workshop with teams and studios is to get them to understand in a very short amount of time what their cognitive limitations are. On the other hand, some of our tools can measure how much time people spend with things, so we can in fact measure the difference User Experience makes.

Making Games How do the teams react when you join them to improve their productivity? Are they sometimes afraid you might destroy their project?

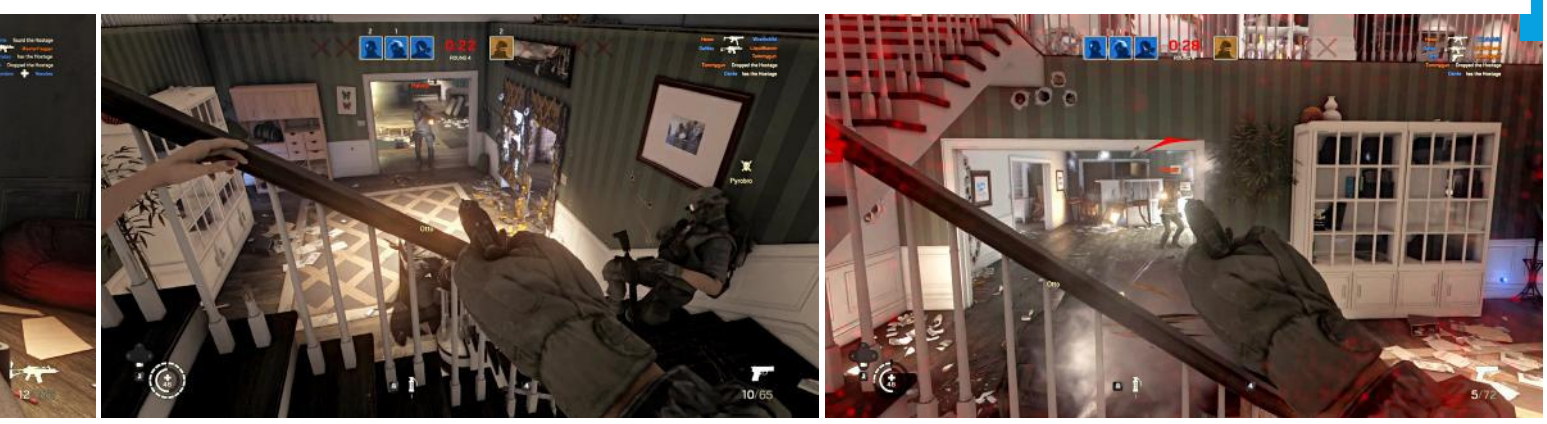
David Lightbown I'm fortunate to have worked in the industry for 15 years. So when I sit down with an artist and he tells me he has to go into Perforce to get a Zbrush file and things like that, I know exactly what he's talking about. So I've been around long enough to understand what they are saying, which is useful and something the teams appreciate. But as a consultant, which is my main role, I'm not tied to one team necessarily; unlike the Technology Group which services all teams worldwide. So I have an external view on people and productions, and I keep telling everyone in our UX department how important it is to go sit with people and watch them work because people do things without even realizing it, and you can find lots of ways to improve their workflow they may not have thought of. Once I'm finished, I move on to the next team and do the same thing all over again.

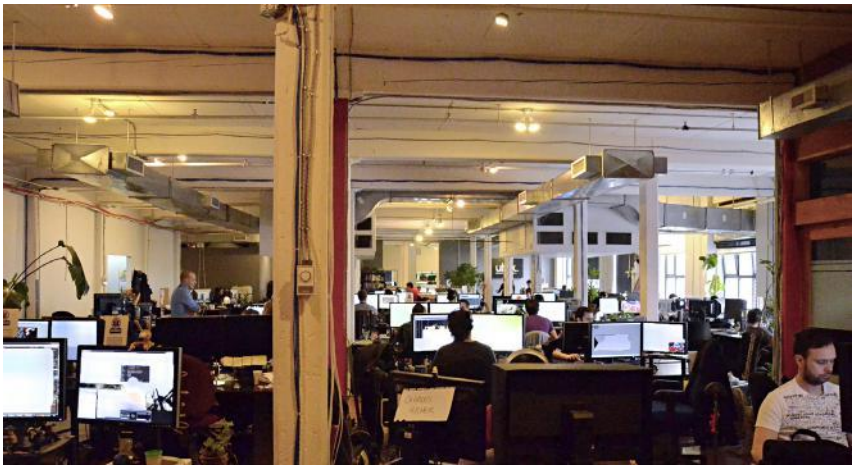
»The human brain works like a computer.«

Making Games How does your work look like when you sit down with an artist? What kind of advice do you give them?

David Lightbown As I mentioned before I think the most important thing really is to sit down with them and to watch closely how they use their tools. There is a story about a Toyota executive who went to the US and spent a month driving around in one of their caravans to really get a first-person sense of what driving there feels like. And I think the CEO of Pepsi once said that he went into a grocery store to see how their products were presented on the shelves because that's how the customers interact with the product. Very few people who create the products really actually see how people use the products, but that is key for two different reasons: For one, it gives you a sense of what needs to be

The multiplayer shooter Rainbow Six: Siege was Ubisoft's first project where an internal UX team was used to optimize the required tools.





When the games industry was still young, there would usually only be one person doing a number of tasks. Today, as the teams are growing, people specialize in certain areas, and consequently tools become a lot more complex to use.

improved, and in our case how we can make a tool better for the specific goal those people are trying to achieve, so that they can think more about their goals and not so much about how to get there. Our task as the UX team is to understand what people are trying to deliver and then to build a tool that helps them reach that goal. The other reason is that there are also certain functions that we spend a lot of time developing because they seem important to us. But when we then watch people how they use a tool, we sometimes realize that we should have shifted our energy somewhere else instead.

Making Games How long do you typically stay with a team?
David Lightbown That varies a lot. It could be anywhere from a few weeks to a few months.

Making Games Can you give an example of how a tool for Rainbow Six: Siege looked like and in what way it improved?
David Lightbown My job for this project was mainly to do a workshop on User Experience with the team and to help them build an internal UX team. I wasn't involved in the actual modification of the tools, it was more the techniques that I put into place. But as I mentioned the most important modification was definitely to make the tools accessible for everybody, so that the destructibility of the levels could be implemented as it now happens in the game. On top, I think it also helped that the team was able to quickly test content and then improve or remove it, simply because it's a lot easier if the tools are easy to learn. Fast iteration times and efficient tools and pipelines are key to delivering top-quality games.

Making Games When you say that the tools should be accessible to basically everybody, you mean that people should be able to use them even without any programming skills or a similar background?
David Lightbown Well, games are made with technology, so of course technology has to play a role in there. However, technology has limitations and as programmers and developers we understand how a computer works, but we also have to understand how the brain works since the brain works like a computer. In the same way that we understand how a program for PlayStation or Xbox works, we should also understand how a program for the brain works. So understanding our physical and cognitive limitations helps us make better tools.

»Intuitiveness doesn't really exist. What people really mean is that it feels familiar.«

Making Games Do you also look at external tools like Unity or Unreal Engine to see how they simplify tools?
David Lightbown There is a concept that some of you might probably have heard of which is called the learning curve, which says that you learn something, understand it better and better and eventually become an expert. But there is also such a thing as a forgetting curve: over time you forget things. So the more obscure the user interface of your tool is, the easier it is to forget how certain things work, especially if they are very specific for that tool. By observing other tools



The fun of the Mario games – in this example »Super Mario 3D World« – is to find the way to the flag at the end of the level. If the game was created effectively, Mario would be placed directly next to the flag.

in the industry and by seeing that consistent patterns are used across these tools, of course, they're a good base for us to build on. When people then open the tool, it feels intuitive for them – although »intuitiveness« actually doesn't really exist. What people really mean is that it feels familiar and that they have seen it before. So yes, we definitely look at other tools and programs for inspiration, and anybody who works in user experience design for tools is inspired by these patterns and guidelines. We just try to take them further and to make them even more efficient.

Making Games How common is this kind of approach in the games industry?
David Lightbown This way to consider user experience during production is becoming more and more common. When I gave my first presentations about this topic in 2012/2013, there weren't a lot of people dealing with the subject. Now you see more and more presentations, e.g., at GDC, and also more people across the industry doing it, even though it's still a niche topic.

Making Games Do you think the topic becomes more important the bigger a company is?
David Lightbown I think it's definitely something that helps when you have reached a certain size. To make our tools and programs more accessible supports cross-collaborations between the teams and studios and generally helps to start projects much faster.

Making Games Why do you think your work is necessary at all? Why don't developers see themselves that a tool isn't efficient?
David Lightbown I think the reason for this is that all the technology, consoles and all the things we use to make games are just too complex, and it's hard to make it simple. We come from an industry where there used to be only one person doing the programming, graphics, sound design, etc. As game development got more complex and the teams were growing, it was a natural evolution that people started to specialize in certain areas. Where you had a generic programmer a few years ago, you now have someone specializing in physics, rendering, etc. So it's only logical that we also have people who specialize in productivity for tools.

Making Games Do you think there will be a point in the future where you can say your job is done?
David Lightbown No, my job will never be done. In fact, that's a good point because with every game and with every generation of consoles the technology becomes more and more complex; the data become bigger, the gaming experiences become richer, and we can't use the same tools every time to be able to deliver these things. We have to be constantly improving our tools and optimizing our pipelines, not only for one game, but for an entire franchise. So while a team is still working on a game of one series, we're constantly working on the tools in the background to make sure that they're already more optimized for the sequel.

> »Once I'm finished with my job, I move on to the next team and do the same thing all over again.«

Making Games Certainly every team has UX designers who are focusing on the players. Are there any overlaps between their work and yours?
David Lightbown That's a good question. I'd say the techniques are the same, but the goals are different. When I take a product like my iPhone, one of the goals to a certain degree is just desirability and pleasure using the device. For us that's not our goal when it comes to tools. Our goal is to be efficient and to iterate as many times as possible.

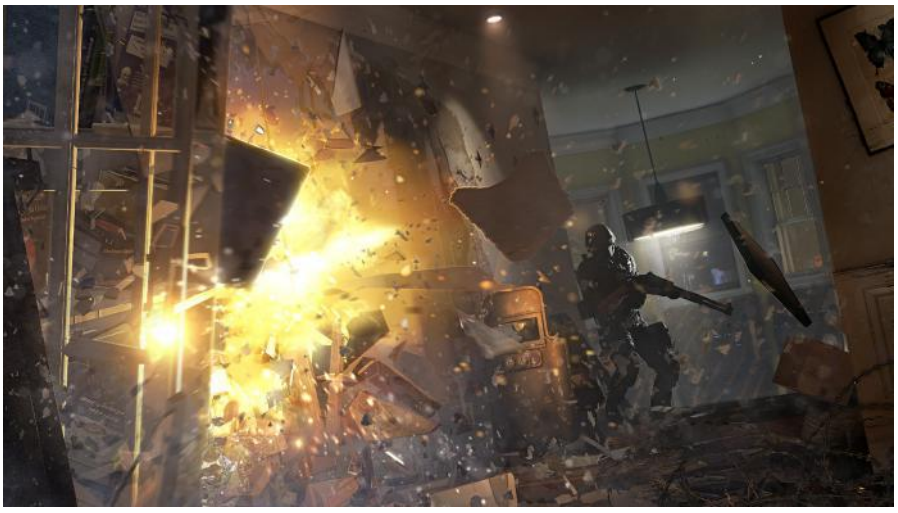
So if I was to make a »Super Mario« game that was efficient, I would place the flag that is to be reached directly next to Mario for him to pick it up right away. However, we have blocks and things that we can jump over to get to the flag, and this way to manipulate the game world is what makes it fun. We want our tools to be the equivalent of this effective Mario; we want the mouse to be next to the flag and to be able to pick it up right away. The techniques that are used to add tension for the player, can also be used to reduce those obstacles as well. So there is some cross-correlation between the UX for the player and the things we do.

Interview: Sebastian Weber

Further information

David Lightbown held two presentations about User Experience for Tools, and he has also written a book, which can all be found at the following links:

- Recorded presentation of »The User Experience of Game Development Tools« held at GDC Europe 2013 from the GDC Vault (free of charge): **www.makinggames.biz/UXGDCEU13**
- Book »Designing the User Experience of Game Development Tools«: **www.makinggames.biz/DavidLightbownBook**
- Presentation slides to the talk »Tools Development at Ubisoft: Building a Shared Vision to Improve the User Experience« held at GDC 2015 from the GDC Vault (a video recording of the same title is available from the GDC Vault at a charge): **www.makinggames.biz/LightbownToolsDevelopment**



When making the destructible environment for Rainbow Six: Siege it was necessary to provide tools that everybody on the team was able to use in order to allow as many iterations as possible.



»PLAYERS WILL BE LIVING IN HUGE VIRTUAL WORLDS, ALMOST LIKE IN THE MATRIX«

We talked to AI developers Julien Varnier and Gabriel Robert who gave insight into Ubisoft's discussion culture and shared their opinion on the presence and future of Artificial Intelligence.



Julien Varnier
Game AI Technical Lead
at Ubisoft Montreal

Julien has been an AI Engineer in the video game industry for 18 years. First working at Quantic Dream in Paris, France, on Omikron: The Nomad Soul, he then developed game-related AI technologies at MASA Group before relocating to Perception Studio in Sydney, Australia. He started working for Ubisoft in 2006 on Tom Clancy's: EndWar at their Shanghai Studio in China before moving to Montreal in 2008 where he has been working on Avatar, Far Cry 3 and Far Cry 4.



Gabriel Robert
Senior AI Programmer
at Ubisoft Montreal

Gabriel is a Senior AI Programmer who has been working in the video game industry for more than 15 years and at Ubisoft since 2008 (both in Montreal and Paris). He is specialized in Artificial Intelligence with a PhD thesis from LIP6 (Laboratoire d'Informatique de Paris 6). He has worked on the AI for games such as Avatar Wii/PSP, Tom Clancy's Ghost Recon Future Soldier and Assassin's Creed 4 Black Flag.

Making Games Let's first talk about UDC where you did a presentation. What do you think is the benefit of having an event like this?

Julien Varnier UDC is now a really big conference within Ubisoft and has been running for seven years now. So the people who organize it have a lot of experience, which makes it easy for us to submit and prepare presentations. The whole conference is very professional and since it's still an internal conference, we're free to talk openly about things. This year, for example, we didn't talk about our current project because we wanted to have a more general talk about production and technical processes. You can't have that kind of open approach if you talk at a public conference because you always have to consider what you should best talk about, what you can't talk about yet, etc.

> »When it comes to AI, we are only just scratching the surface.«

Making Games If you do a more general presentation to inspire people, is there a way for them to contact you after the talk?

Julien Varnier There is no official channel as we are all in the same company and everybody has got everybody's email. Even in Montreal where we are many people, still a lot of people know each other, so it's very easy to continue the discussion after a talk. It's a good question in the sense that also what we try to do in this kind of presentation is to open up a conversation on various topics and to come up with ideas, not to solve them necessarily. Today we talked about situated AI and provided ideas of how to build a production pipeline. Those are things that everybody is aware of, but we try to formulize it a little bit, and the idea is to make people think about it, share their thoughts and to open a discussion on that.

Gabriel Robert The idea of this talk was also to get feedback and maybe that somebody from a team who attended the presentation would explain it to others and talk to them about it. Of course we follow up on this and people can always contact us, too.

Julien Varnier Especially today with a lot of productions being open world these days, the topic AI is even more relevant. Often issues have been boiling for a while, without anybody really understanding what the problem is. That's why the first step to finding a solution is to ask the right questions, and when we prepare a talk like this, we first think about the questions that we're asking ourselves. And that's the ideal moment to start a discussion.

Gabriel Robert What's interesting is that at UDC you have conversations with different people from different projects and studios on the same topic, and you can really see the synergy on projects. We may not have all the answers, but we have the feeling that we move in a common direction.

Photos: © Photo Thierry du Bois/Cosmos Image/Ubisoft



The Artificial Intelligence in today's open-world games like Far Cry 4 (left) and Far Cry 3 (right) isn't really a lot smarter than a few years ago, but it can deal with much more constraints.

Making Games Is there a specific way to keep everybody in the company in the loop in regards to these discussions and outcomes?
Julien Varnier All presentations are recorded and streamed internally.
Gabriel Robert And of course you can have a one-to-one conversation with the presenter or we exchange emails. But what I actually try to do when I receive a question that I think can benefit the community is to post it in our internal social network. Then I send an email to people in my network, hinting to that question, and maybe we can iterate on it. All this happens publicly within the company and not just in a hidden thread somewhere.
Julien Varnier A good thing at Ubisoft and in Montreal in particular is that there is a lot of communication between projects and teams. With presentations and such we basically plant seeds, knowing that afterwards some communication will happen and people will start talking about it.

Making Games Change of topic since you're both AI programmers. How do you think AI systems have changed in the last five to ten years?
Julien Varnier First of all, AI is a super exciting part of video games because it's probably one of the areas where we are actually only just scratching the surface. I have no idea how far we can go regarding AI, but I'm pretty sure when we look back in 10 or 15 years it's going to be a whole different world. It's not just a question of whether we can make a better AI and smarter characters. We have already made huge progress at having an AI that can work properly in a complex virtual world, something that took a lot of effort in the past. Basically, the AI we have today is not necessarily making more complex decisions than in a corridor shooter, but it can deal with much more constraints. Building a character in Far Cry today who is able to engage you in the open, to shoot, avoid fire and so on is a lot more complex than doing the same type of soldier several years ago. Today we expect characters in a game to follow their own agenda and therefore to do something particular or not, or maybe they can even remember what the player was doing and act accordingly. Here we are already touching the limit of what our AI can do, even though we are still making big progress. So from that perspective it's very exciting to work in this area.

Today there is also something we call »AI Director« where the AI tries to balance the game in real-time, like Valve used in »Left 4 Dead«. Maybe at some point we can start to build games where the AI is kind of like the game master in a tabletop RPG. This certainly won't happen within the next five or ten years, but it's a direction we're heading for. Last but not least, another exciting thing about AI is that it's in the middle between pure technology and design. It's not about creating the best or perfect AI; for me AI has a gameplay purpose, for fun, and is directly impacting the game experience.

> »For us, AI has a gameplay purpose, for fun, and is directly impacting the game experience.«

Gabriel Robert As Julien said we really aim to have virtual characters that can understand a situation, behave accordingly and remember different things. If we look at what we have and where we want to go with these AI characters, it's still a long road ahead. The effort it takes though depends obviously also on the scope of our productions. 10 to 15 years ago the games were simpler and it wasn't a massive amount of work to create them, so from one production to another some technologies were developed again from scratch. You simply couldn't do that today. And that's where conferences like UDC or our general approach to share knowledge between the studios and teams prove to be helpful. This is the only way we can make sure that we don't re-invent the wheel every time.

Making Games When you say you basically want an autonomous AI, wouldn't it be a big disadvantage compared to an AI that acts exactly as the designer intended?



Not all presentations at Ubisoft's internal developer conference are intended to find immediate solutions, but rather to open up discussions about current questions.

Gabriel Robert When people think of AI, most of the time it's about having a smart AI that works perfectly and always finds the right solution for a problem. But that's not our approach. In our opinion, AI really should be about supporting the gameplay experience. AI isn't meant to find solutions, but to entertain the player and to interact with him without looking stupid – unless you want it to look stupid. But we don't want it to be perfect because that would feel really odd to the player.

Making Games In your opinion, what is currently restricting the development of AI systems?

Julien Varnier There are some limitations, but it's usually a matter of scope and how we can actually express what we can do – it basically comes down to animation and audio feedback. It wouldn't be that complicated for us to create a lot of different kinds of behavior, but you need the animation to do that, and for this animation you may want some variety; in addition you might have different versions for different kinds of NPCs with different weapons, etc. So sometimes we just don't do an AI behavior because producing all the assets for it could be very expensive. We're definitely making big progress on the animation side, also in terms of procedural generation, and obviously it's a lot less expensive to just read data instead of generating animations, but it will still take some time for us to do it the way we envision it, which will eventually make life a lot easier for us AI programmers.

It's the same with the audio feedback. It's quite complex to program an AI that understands exactly what's happening in the game and replies accordingly in conversation. But even if we are making progress here, we can't use something like voice generation, although these systems are slowly getting better and better. Instead, every time a guy says something smart or something about what's going on, we need to record their voice, edit it, and then it needs to be localized into different languages, so there are very high upfront costs.

But I'm confident that in two, three or five years these limitations will disappear, which will also benefit the game design. At the end of the day, at Ubisoft we always think from the player's perspective, and that's why it's always about enhancing the gaming experience.

A good example where we actually learned that kind of humility was Far Cry 3. I think the AI was working really well in the game, but when we looked at the reviews, nobody really talked about the AI. At the same time people were saying the combat was well balanced and stealth was working well. So when I read that, I know that AI is part of this, even if they don't say anything about the AI. But if your AI and its behavior isn't well balanced, combat and stealth won't work properly either. So of course we want to do the best AI possible, but the real goal is that it fits the gameplay.

> »Open World is the ultimate gaming experience where players can romp around.«

Making Games So you would like people to take more notice of the AI as part of the game?

Julien Varnier I would have said that a few years ago, but today I don't really care. As long as the experience in the game is getting more interesting with more content and more interactivity, and we can create more dynamic stuff and deeper things like a real dynamic narrative, I know that AI will be a big part of this. When people play our products and maybe something cool happens and an NPC in a later mission remembers it, and when the people then share this on YouTube or Twitch, it's a win for me.

Gabriel Robert That's exactly the kind of things we want: We want the players to experience something and basically create their own story. Especially if something unexpected happens and the player still gets a reaction, this is thanks to our AI, so we are really happy to get this kind of feedback Julien was talking about.



Artificial Intelligence could ultimately have an »AI Director« balancing the game depending on the player behavior - a system Valve already introduced in »Left 4 Dead«.



Since Ubisoft puts a long-term focus on open-world games, they want characters to act more and more independently and authentically, to remember things, etc. The characters in Far Cry 3 (the top two) and Far Cry 4 were still acting strictly according to what the designers wanted them to do.

Making Games Everybody in the industry is currently talking about open game worlds and systemic game design – but where do you think the industry and Ubisoft will be headed to in the next few years, and what could this mean for Artificial Intelligence?

Julien Varnier One thing that's here to stay for sure are open-world games. I'm pretty sure about that not only as a developer, but especially as a player. I'm not saying every game should be an open-world game now, and we can see that the industry has diversified a lot, right now with big games like this, but also mobile games or indie games, so there are games for everyone. But I think open world is the ultimate gaming experience where the players can romp around.

»We don't want the AI to be perfect because that would feel really odd to the player.«

Of course, a lot of the things we're doing today are still subject to change, and we will improve or remove one thing or the other. I'm pretty sure that players dream about »living« in a huge virtual world in 20 to 30 years, almost like in »The Matrix«. A game world which can be everything; it can be fantasy, it can be realistic – I think that's not a short-term vision.

The things we do in open world, even though they're very limited right now, are basically just small steps in this direction. The basic content, an open space with smart interactions in it, characters, animals and so on will be the same, just with deeper stories and no scripted content. Again, as we said, we're just scratching the surface of what's possible. Most of the open-world games right now still have a very linear campaign despite all that free space, that's why everybody is trying to go into a more systemic direction and create an entertaining ecosystem.

Gabriel Robert We can't look too far ahead into the future since we don't know how fast AI will progress, e.g. if we look at other industries like robotics. But when we look at the progress of smartphones and how you can talk to those devices, I think we will have a very different relationship to AI in just a few years. Coming back to videogames, today the characters have very little autonomy and mainly act based on a script created by a designer. But I think in ten years those characters will have their own history, their own memory and will be able to memorize things and recognize characters. The challenge then will be to find new ways to tell a story with those characters who live in the virtual world and have their own story, but at the same time are supposed to be there to entertain you.

Interview: Sebastian Weber



»I DON'T CHECK EMAILS ON MY SMARTPHONE«

Intrigued to work on Assassin's Creed, Sadia Rahman Zain started working at Ubisoft Singapore. Working on global teams poses a lot of challenges, but also creates opportunities.



Sadia Rahman Zain
Senior Gameplay Programmer at Ubisoft Singapore

A Gameplay/AI Programmer for more than nine years, Sadia started her career as a Software Engineer at EA UK where she worked on the Harry Potter and Hasbro Family games. She then joined a startup called Supermassive Games where she worked on several PS Move titles and Until Dawn. In 2012, Sadia moved across the seas to join Ubisoft Singapore as an AI Programmer on Assassin's Creed 4 Black Flag and Assassin's Creed Rogue (you can blame her for the notoriously difficult legendary ships). She is currently lead GPP on an unannounced project.

Making Games What's it like for you being at UDC, and what do you expect to gain from this?
Sadia Rahman Zain It's a really cool experience for me because it's a good time to actually meet other people who are doing surprisingly similar things like you. Often I realize then that they're working on something that we're also looking into, and so you start to talk, find out what their problems were or what they did to try and solve it. It's also a great opportunity to see where the games market is going in general and what kind of developments are currently being pushed. I also listened to a few talks that were very relevant to my daily work and that I want to follow up on with my team at home. A good example was the talk about an animation technology that is currently being worked on. As a gameplay and AI programmer we're always limited to what the animation can do. The presentation showed that characters can even be realistically animated in a dynamic environment, which opens a lot of doors for us in terms of AI.

> »I like the idea of players creating their own stories instead of the game being fully scripted.«

Making Games Can you give a specific example of how the insight you gained at UDC might help you in your daily work?
Sadia Rahman Zain Some of the previous games I worked on had an open game world, and the systemic AI talk I attended this morning was specifically about that: how gameplay and AI programmers have to keep pushing the message that we have to let go of control. Since open-world games have only become more and more







popular maybe in the past five years, the paradigm is slowly shifting because in the past we used to script everything. But that doesn't work for that specific type of games, and it was kind of good to see that it's not only my team, but also other teams having to deal with that problem. It was also interesting to see how they solved it, and I think we can take something back from that and apply it to future projects.

Making Games So do you think systemic AI is the future?
Sadia Rahman Zain I actually really like systemic gameplay and the idea of players creating their own stories instead of the game being fully scripted. At the same time, there are some games I like that work differently. So I wouldn't say that one is better than the other; they're just different types of games. What I find more interesting to work on is systemic AI and such things because it's more challenging for me.

Making Games The studio in Singapore was involved in the development of the sea battles in Assassin's Creed 3. What was your personal experience about working on such a big project? Did you have the feeling to be working on an important part of the game or more like a small cog in the big wheel?
Sadia Rahman Zain That's right, in Singapore we were working on the naval section of Assassin's Creed 3 and everything around that, and it was the first time for us to be a part of one of these big AAA projects, and with the scale of an Assassin's Creed to boot. Other studios, including Montreal, worked on missions on shore and other parts of the game.

We were given the mandate to work on this part of the game and to be creative and go crazy, and in the end the naval experience was so awesome that it was used to announce the title at E3. It was obviously nothing that could stand on its own, and we had to find a way to integrate it with the content of other studios. But especially in the following »Assassin's Creed« titles like Black Flag, which had sea battles as well, we were able to focus more on how to merge the different game experiences more smoothly.





For Assassin's Creed 3, the studio worked on sea battles for the first time. This part of the game worked out so well that it was extended in the sequel Black Flag (see picture), for which a pirate setting was chosen.

»It can be very scary for a small studio to take on a AAA project.«

Making Games Can you describe how a production of that scope, which is spread across the globe, is organized?
Sadia Rahman Zain For Black Flag, for example, I think there are nine or ten studios across the world, working 24 hours on the game. Montreal and Singapore have a time difference of exactly 12 hours, which means somebody is always working on the game. Of course, working with multiple studios has its challenges; communication is definitely one of them, but

Ubisoft's Singapore studio was opened in 2008 and is regarded as one of the best IT employers in the region. They now have over 300 employees.







at the same time the benefit we get from it is really worth it, having different cultures and different experiences. Also, this kind of collaboration allows smaller studios to specialize on certain aspects on the one hand and to be part of a AAA production on the other hand. It can be very scary for a small studio to take on a AAA project, but this way the team is slowly eased into it instead of being left on their own.

Making Games Isn't it extremely stressful to work in a 24-hour environment, always feeling like having to check emails, etc.? How do you cope with that?

Sadia Rahman Zain It's quite simple: I don't check emails on my smartphone. I specifically don't do that because I think it would be hard to switch off. If you read that someone in a different studio has a problem with something and you know the answer to it, you want to help out. But I think it's not too much of a problem as long as you're self-disciplined. You have to take your time to rest, otherwise you will just be wasted the next day. Generally I wouldn't say though that it's necessarily more stressful as if just one studio was working on a game every day. Just think something goes wrong and you detect a massive bug in a game. In that case I just send an email to, let's say, somebody in Montreal, saying we really have no clue how to fix this, and the next day I come to work there is a response telling you how to do it or the problem has already been solved. Seeing the changes to the game that sometimes happen overnight or detecting new content has some effect of motivation and inspiration.

»Of course, working with multiple studios has its challenges.«

We had such a case with Black Flag once. One day, we came to work and saw that there were dolphins swimming next to our ship in Black Flag, which some colleague from another studio had added to the game. We liked them so much that we brainstormed about how we could use them as a gameplay element, and eventually we used these dolphins to lead the player through the world in certain situations. Things like that are clearly the positive effects, but of course, like I said, there are negative ones as well.



One day, the team in Singapore detected dolphins swimming next to the ships in Black Flag, which a colleague from a different studio had implemented into the game. So the team quickly decided to use them to lead the way for the player.



Before Sadia joined Ubisoft she was part of the development team of the PlayStation 4 horror game Until Dawn.

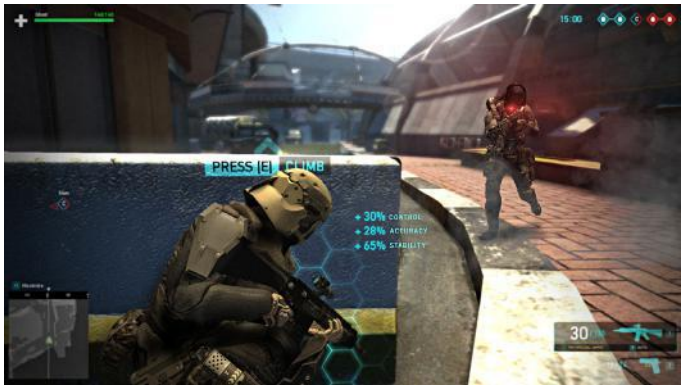
Making Games Are you saying that although so many studios work on a game in various locations, there is still a sense of community?

Sadia Rahman Zain Yes, definitely. We do studio visits, especially the lead programmers and other key people travel a lot, but even people in junior positions travel around the studios. And we always have conference calls, video chats and, of course, we stay in touch through email. At the Singapore studio we have a kind of policy that if you get an email from someone from another studio, you should answer it within 24 hours to keep a constant communication flow going. At the end of the day, the fact that we are in different studios doesn't matter: if the co-worker sitting next to you is bad at communicating, being in the same office doesn't help.

Making Games You've been working at Ubisoft for quite a while now, and you were at other development studios before that. If you look back, do you have any general advice, like the ultimate game dev wisdom?

Sadia Rahman Zain If there is one thing that I would say to any kind of junior gameplay programmer it would be to always build prototypes, even in small pieces. It's always great to have a fantastic design on paper and people who are talking about this and that, but often the best ideas fail when you actually try them. That's the one thing that's definitely key for both designers and programmers: to try something and then – based on what you





Since Assassin's Creed 2, the studio has been involved in the Assassin's series. But the team was also part of Prince of Persia: The Sands of Time (left) and acted as the lead team of Tom Clancy's Ghost Recon Phantoms.

learned from the trial – to iterate, tweak and improve it.

Making Games Can you give us a specific example of this from one of the projects you have worked on?

Sadia Rahman Zain We had an idea for Assassin's Creed 3, which was dismissed because everybody thought it would be too hard to do. When we then worked on Black Flag, this idea came up again, and everybody loved it. Basically, it was about having so-called legendary ships in the game world, for example four ships in four corners of the world, which would be tough to defeat. When I discussed it with my game designer, it was first about how the challenge should look like at all. We gave the rough estimate that only about five percent of the players should be able to defeat them. When we then designed the legendary ships, we designed one to be »twin ships« – two ships. My designer built prototypes, also experimented on paper with regard to how these two ships should interact, but the idea kept being pushed back because it was thought to be too hard. I just didn't want to believe it; I thought it was a brilliant idea to have two ships launching coordinated attacks against the player, creating epic cinematic moments. So I basically told my designer to leave it with me and let me play around with it.

When we work on the AI for something, we only take small steps. So the first thing I did was treat each ship and the player's ship as just points, with no directions and only a position in the world. When you start with such small steps, there isn't much room for complications and mistakes. An AI should always be kept as simple as possible because the more complex it becomes, the more likely it is that something will go wrong. Besides, the simplest AI is often the best AI, and people think it's a lot more complex simply because it doesn't fail. That's definitely the key.

In my case I had these three ships which I treated as points, and I got the ships to treat their enemy as a point, to keep circling them, keep going, one way down each side, broadside, turn around, broadside, etc. And that alone created this epic moment. When I gave it back to my designer, he was happy that it worked. From this point on we just iterated on that a little bit more, tried this and that, added options of different weapons, etc. So to reach a great AI and epic moments, it's definitely all about starting small and prototyping.

> »At the end of the day, the fact that we are in different studios doesn't matter: if the co-worker sitting next to you is bad at communicating, being in the same office doesn't help.

Making Games Last but not least: What was the reason for you, coming from Europe, to join Ubisoft Singapore?

Sadia Rahman Zain Um, the weather (laughs). I have a cold right now, which isn't great at all. Here in Singapore it's like 35 degrees almost all year round. Joking aside, it was also the opportunity to work on Assassin's Creed. Previously I worked on Until Dawn which was more like a horror game, and in my earlier career I was doing more or less kids games. I wanted to be part of a project that had more of a AAA character and was more challenging for me as an AI programmer; something that had more in-depth AI.

Interview: Sebastian Weber



»FORCING A GAME TO FIT VR DOESN'T WORK«

Eagle Flight, the game Olivier Palmieri is working on, is Ubisoft's first VR game. He gives us insight into the research phase at the beginning of production and the way Ubisoft's creative department, Fun House, works.



Olivier Palmieri
Game Director of Eagle Flight at Ubisoft Montreal

Olivier is Game Director on the upcoming VR game Eagle Flight, developed by a small team in the Fun House department at Ubisoft Montreal. Olivier joined Ubisoft in 1998 and has worked on a number of different AAA projects over the years like Far Cry, Assassin's Creed, Rayman, Tom Clancy's Splinter Cell and Naruto. His expertise in gameplay, game design and level design led him to also take up the role of International Game & Level Design Trainer for the Ubisoft Design Academy, an internal training program that has benefited more than 500 designers to date.

Making Games Eagle Flight started out as an R&D project in the first place. Can you describe how the project was kicked off and whether different studios are involved?

Olivier Palmieri After finishing my work on Far Cry 4 I wanted to familiarize myself with that new thing, VR, just a few days later because I had always dreamed of flying around like a bird. I was able to convince my bosses that we had to analyze various VR devices and everything that was possible with this type of medium from scratch, and so we started a few smaller R&D projects with just a few people on each project – at first we were only three people. The problem is that we can't just take an existing game concept and game universe and adapt it to VR. That's why I wanted to really learn about those issues and familiarize myself with the whole matter to find out how we could make the best out of VR in terms of controls, accessibility, etc. It's this kind of experience that Eagle Flight is based on today.

> »In VR you have to be very careful not to take the camera control away from the player.«

For a few months we were working in our small team only, but over time I was able to get a few people from other teams in Montreal on board, who often joined us on a short-term basis, supported us with a certain task and then switched to a different project again. That's the great advantage of our Fun House department (see page 47). It allowed us to experiment with and develop a lot of things, even if not everything we were testing actually worked out. I was also in contact with other studios, for example in France, or with Red Storm to share our experience and knowledge or even assets and scripts. We're generally used to this kind of approach, and in terms of VR it makes even more sense.

Making Games Does that mean that if a team starts a VR project, they can contact you anytime to ask for advice or get feedback on prototypes?

Olivier Palmieri I know most of the teams that are currently working on VR content,

Photos: © Photo Thierry du Bois/Cosmos image/Ubisoft



In late 2015, Ubisoft announced Eagle Flight as their first VR game for PlayStation VR, which gives players the chance to soar freely through a post-apocalyptic Paris as eagles.

and we're trained to work together as one big group, share our knowledge and keep up a good relationship in general. So I did indeed receive emails from peers asking me for advice.

Here at UDC we also have a VR track and I'm one of the committee members and organizers of this track. For that reason alone I'm in touch with pretty much everyone working on VR.

Making Games Can you give a few examples of things you have experimented with and which maybe didn't work or how you had to adjust them to make them work since VR is still a very new medium?

Olivier Palmieri Of course we tried a lot of things that didn't work. For example, in VR you have to be very careful not to take the camera control away from the player as this can cause nausea for some players. We also experimented a lot with the controls or the frame rate, and little by little we were able to understand what was working and what wasn't.

I love science and every night before I go to sleep I read articles or watch scientific videos. I find it very exciting to take a scientific approach to VR, learn about biology and understand why VR can cause discomfort at all and what we should do or shouldn't do to avoid it. Once we found out about all these things, it was rather easy to try out certain things and come up with solutions.

At the moment, a lot of people create games and then try to adapt them to VR, but that doesn't work. We had the same situation with mobile back in the day when a lot of developers were trying to port existing titles one to one to smartphones and tablet PCs, which just didn't work. These guys had to find out as well what was possible on mobile devices and what they were best suitable for. And now we're doing the same thing with VR because it is a new medium and we can't just port anything. Of course there are ways to port existing games, but you first have to figure out what you can do with a new type of medium and what you shouldn't do.

Making Games I have often read and heard that acceleration and fast movement in VR games can cause discomfort for the player. Do you have an idea why this isn't a problem in your game but in others it might be?

Olivier Palmieri As I mentioned we studied for weeks, if not months, what might cause





Virtual Reality is currently a hot topic for Ubisoft, with all teams that are working on content for the new medium (like Red Storm with »Werewolves Within«, see below) being in close contact.





Palmieri and his team have done a lot of research to find out what causes players to feel uncomfortable while playing. As a result, Eagle Flight causes nearly no motion sickness despite fast movement and changes of direction from a first-person perspective.

discomfort in the player and what we could do to avoid such situations. As you saw while playing, our game Eagle Flight has a lot of motion, sometimes very fast, and it's not the static experience many other projects provide at the moment. But we wanted to tackle this problem and find solutions head on, simply because we firmly believed that these solutions are out there somewhere. Through our research we found a few feasible techniques. We're still in the process of fine-tuning them, but we can safely say that there are things you can do in VR and others that you shouldn't do. And if you study them long enough, you will eventually find solutions for those big issues as well.

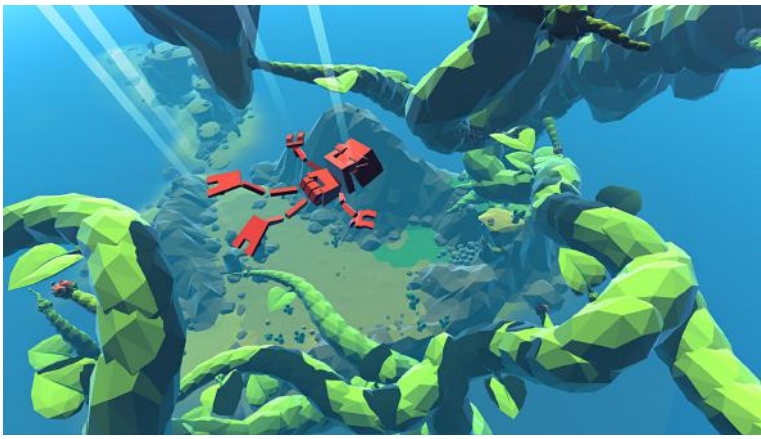
Making Games I assume you had a lot of people playtest Eagle Flight – after all in Montreal alone you have several thousand people who can try out the game. Were you able to find out how long somebody could play a game like Eagle Flight without feeling discomfort?
Olivier Palmieri Basically, Eagle Flight is designed to be playable in short sessions. A multiplayer match, for example, is approximately five minutes of play and then players are taken back to the lobby which is a more static experience. But in fact there is no real time limit from our experience. VR can best be compared to a new set of glasses. At first it feels weird, but over time the body and brain adjust to the new glasses. The same thing happens in VR. If you try the headset on for the first time, it takes a little while for the brain to get used to the new experience, but the more you use VR, the less sensitive you become. Now I can spend hours in VR – which I actually do – without any problems at all.

In the end it depends on the player. If you prefer shorter sessions of maybe 15 minutes and you feel fine, it's no problem to play VR like a mobile game. On the other hand, you can also play VR games for hours. Coming back to the glasses example: For newcomers to VR it's important to create experiences with short sessions of play, and then people will begin themselves to slowly increase their play time.

»Of course there are ways to port existing games.«

Making Games How do you think people will adapt to VR in general?
Olivier Palmieri VR is amazing because it allows you to make so many new experiences,





The Fun House division inside Ubisoft Montreal encourages people to propose new ideas for games. Games like »Grow Home«, »Child of Light« and »Valiant Hearts: The Great War« were created this way.

and this is only the very beginning. It can also be used in all sorts of ways in terms of technology. You could create a virtual museum, send people virtually into space, you could use it in health facilities, etc. Having said that, I don't think VR will replace anything. When the radio was invented, everyone said that people won't be reading books anymore because they had the radio now. Then it was television, and people said that nobody would be listening to the radio anymore and so on. The way I see it is that VR will simply be integrated into everyday life. People go see a movie, watch their favorite show when they get home at night and then maybe also a little VR. I've been working with this new medium for 18 months now and some days I use it for hours. But I always get to a point where I happily take off the headset and go out with my friends or just do something else. So VR for me is simply another – a new – way of experiencing certain things.

»For newcomers to VR it's important to create experiences with short sessions of play.«

Making Games Eagle Flight was originally a project of your Fun House department. Can you explain what Fun House is all about?

Olivier Palmieri That's right, Fun House is a division inside Ubisoft Montreal. The idea is that anybody in the company can propose ideas for a game. You don't have to be a game designer or work in production; if you have a dream, you can propose it to Fun House. There is a process to analyze the idea in terms of whether it can be done at all, how many people would be needed, how long it would take to develop it, etc. Once you have passed this step, there is a great chance for the project to be developed, even if you've started as a »nobody«.

This is how Eagle Flight came about. I proposed the concept, I even proposed a prototype, and Fun House then decided they liked the idea and to go ahead with the project. So I was able to develop my game. Of course there are also a lot of other projects apart from VR, which have been created this way. The basic idea is simply to motivate people to bring in new ideas that may differ from other Ubisoft games. You might say that Fun House is something like an incubator for new ideas where you can try concepts a lot faster or build prototypes to see whether an idea can work or not.

Making Games If anybody can propose an idea, what would it look like if somebody from PR or marketing presented a concept? Would Fun House just take the idea and make the game, or how do they include somebody with no production experience?

Olivier Palmieri Actually, there are currently some people working on Fun House projects with no production experience at all. In this case we just use the experience they have – if you're from PR, we will use you to coordinate the PR for your project. Everyone who puts in ideas will also become part of the team. It's Fun House's job then to determine how many game designers, programmers etc. will be needed to start a project. Since it might be difficult for people with no production experience to propose a concept, there is the so-called Fun Box. It's a questionnaire detailing which information is necessary in order to propose a concept to have it analyzed. [see details about Fun House from page 37]

Interview: Sebastian Weber







»UBISOFT IS STILL QUITE THE FAMILY BUSINESS«

Running Ubisoft's Toronto-based studio, Alex Parizeau is looking after 400 employees. He tells us about future plans for the studio, explains how the worldwide locations work together and based on which criteria the projects are split between the teams.



Alex Parizeau
Managing Director
at Ubisoft Toronto

Alex Parizeau is Managing Director at Ubisoft Toronto, responsible for studio strategy, management and production oversight. A founding team member of the Toronto studio, Alex was previously Senior Producer working on Tom Clancy's Splinter Cell Blacklist, managing the game production team and studio collaboration for the project with Ubisoft Montreal and Ubisoft Shanghai. A member of Ubisoft since 2005, Alex previously worked as Producer on several critically-acclaimed titles including Rainbow Six Vegas and Splinter Cell: Conviction. Prior to joining Ubisoft, Alex developed a broad range of experience within game production, building on his early roots in quality assurance and game design. He has produced licensed-based titles, working in partnership with Warner Bros. Interactive, Take 2 and THQ. Alex has worked in the video games industry for over 15 years and holds a Masters in Communications.

Making Games Let's assume a new part of the »Assassin's Creed« or »Far Cry« series is in planning – how do you kick off a project like this? Do the studios pitch or put forward ideas for the game, or do you assign the production to a studio?

Alex Parizeau It's not necessarily about pitching. What we're trying to do is to establish collaboration as a philosophy for creating great games and great teams by building long-term relationships between the studios and brands. One of the key elements for successful collaboration is building up trust. In Toronto, e.g., we've been involved in the »Far Cry« series and have just finished a project. When we started the collaboration it was all about building the expertise and skills it takes to support the main studio in the best possible way. This process takes some time, but at the end of the day it brings more value than to simply work on that one project and then switch to the next. So it's more about building strategic partnerships between teams that we see evolve in the long term as the studios are growing and as the brand is evolving.

> »The collaboration with other studios is a strategic decision.«

Making Games How do you eventually determine which studio will lead a project?

Alex Parizeau It basically depends a little on where those brands were originated. Let's take Assassin's Creed: It started in Montreal and later Quebec joined supporting the team here and there; so because of that long-term partnership they eventually got the chance to make their own »Assassin's Creed« game. Our partnership with the Far Cry brand started with Far Cry 4, so now we were also involved in Far Cry Primal, and obviously we would like to be part of the brand on a long-term basis. The collaboration with other studios is a strategic decision and depends on which teams are interested in working with you or can offer experience, but also where we see a long-term fit.

Making Games Once a production has kicked off, how do you coordinate the many studios around the globe?

Alex Parizeau That's obviously a rather complex task and something we developed over time. The way it is typically set up is to have a lead team in a studio, and usually the lead

time will have a team dedicated to support other studios. For example, they make sure that communication is running smoothly and questions are answered quickly. Sometimes we have collaborations between teams in different time zones, so it can be tricky to coordinate. Maybe the most important part is the very beginning when we start to decide what the different teams will be responsible for; trying to identify a good, strong and interesting mandate is the key as we found out over time. It's about finding the right balance between something the team can be motivated by and can take full responsibility for, where they can be creative, express themselves and bring something to the game. On the other hand, it needs to be something the leadership team can still oversee, making sure that a title in the end still matches the overall vision of the game. We have experimented with a number of approaches. When you try to be more directive and make the teams feel like an outsourced agency, it doesn't work so well. Instead we try to entrust each team with a task that makes them feel responsible for something that is significant in the game. This motivates the team members and boosts their creativity.

In Far Cry 4, e.g., the Toronto studio was in charge of the Shangri-La missions and the tiger. The team came up with a lot of ideas how to expand the use of such a beast in the game, something we later also discussed with the »Far Cry Primal« team. They gave us a lot of room for experiments and creating prototypes, which eventually led to the beastmaster elements.

> »Making a studio feel like an outsourced agency doesn't work so well.«

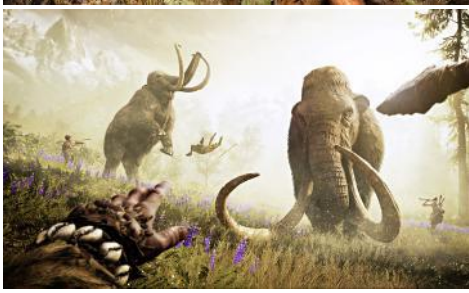
That's only one example of how a game can benefit from granting the individual teams enough space for creativity. But as I said, the usual set-up is to have a lead team, a collaboration team within that lead team that connects all the studios, and then within the studios we are split up in smaller teams of about 60 to 100 people that work on something very specific, but still with plenty of room for expression. And then we're basically going back and forth, working on milestones, communicating and making sure we have the best success.

Making Games Aside from AAA productions, how easy is it for studios to pitch ideas and concepts for new games?

Alex Parizeau Ubisoft is a company that embraces innovation and is usually very receptive to ideas coming from the teams. Obviously we have the big brands, but there is also room for other projects – unfortunately, I can't talk about all the things we're doing here in Toronto. We have a lot of opportunities for teams to propose ideas, some of which we embrace and others we actually end up working on. That's not only the case in Toronto, but across the entire Ubisoft family [see article about Fun House on page 37]. Some of those ideas remain smaller projects, others we expand to become bigger projects like, e.g., »Grow Home« which was created by our studio in Newcastle. There are a lot of projects like that across all studios and genres.

Making Games What does the organization within a studio look like? You mentioned that Toronto worked on Far Cry Primal which is now







Ubisoft's Toronto studio has been involved in the development of the Far Cry series since Far Cry 4 and was hence also part of the Primal team since Ubisoft puts strong emphasis on collaborations between studios and within brands.







The Toronto office has currently about 400 employees, but is expected to grow to approx. 800 people within the next five years.

finished. Will those team members who were involved in this project now be shifted straight to the next one?

Alex Parizeau Usually, a few months before the end of the project, we have conversations with all of our employees where we present them with some options they could be working on next or projects for which we would need their expertise. So our employees always have the chance to choose what they're interested in.

»Ubisoft is a company that stimulates innovation.«

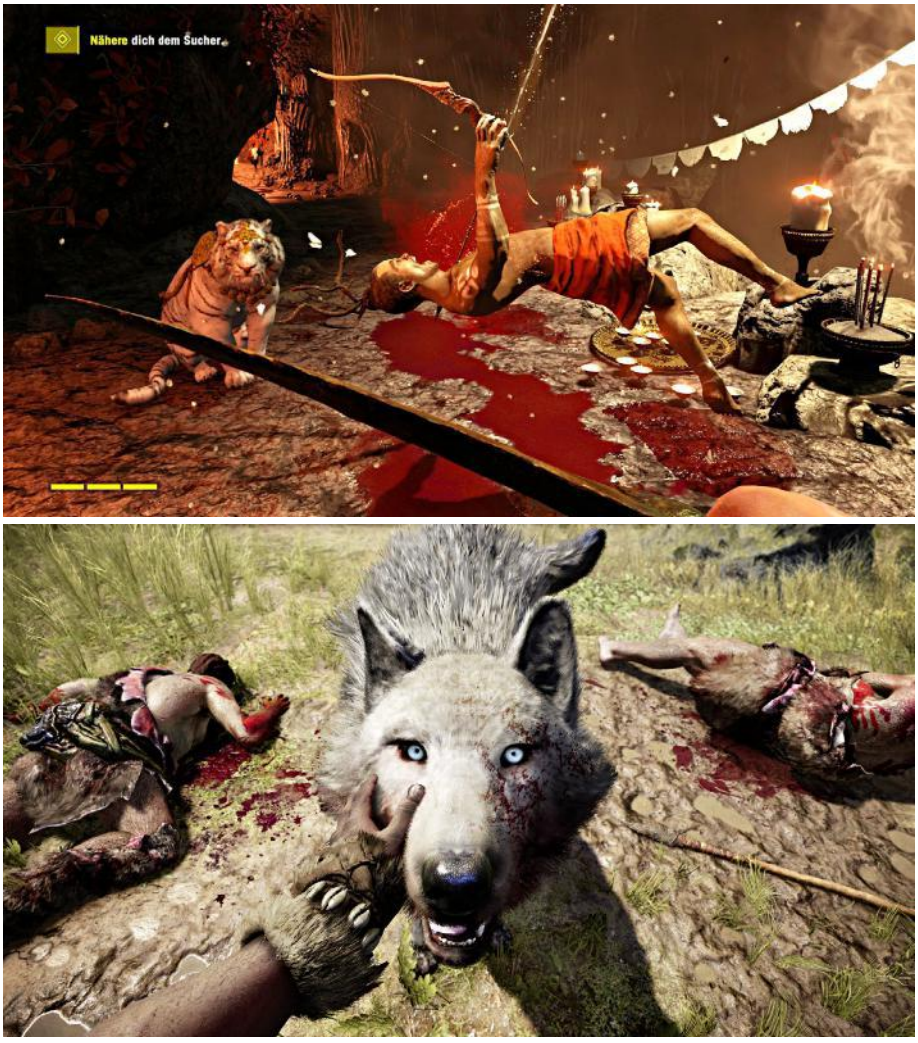
Making Games Does that mean there might also be an option for an employee to spend some time, let's say two months, to work on an idea of their own?

Alex Parizeau Usually, that's not even necessary since there often is a down time of a month or a month and a half when you join a project, and that's the time people use to propose their ideas. Others use the time to train certain skills or develop their skills in some way, and that's something Ubisoft is happy to support. We encourage our employees and allow them to use the gap between projects to invest time in themselves – no matter if they want to learn something new, build prototypes based on their ideas or even improve tools they have been frustrated with during a production because a program didn't work the way they wanted, and then they make that their project during the down time.

Of course we can't invest in every idea, but others we just have to adopt. That's the problem with great ideas: If we think they're cool, we have to do them, and that usually means investing one or two years to complete them. But obviously we can't always start new projects, so again it's difficult to find the right balance. But at the end of the day I'm convinced that Ubisoft is doing a good job granting employees a lot of freedom to put their ideas forward, discuss them and give chances to those that have the most potential to be completed and published.

Making Games The studio in Toronto currently has about 400 employees. What's your thought on how big the studio should become? Would you like to see a second Montreal?

Alex Parizeau Well, Montreal is a lot bigger than we are. True, we currently have about 400



In Far Cry 4 (top) the Toronto team was in charge of the Shangri-La missions. The tiger they created for it was further developed into the beastmaster features for Far Cry Primal (bottom) - a part of the game that was entirely created in Toronto.



Ubisoft has studios around the globe which makes collaboration difficult. For that reason there is usually a lead team in charge of the other locations, overseeing respective tasks.

employees in Toronto and we're still growing since we are working on a lot of projects and see a lot of potential for the studio. So we currently expect to grow to 800 employees by 2020. That's basically our 5-year plan.

The entire aspect of how we work with other studios and how knowledge is shared with events like UDC is extremely exciting for us. Since we're a lot smaller than Montreal, employees who start working for us have the chance to very quickly switch from one brand to the other, test different engines and pipelines etc., and that's a great benefit for our teams. Sometimes people join our studio who had already worked on Splinter Cell, Assassin's Creed or Far Cry and share their experience with our teams. Our employees also meet a lot of people from the Ubisoft family. That's of course good for the personal growth of every individual and it gives us the opportunity to combine this experience and use it for our own projects at a later stage.

Making Games You just mentioned UDC. In your opinion, what is the biggest benefit of sending employees to UDC instead of, let's say, GDC?

Alex Parizeau One of the biggest, obvious advantages is the fact that it is an internal event. There's absolutely nothing that needs to be kept confidential. We can be completely open about everything: What we're working on, which challenges we're trying to complete, and the speeches, too, can be much deeper. Another benefit is that often we are involved in the same brands or projects and can therefore discuss things we're all concerned with. We are very ambitious about open-world games and other very specific things and problems we try to solve, which is why it's a great opportunity to meet here and do it together.

Of course, with all those collaborations going on between the studios it also helps to see somebody face-to-face from time to time since it boosts the relationship. If you go to GDC, you obviously meet a lot of people, but you usually don't work with them on a regular basis, while here you meet people you are involved with for the next few years, working with them every week and every day.

Making Games But don't you think that it might slow down the solution process if thousands of employees could potentially add their opinion to the discussion after UDC?

Alex Parizeau Why should that slow down the process? Usually the opposite is the case: it inspires people because a speech might trigger one idea or the other which nobody thought of before. For example, when I attend GDC and see something which I might not even agree with, it always makes me think about it and I try to figure out what my opinion actually is, or I start talking about it to others. And that's what I find to be extremely motivating and energizing, and I think that UDC serves the same purpose. Here you are surrounded by a lot of competent peers who inspire each other

»Our employees always have the chance to choose to do what they're interested in.«

or, if they don't agree, challenge the other ideas or opinions, which might again produce new ideas. I'm convinced that it's critical for innovation that this process happens. We need people to share their ideas and discuss them with others.

Making Games You have been part of the Ubisoft family for about ten years. How do you think the work processes have changed during this time?
Alex Parizeau The size and scope of projects has increased significantly, also in the industry in general. Teams got bigger and that's why collaboration became a big thing. But that's also something that makes our productions so successful.

Making Games Would you say that the company culture has changed as well?
Alex Parizeau That's hard to tell. I think that Ubisoft is still quite the family business. I don't know if you have ever met Yves or Serge. Anyway: Even if the company has grown, these two still look at every single project and are part of our day-to-day work. We talk a lot to both of them. And that focus on quality and the importance of having the best people, sharing expertise and stimulating innovation inside the company very much comes from them. And I think that's what motivates people and makes Ubisoft so successful. It's this core culture that defines Ubisoft and makes me love working for this company. I don't think this has changed a lot in recent years, even if the projects and teams have grown. When we first built Assassin's Creed and there weren't a lot of open-world games yet, we were pioneers. Today, players expect top-quality and innovative games, so we constantly have to push the boundaries of what's feasible. So it's more the market that has changed rather than the company. Today, there are indie studios creating really innovative projects, other competitors pushing open world, etc. It's a completely different business than eleven years ago when I started. Games are everywhere these days and are much more relevant, which makes it even more motivating to work in this industry.

> »We encourage our employees to use free time to invest in themselves.«

Making Games Speaking of changing markets: What is your personal opinion on VR? Do you believe it is going to be the next big thing, or is it just hype? Or what do you think it takes to actually make a breakthrough this time?
Alex Parizeau That's the million-dollar question. Everyone is wondering if VR is going to be the next big thing. I have rarely seen more passionate debates about a technology, so I'm assuming that VR is going to be relevant in one way or the other. But the actual question is which place it will take in the entertainment industry since it can't be all about games, but more how VR can be used for a broader audience. I think VR is going to have a slow build because people are waiting for the devices to become available so they can experiment and play with them. I don't think though it will replace traditional entertainment as we know it, but rather find its own niche where we will get to experience incredible things. Wearing one of these headsets alone – this feeling of presence – is just unbelievable. The space around you disappears within seconds, and that's something you don't feel with the traditional types of entertainment. We're trying as well to produce immersive entertainment and give the players the feeling of living in a different world, but VR is the missing piece here. At the same time the technology poses a lot of new challenges, especially with regard to the controls and the movement within the virtual world. I think a lot of problems around motion sickness that still exist today are related to input methods, and it's up to us to find out not only how to give the player the feeling of presence in the game world, but also to find ways to actually create this presence. So far, nobody has found an answer to that yet, so we still need to experiment a lot.

Interview: Sebastian Weber



Even though Ubisoft's studios are mainly working on big brands, employees are always encouraged to put forward ideas for new projects which might actually make it into production like, e.g., Grow Home.



FUN HOUSE – UBISOFT'S LABORATORY FOR GAME CONCEPTS

The Fun House division was originally meant to develop casual games but Ubisoft Montreal quickly decided to open up the division. Here employees explain what is great about this innovative approach.

Readers may already be familiar with the Fun House, launched last year at Ubisoft Montreal. Fun House is a unique space within the studio, a laboratory for original casual and hardcore game concepts, built by small teams. Essentially, it's a space for teams to try out an idea, prototype, push forward quickly, get things wrong, make things better, and hopefully come up with an exciting new game experience. We caught up with the team to find out how things have evolved over the past few months.

Patrick Plourde on why to embrace Failure and why Gameplay Mechanics are most important

As time goes by and projects move forward, we're learning a lot! We're still riding a really positive wave within the company, and it's really inspiring to go to work every day! No one project is the same, and because we're focusing on being first to new tech, we're playing with ideas and concepts that feel like they're five or ten years in the future. We've been trying to understand how many projects we can handle at the same time, how to enhance our processes and creativity, but also, what makes a project a Fun House project.

We're experimenting and learning on all these fronts. So far, our major takeaway is the fact that we have to focus on gameplay first. We want to create fun games, therefore we need to make sure our gameplay has a strong identity and depth, and I'm willing to delay, postpone, or restart the conception of a game if the team needs more time to find that. Graphics, engine, story can be built around the mechanics, but they need to be strong first and foremost. We're inspired by how Nintendo makes games in this sense. »Eagle Flight« is our perfect example: The team was really challenged to develop strong gameplay mechanics, and I think it was a good strategy since people really love the game so far.

At the Fun House we embrace failure, it's something that allows us to learn, to build better projects and to improve next time round. In terms of learnings that can be applied to other projects at the studio, I think it's too soon to start giving lessons. It's important to remember that constraints on Fun House projects are really different to AAA projects, so even if we are able to reach conclusions, they can't necessarily be applied in a straightforward manner. Also we need to ship several projects to have visibility on projects in their entirety, as of now we only have projects in development.

Our main focus today is on making the Fun House a successful initiative that is here to stay, and develops many games. There are five projects underway right now, two of which are in VR (Eagle Flight and the partnership



Patrick Plourde
is Vice President
of Fun House
at Ubisoft Montreal.

From 3D artist to Game Design and Creative Direction, Patrick grew a lot at Ubisoft Montreal over the past 12 years. He has had the pleasure of working on bestsellers and critically acclaimed games like »Assassin's Creed II«, »Rainbow Six: Vegas« and »Far Cry 3«. In 2012, he decided to develop a new IP based on his love for Japanese RPGs and he created »Child of Light«. Eager to foster creativity and exploration, he is currently leading the Fun House. His mission is to shape an environment where small and offbeat projects can flourish and turn into vibrant new games.



Caroline Martin
is Executive Director
of Fun House
at Ubisoft Montreal.

Caroline joined Ubisoft Montreal in 1998, and quickly became Head of Production Marketing. Putting into action her Master Degree in customer research, she founded the Playtest Lab, a small revolution in the studio. She then produced some of the most daring games of the studio, including »Scott Pilgrim vs The World: The Game« and »Child of Light«. As Executive Director of the Fun House, she proves every day that small can be big by providing to in-house creators the environment they need to shape their dreams into successful games.





One of the main focuses at Fun House at the moment is VR as the division is used to work in small teams and iterate fast. That helps to gain knowledge about this new medium.

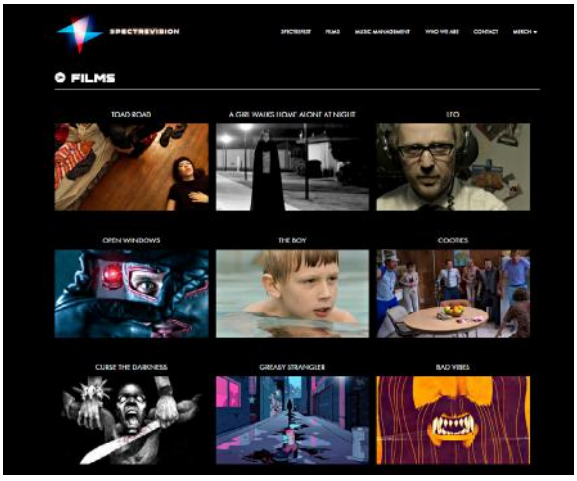
with Spectrevision), whilst the others are in completely different fields. We're developing stand-alone full products, so no demos or mini-games. I want us to be able to touch everything that is new or radically different. I'm happy with the portfolio we're building and people should expect to see a couple of games from the Fun House every year.

Caroline Martin on the Advantages of small Teams and Sharing of Knowledge

The Fun House team today is composed of a real mix of profiles, although we're still waiting for our first accountant to pitch us an idea! There's a distinction to be made between people who pitch their ideas to the Fun House and people who work at the Fun House.

In the first group we have different profiles, not only game designers or creative directors: we also have a programmer, an art director and a digital business specialist. It's actually very stimulating to see that each of them has their own vision, their own analysis of the current industry and their own way of changing it. Each time it's like we add another color to the Fun House palette. Regarding the people who work on these projects, the key word in their profiles is versatility.





For their VR ambitions, the Fun House employees have well known support: In January, Ubisoft announced to have partnered up with movie experts Spectrevision (amongst others founded by Elijah Wood) to develop a new game.

During the R&D phase we want to be quick, to prototype a lot, to learn from rapid iterations, so the smaller the team the faster it goes. Even during the more regular production phases, we still keep the team as small as we can, so a modeler or animator or an audio or game designer is always welcome!

So far the reaction from the other teams in the Montreal studio is very positive. Over the last year, around 150 employees have contacted us to know more about Fun House, but mainly about the best way to pitch an idea, so there are plenty of creative brains around us! Some people were just curious, lots are still working on their ideas, a few have pitched ... It's always great to see what they have in mind and how it can potentially become something amazing. It's up to us to keep working on our tools and visibility to make sure pitching is simple and we can have more and more great ideas.

Sharing what we do is a pillar of the Fun House; it's the way we work: sharing of knowledge, tools, tech, people ... everything! We learn much faster when we're open to others' experience, and collectively we have an amazing amount of knowledge! So we're completely open, talking a lot about what we are doing, just in case somebody might be interested or knows someone who knows THE thing we need!

We have a few regular events, like playing sessions every few months so people inside and outside the Fun House can try and give feedback on all our prototypes. We also have monthly meetings where each Fun House team presents to the others, so everybody knows the good and (I insist) the bad things that have been done. We push for sharing across the different teams at Ubisoft a lot, using all the different sharing tools and activities in the company.

Why Teamwork is important and serves the Players

Kim Parenteau works as an illustrator at the Fun House division and explains: »I see the Fun House mandate as working on creative,

Eagle Flight



Eagle Flight is a game that sends players to the skies of Paris to experience the freedom of flying and aerial adventures. The game is set 50 years after humans vanished from the face of the Earth; nature reclaimed its cities, leaving players with a playground in Paris. The game will be released for PlayStation VR, Oculus Rift and HTC Vive in 2016.

Future Projects

Beside of the already known or even released games, Fun House is working on four more projects currently. One is developed together with Spectrevision and is said to be a horror game in Virtual Reality.

UBISOFT
FUN HOUSE
MTL

Child of Light



Child of Light is an RPG inspired by fairy tales. You play as Aurora, a young princess with a pure heart whose soul is brought to the kingdom of Lemuria. There the player embarks on a quest to recapture the three sources of light, defeat the Black Queen and restore the kingdom of Lemuria. The game was released for PlayStation 4, Xbox One, Wii U, PlayStation 3, Xbox 360 and PC in 2014.

Valiant Hearts



In Valiant Hearts, the lives of four characters are inextricably drawn together over the course of the game. Friendship, love, sacrifice and tragedy befall each one as they help each other to retain their humanity against the horrors of World War I. The game was released for PS4, PS3, Xbox One, Xbox 360, PC and iOS in 2014.

Grow Home



In Grow Home, players are free to explore the strange, open world full of floating islands, majestic caves and waterfalls, all set in a minimalist artistic style. Players take on the role of BUD (Botanical Utility Droid), an excitable, child-like robot sent on a mission to search the galaxy for a new species of plant that can oxygenate his home world. By growing and directing the giant Star Plant into a towering organic sculpture, players reshape the planet and climb to new heights. The game was released for PC and PS4 in 2015.

original brands, that will bring new experiences to players. It's a creative platform with a solid structure enabling the development of BIG ideas. The cooperation between the different projects within the Fun House really makes for a sense of belonging. We are team players in a big family. The group of exceptionally talented, creative and mature individuals gathered here makes for an extremely enriching experience on all levels.

We are living a true team experience; this is the quality I appreciate the most working here.«

Marketing and PR can be difficult

»As for how we market and talk about Fun House products externally, marketing and PR techniques and methods have to be adapted and tweaked as the type of projects, the production timeline and the consumer target vary greatly. We are very lucky to have a strong and experienced marketing and PR network at Ubisoft, but we also have to manage the fact that all projects contain a huge part of innovation ... and innovation means the great unknown!«, says Claire Chabert, Brand Director at the Fun House division.

»Let's take Eagle Flight as an example ... The game's core is to let you experience the exhilarating sense of freedom of flying ... that is only feasible thanks to VR, but since VR is really an experience that people need to try to be convinced, it challenges how we communicate and how we show the game. This adds also with the facts that VR is very new, and there are not a lot of devices out there yet. There is no simple recipe when it comes to innovation, so we experiment with new things for our games, but also how we present them.«

Caroline Martin, Patrick Plourde



TRAVERSAL LEVEL DESIGN

Uncharted, Tomb Raider, Assassin's Creed – there are a lot of big franchises out there today that are doing great at traversal levels where players are free to move around the way they like. Travis Hoffstetter explains the principles that make great traversal possible.



Travis Hoffstetter is Lead Gameplay Designer at Visceral Games.

Travis is a Lead Gameplay Designer at Visceral Games in the San Francisco Bay Area. He has shipped »Battlefield Hardline«, »Tomb Raider«, »Transformers: War for Cybertron« and »The Bourne Conspiracy«. Travis received a Master's degree in Level Design from The Guildhall at Southern Methodist University and a Bachelor of Science in Computer Science from Saint Louis University. In his spare time, Travis likes to surf, eat breakfast burritos and make funny noises with his son.

In 2010, I started to work on the »Tomb Raider« reboot at Crystal Dynamics. Within my first two weeks, I was entered into a »Thunderdome« exercise, in which I had two days to revamp a traversal level, competing against a senior level designer who had worked on »Assassin's Creed 2«. A winner would be chosen after the time expired and that level would be used in the game.

Prior to that, I had never designed a level for a game using more than a simple jump mechanic where the players could only land on their feet. Tomb Raider allowed the player to climb, swing and jump in all kinds of different ways. You can probably guess that I lost, but what started was my education into the intricacies of laying out levels that use complex player traversal mechanics.

I define Traversal Gameplay as any gameplay where the player is actively moving around a space. Stepping up and down, hurdling, mantling, climbing and swinging are all examples of Traversal Gameplay (there are a lot more types!). Traversal Level Design is simply when you create a level using those Traversal Gameplay mechanics.

Traversal Level Design is an interesting level design topic because it is very niche and hard to get experience in, unless you actually join a team that is making a traversal game. Traversal Level Design is niche, because there are way more games out there with simple traversal mechanics than there are games with complex traversal mechanics and therefore there are few designers with the knowledge to share.

It's hard to get experience in Traversal Level Design, because most of the tools people use to learn level design do not come with free assets to build a traversal level and there is a lack of knowledge being shared freely on the best layout techniques. Most level editors, such as Unreal, Radiant and Source, have free shooting and cover gameplay mechanics to build a level around, not a large suite of climbing, jumping and swinging traversal gameplay mechanics. That means most people start making simple shooter levels when they try to learn level de-



Traversal Gameplay can be defined as any gameplay where the player is actively moving around a space.





A shooter level is much easier to create as the design is very open as long as main routes etc. are working. In a linear setup there is only one path for the player to go, which mainly is for action or story moments.

sign. Even if you google Traversal Level Design to try and learn layout principles, there is very little written on the subject compared to topics like combat level design or level design pacing.

In order to gain knowledge about creating fun traversal level layouts, a level designer needs to learn how to overcome a few new level design challenges and understand the basic principles of Traversal Level Design.

Traversal Level Design Challenges

There are two challenges that designers face when they switch to doing Traversal Level Design. First, the designer must learn how to work with level artists on a different style of gameplay space that has strong art requirements. Second, the designer must learn a complex set of metrics and rules for a lot of different gameplay mechanics.

Designers have to work closer with artists on sections of the level than they would on a traditional shooter like »Battlefield«. In a shooting level you define combat routes, landmarks, enemy spawns, etc. where you have walls and ceilings to define the playable space and act as cover. In a traditional shooter game, there is not too much back and forth with the level artists over walls and ceilings as long as the integrity of the play space is maintained. For example, in the Trailer Park level of Battlefield Hardline which I worked on, as long as the artist didn't make the trailer park walls stick out awkwardly into the gamespace or break any combat routes, they could do almost anything they wanted.

The opposite is true for traversal levels, where walls and ceilings often are the playable space. That means that the gameplay block mesh needs to be better integrated with the final look of the level. The designer has to understand that final look and build closer to it, otherwise their work will not make it into the shipped game. For example, if a designer needs to make a climbing path along a building, they have to make sure their climbing path is placed on architecture that makes sense for the building. Otherwise the artist cannot make it look like a building and the level will not get built. Notice how the traversal primitives (roof edge, brick tops, shutter) on the building in »Assassin's Creed: Syndicate« in **figure 1** make architectural sense and create a good climbing path.

Second, designers have to build their level according to complex traversal metrics, which creates an extra hurdle to get over. Just like how cover mechanics in games have metrics for size, shape and space between, so do all the traversal mechanics in the game. Let's take ledge climbing for example. The designer needs to know the minimum width of the ledge for the fingers to fit, the space between other ledges to grab and climb onto, the valid distances to jump up to, the valid distances to jump horizontally to and the height the ledge must be off the ground. In the screenshot from »Uncharted: Golden Abyss« (**figure 2**), notice the spaces between the



Figure 1: The traversal primitives on the buildings in Assassin's Creed: Syndicate make architectural sense and create a good climbing path.



Figure 2: Ledges in Uncharted: Golden Abyss are laid out apart according to specific distances so the climbing mechanics function correctly.

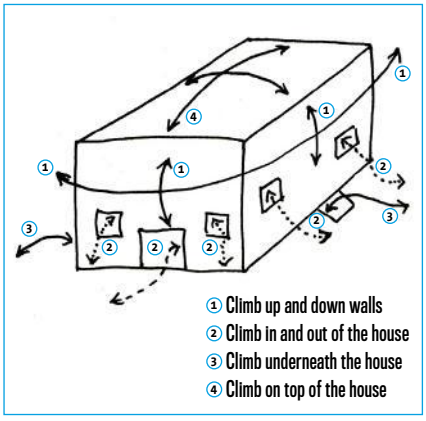


Figure 3: A good traversal level features numerous paths for the player to use.

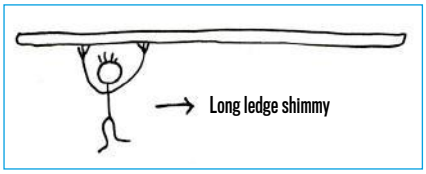


Figure 4: A ledge like this would not be very interesting for the player to climb.

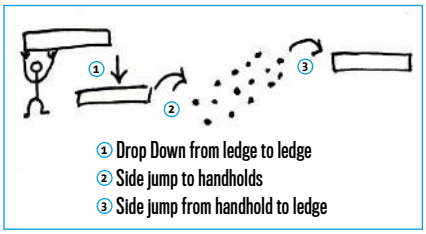


Figure 5: Breaking a long ledge and inserting some climbing, spice up gameplay for the player.

ledges. These ledges are laid out apart according to specific distances so the ledge climbing gameplay mechanics function correctly.

If you think about just the common traversal mechanics a big AAA traversal game has: step up and down, hurdle, vault, mantle, jump, ledge climb, hand hold climb, drain pipe climb, pole swing and balance beam; that's ten mechanics each with around four different metrics making 40 individual metrics to keep track of. That is a lot of metrics just for traversal and doesn't include cover and any special mechanics the game might have. Because of the amount of metrics, it takes a while to learn how they all work together and what the best mechanic combinations are. That leaves a bigger ramp up time for the designer to become productive than a game with simpler mechanics would.

Once you start creating levels using the mechanics and working closely with your level artist, it all becomes much easier with time and practice. It's just that entry into the design style that is rough. After understanding the special challenges of Traversal Level Design, next let's define the types of traversal levels that can be created.

Types of Traversal Level Design

There are three types of traversal levels: Linear, Open and Enemy Encounter traversal levels.

Linear

Setups where there is only one main path. These can be action moments, story moments or just a simple down beat in the level intensity ramp. Think Nathan Drake climbing up a cargo net hanging behind a plane in »Uncharted 3« or climbing up a radio tower in »Far Cry 3« or »Far Cry 4«.

Open

Setups that have multiple branching paths. Open setups allow the player to choose one of many branching routes to get to the objective. Examples include hubs in Tomb Raider or any of the big cities in the Assassin's Creed series.



Figure 6: An example from Assassin's Creed: Syndicate where the player is not locked into a straight path while shimmying along the roof.

Enemy Encounter Traversal

Either an open or linear traversal path that has enemies in it. When enemies are present, you need to adhere to a few extra rules to make sure their behaviors play out correctly and the space allows for smooth fluid engagements. Think of bounty hunts or assassinations in the Assassin's Creed series or any of the Tomb Raider and Uncharted levels that use lots of verticality and combat.

General Principles

Principles that can apply to any of the three above types. My focus is on third person and first person games, some of these may not apply to side scrolling games.

1. Utilize 3D Space

When I lost in the Thunderdome exercise, my big mistake was only making a path the player could climb up, down, left and right. Great traversal setups let the player move around a space, inside the space, on top of the space and even under it, like in **figure 3**.

Think of a cabin in the woods. For it to be »great« to traverse around, you need to be able to climb the outside walls, into the cabin, out of the cabin, onto the cabin roof and even under the cabin through the crawlspace.

Utilizing all the space makes the setup feel more natural, encourages choice and promotes replayability. Even if a path is linear, utilizing the 3D space will freshen up the gameplay path and make it feel more exciting.

2. Variety is the Spice of Life

Most games with traversal these days have a lot of traversal mechanics: ledge climbing, hand hold climbing, drain pipe climbing, pole swings, etc., etc. The last thing you want to do is use the same move over and over again when there is such a large bucket of moves to choose from. A general rule is to never use the same traversal mechanics more than two or three times in a row. Please note, this is a general rule. Certain mechanics like monkey bars are fine to use more than that because the move is so quick. When in doubt, feel it out through play testing.

Another general rule is to not go in the same direction for too long. Another way to think of it: »Is the player pressing the same inputs over and over again in my level?« An example of this is a ledge the player is shimmying along. A designer could layout a long straight ledge that takes thirty seconds to scoot down, like in **figure 4**.

That can be very dull. Spice it up by breaking the ledge and making the player go up or down to continue. For example, in **figure 5** we can break the ledge and insert climbing between ledges and handholds as well up and down movement. By taking one action and breaking it into six different actions, you get a more varied and fun feeling to the path.

Open setups can have long straight paths because they are architecturally necessary.



Open setups like in the Assassin's Creed series have multiple branching paths, e.g. big cities to roam freely.



Enemy Encounter Traversal can use an open or a linear path and is defined by presenting enemies to the player.



The Last of Us uses the color yellow to draw the player along through the levels.



Places the player would fear in real life are great places for traversal in a game.

For example, the Assassin's Creed games have them all over the place, but it works because the player can break out of them at almost any time. This can be seen in **figure 6**, where the player is not locked into a long straight path while shimmying along the roof.

3. No Vague Traversal Geometry

If the player can climb on it or perform any other traversal move, it needs to be crystal clear that it is a traversal path. If it looks like you can traverse something, players will try. If they think they can traverse and cannot, then they will get frustrated. This usually becomes a problem when a level goes from being block mesh to being arted. Keep an eye out for it yourself and in play testing.

Take a look at how other games deal with this. »The Last of Us« uses the color yellow in their traversal primitives to draw the player along.

Uncharted places special blocks for handhold climbing that are different than the wall they are placed upon so they stand out. Look at how the red handholds pop against the blueish grey wall in **figure 7**.

Most of the climbable ledges in Tomb Raider have an element of white in them for the same reasons.

4. Go with the Flow

When you learn the mechanics in your game, you also need to learn which mechanics flow or transition smoothly into one another. In other words, certain moves transition smoother into one another and finding out what those are and putting them together to form a path will create a more fluid experience. Those combinations can be taken and turned into small quick branches in a path or into a path itself.

For example, take a look at the screenshot from Assassin's Creed Syndicate in **figure 8**.

Notice how I can do a short jump to a balance beam, followed by a long jump to another balance beam and then a short jump to slide down the ramp. This is an example of a quick traversal path with flow because I perform these traversal actions in a row that fluidly transition into one another. These are most common in open level types because there is so much room for them but can also be used in linear types as well.

5. Forbidden is Desirable

The coolest places to traverse are the places we fear to be in real life. That's one reason why climbing on tall buildings is fun in the many games that have that type of experience now.



Figure 7: Uncharted places special blocks for handhold climbing.



Figure 8: An example of a quick traversal path with flow

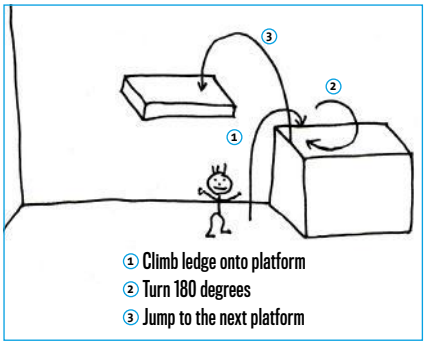


Figure 9: 180 degree turns should be avoided as they are not intuitive for the player who expects the path to go on in front of him.

When you need to spice up your level, add a dangerous feeling place for the player to go. You don't have to add extra gameplay for it to be special, just the feeling of being in that place will be meaningful and fun for the player.

For example, in »Batman: Arkham Knight« there is a level where the player must glide on top of a zeppelin. I spent ten minutes just walking around on top of the zeppelin, looking around and soaking in how cool it was to be a dangerous place with an awesome view. For me, it was one of the coolest moments in a game filled with awesome traversal moments and all I did was one glide move and then walk around.

Another similar example is the Synchronization Viewpoints in the Assassin's Creed series. You do the same traversal moves as you do elsewhere in the city, but it feels great because you travel up so high.

Naughty Dog is the king of these with examples like the train hanging over a cliff, hanging behind an air plane, climbing along moving trucks and trains. All of these are spectacular because they put you in a place that you fear in real life.



Figure 10: Shadow of Mordor embraced the concept of making the levels under contruction or falling apart.



Figure 11: An example from Shadow of Mordor where there is lots of room for the camera to move around the player.

6. When in Doubt, Move it About

Adding movement is a great way to make an aspect of your traversal more exciting. Something static in the environment can become a fun moment with just a little bit of movement. Uncharted 3 had a cool example with two chandeliers the player swings back and forth on in the Chateau level.

Assassin's Creed: Syndicate added a super fun element when they allowed the player to climb on moving horse-drawn carriages. The movement took what could have a been a simple set piece and turned it into an exciting adventure.

7. 180 Degree Turns are bad

When the player gets to the next step in a traversal path, try really hard not to have them turn 180 degrees around to continue along the path. It is not intuitive for the player's next path to be directly behind them, a lot of players look 180 degrees in front of them for the next path. Most of the time the 180 degree turn must be done in a small space, when a larger space is needed for the turn to feel fluid and smooth. For example, if the player jumps up to a ledge, pulling himself up, like in **figure 9** do not make him/her turn all the way around to make the next jump.

8. Under Construction

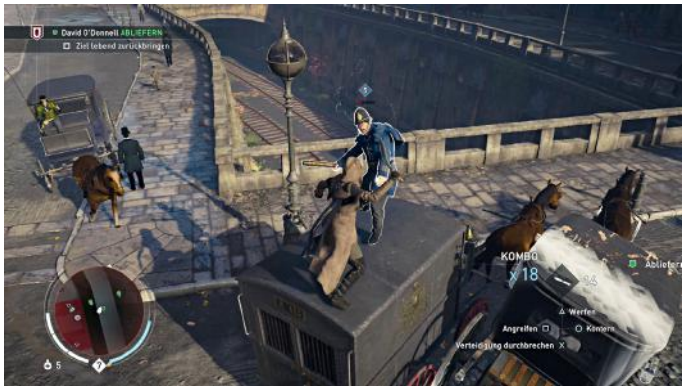
A time will come when you are stumped on how to set up a good traversal path. A super common trick to add easy traversal is to make the level you are working on under construction or falling apart. With either of these routes, it's easy to start adding elements into your path that would not normally be there but allow for fun gameplay moments.

You can see this all over the Uncharted and Tomb Raider series where you often climb on the ruins. My favorite example is »Middle-earth: Shadow of Mordor«. They totally embraced the concept and made everything run down and falling apart, allowing them to have traversal paths almost anywhere in the world they wanted them. That can be seen in **figure 10**.

Another example is in the beginning of Tomb Raider 9. A crashed, dilapidated plane to climb upon was added for variety and interest along a traversal path set upon a cliff side. The plane was a nice pallet cleanser after climbing on so much organic geometry. Because it was a crashed plane, we could put at the angle that would provide the most fun gameplay.

9. Leave Room for the Camera

The camera lets the player see all the cool traversal moves the character is doing. Problems with the camera mostly occur in third person games because the camera is on a track behind the player. Because the camera follows behind the character, it can knock into geometry in the environment. When that happens the camera can jerk or pop. Notice in **figure 11** from Shadow of Mordor how there is lots of room for the camera to move around



Climbing a carriage in Assassin's Creed: Syndicate makes a standard game situation much more exciting.



Web swinging in Spider-Man is breaking some of the rules of traversal gameplay but makes a lot of fun.

the player as they travel along the ropes, allowing for smooth camera movement.

Special Enemy Encounter Principals

These are specific to when there are enemies in the traversal level.

1. Quick Paths are Good

When the player gets attacked by enemies, they need fast traversal escape routes to allow them to get out of harm's way. A fast route would be a traversal path that the player may maneuver through quickly. If all you have are long traversal routes that leave the player exposed, then players will get frustrated because you are leaving them open to attack for too long.

Not all your paths need to be quick, but enough so that when the player assesses (makes their combat plan), they can easily find a quick way out. The Shanty Town level in Tomb Raider 9 is a good example of lots of quick traversal routes to get the player out of danger (see **figure 12**).

2. Allow Enemy Attacks Everywhere

Sometimes when a level is created or enemies are added to an existing level, the layout may have places where the player cannot get attacked when the enemy AI is aware of the player. This is bad not only because it makes the enemy AI look unintelligent, but also because it can break the 4th wall for the player. Also, the player does not have to move, breaking the basic combat loop of »assess, move, attack«. That breaks the intensity of the encounter.

A flushing behavior can be anything that harms the player and forces the player to move. The easiest solution in most cases is to modify the layout just enough so the enemy can throw a grenade or perform a flushing behavior to get the player out of the safe spot and back into battle. In Tomb Raider 9, I inherited a level with raised platforms certain enemies could not climb up and I didn't have art cycles to change it. If that enemy type was the last type alive, things could get wonky because the enemy did not have a behavior to force the player off the platform. An example of one those platforms in the level can be seen in **figure 13**.

Rules are Meant to Be Broken

Please keep in mind that these are all general rules and that special case situations or mechanics can change that. For example, the glide and grapple combination in the Batman series is one of the most fun traversal combinations ever made and feels great to repeat often. The same can be said of web swinging in »Spider-Man«.

These principles may apply to most of your game's traversal mechanics, but always feel through your layout and watch play testers to decide when something is a special case scenario. Thanks for reading!

Travis Hoffstetter



Figure 12: The Shanty Town level in Tomb Raider 9 is a good example of lots of quick traversal routes to get the player out of danger.



Figure 13: An example of platforms in a level from Tomb Raider that the AI could not climb.



THE CASE FOR STEAM EARLY ACCESS
HAEMIMONT GAMES' VICTOR VRAN

Haemimont Games explains why Steam Early Access is still a great measure to improve and polish a game if done right and shows how the community changed their Action-RPG within just a few months.



Boian Spasov
is Lead Designer
at Haemimont Games.

Boian Spasov is a lead designer of »Victor Vran« at Haemimont Games. His colorful developer history includes the design of a Roman RTS city builder; a Caribbean dictator sim series; a medieval cooperative action-adventure and a Prohibition era tactical gangster game.

Steam Early Access gets a lot of bad rap at times. Gamers are certainly justified to criticize incomplete or abandoned titles and developers who don't deliver what they promised. Still, our experience with Steam Early Access as developers has been nothing but positive and I am certain that it helped us make a better game.

»Victor Vran« was in Early Access from February 2015 till July 2015. Five months may not sound like a lot of time for a project that was developed for several years, but it was in those few months that the project grew more than in any other period throughout its development.

Can you have a proper Release after Early Access?

Nobody out there knew about our game when we first released in Steam Early Access. This was, of course, our fault – we should have started with the marketing earlier. If you have opened our Steam forums on the first day, you would have found four threads titled like »Confused. Is this game like ›Van Helsing‹?« and not one thread that explains what the game actually is.

Thankfully, Early Access gave us the time to remedy the situation and things were very different on our actual release day five months later. By that time we had thriving Steam forums with many players that knew the ins and outs of Victor Vran and were more than happy to welcome newcomers and »very positive« user reviews with 93 percent user approval creating a very different impression to people that discovered our game for the first time.

There is one myth I had heard repeated several times – that you can't have a proper game release after Early Access and that players, YouTubers and journalists tend to ignore titles that were previously released to the public. I can't speak for other titles, but this was certainly not true for Victor Vran. Our first few weeks after release were very strong both in terms of sales and press coverage and I think that the community contributed to this as several reviews commented that the Early Access players present the game very favorably in their reviews.

Communicating with the Players

As with any relationship, communication is crucial and perceived problems will get worse if you stay silent. This is particularly true for the Early Access community that consists of gamers who want to talk with the developers and help them shape the final form of their game.

We encouraged the whole design team to partake in forum discussions, comment on our reviews and generally talk with the players as



The inventory system was one of the most iterated on user interfaces, with a lot of features suggested by the players.



»Victor Vran« was meant to be a »Bastion«-like game but players of the Early Access version saw the game more like an traditional ARPG, so Haemimont changed the game accordingly.

much as possible, even in-game. The programmers also got in touch directly with any user that experienced some kind of issue that we can't replicate.

This was all very new to us and we have certainly made our share of mistakes, but in general we adhered to two simple rules that helped us avoid a PR nightmare:

- Never promise anything that is not yet set in stone.
- Never, for any reason, lie.

My single most memorable experience during our Early Access stay was related to a negative user review that was, for quite a while, the most upvoted one at our Steam page. Unlike most other negative reviews at that time, that particular reviewer was articulate, thoughtful and his criticism actually had a point. Instead of ignoring him, we replied to the review, thanked him for his constructive criticism and kept updating our reply as some of his concerns, like lack of character customization and in-depth itemization system, were addressed in our updates. Ultimately, he changed his review to positive and praised our development efforts.

So, here's my advice: Talk to your players, listen to them, act on their feedback, explain your decisions with logical arguments and be the engaged, communicative developer they deserve. Always keep in mind that they want the same thing as you – a better game. In the end, when your players know that you listen and respond, you will receive much more valuable feedback from them.

Steam User Reviews

The following reviews have been published by Steam users while Victor Vran was in Early Access. I think they are excellent examples why this time was so productive and important for our project.

Author: COCHISE, June 27th, 2015

Glad I bought it. Simple as that.

Edit: If there is any developer out there that deserves respect for timely updates as well as comunication on the forums it is without a doubt this one. Victor Vran has been a true delight. I know everyone is different so take my opinion with a grain of salt but I was hooked for 30+ hours over the course of a week. Currently just waiting for the final release but they tease me so with a whole new game mechanic! Transmutation!

Author: ZULDAR, July 8th, 2015

Victor Vran is to ARPGs genre what »Guild Wars 2« is to MMOs. The best »outsider« out there.

I've often been comparing those two games because the feeling is very similar. You feel that you have more freedom than the other games, a more clever approach of the combat system, smoother animations.

The weapons / cards / powers / consumables system gives a lot of possibility regarding the build you want to play.

There is also grinding and difficult content for people that like it.

The game has been evolving in a very good way since the release of the Early Access. A lot of stuff has been added and more is coming. The game has been updated once per week for the past month with a lot of playtime added.

The devs seem to read very often what we have to say about certain aspects of the game and they also answer most of the questions on a daily basis.

Overall, this is a very good game and I'm looking forward to the release.

Author: JOEYHAVOC, July 17th, 2015

First off, the gameplay is great. Fast, fluid, and plays equally well with a controller or keyboard. Well thought out class system that lets you play however you want, without being stuck in any one build. I love that someone finally added a little verticality to an ARPG. Graphics are nice and sharp. And now onto the developers. These guys update and improve this game constantly. I mean constantly. I've played games that have been out for five years that haven't seen the updates this game has. Kudos guys, keep up the good work. Also, if you should happen to read this review do yourself (and me) a favor and buy this game. The world needs more games like this.



Haemimont was concerned about the Slavic names of the characters. In the end the team kept the ones easier to pronounce by western audience and changed the rest.



Haemimont introduced the story near the end of their Early Access stay, when the game was nearly complete.

We imagined Bastion, they saw Diablo

Whatever expectations and plans you have before your game is live, Early Access will surprise you and change your priorities. People will see issues where you haven't expected, they will be passionate about new features that you haven't thought about or have previously dismissed.

We entered Early Access with the budget to develop several major features based on the feedback we receive. Before the game went public, we imagined it a title more similar to »Bastion« than to the mainstream hack-and-slash RPGs. It turns out, the users saw »Diablo« instead. Since the comparisons with hack-and-slash RPG kept coming, we decided to deliver what the user wanted, even if it meant reworking our core features.

We introduced a new point-and-click movement control scheme, similar to the one used in mainstream ARPGs as an alternative option to our normal »action« control scheme. We redesigned our leveling system from scratch because the players were not happy with the initial one. Our itemization felt too simple for a game of this genre so we introduced a crafting system and reworked several types of items (Demon Powers, Destiny Cards) to add more depth to the possible build variations. We added tons of quality of life improvements to the interfaces based on user suggestions. As we imagined a very different playtime and endgame gameplay initially, all endgame features we ultimately introduced were wholly based on user feedback and expectations.

Every single one of those decisions was received favorably by our community. Remember to pre-allocate the time for such features in advance – this flexibility will pay off in the end.

Early Access was also very good at telling us what wasn't worth our efforts. We tested the waters with a PvP multiplayer arena but since the vast majority of players showed absolutely no interest towards competitive gameplay, we didn't push further in this direction.

Pacing our Content and Feature Releases

Our first Early Access release contained roughly a third of the content and features that we internally considered »shippable« at the time. This was intentional – we wanted to focus the spotlight of the players' attention at the early game experience and certain core features like our major UI dialogues, the basic combat flow and controls.

We iterated on these features for a while, then introduced new features and content with the next major update and so on. Introducing new weapon types one at a time followed this philosophy – each weapon was individually discussed, evaluated, compared to previous releases and iterated on in the corresponding update. We used the same release approaches for the game world areas, monster



New weapon types were introduced one by one, each update keeping the spotlight of player and developer attention on the newest one.



Lightsabers may not belong in the gothic world of Victor Vran, but they were very well received as an Easter egg for the »Star Wars«-fans.



The players wanted Victor Vran to be a game like Diablo. Haemimont was happy to oblige.



Early Access gave Haemimont the opportunity to experiment with different types of rewards and loot drops.

families, level cap and new major mechanics, guaranteeing each will receive its fair share of attention and developer love.

This strategy produced an unexpected positive side effect. Our users were very happy that they received big juicy updates on a bi-weekly to monthly basis, containing content and features that couldn't realistically be created in such short timeframe. They had reason after reason to return to our game and see significant new releases since the last time they played it.

Beta Testing, integrated Bug Reporting Tools

We don't have a large QA team. For previous projects we have outsourced QA testing. For Victor Vran we relied on Early Access as our primary platform for testing the game before official release.

We added integrated tools for feedback and bug reporting in the game itself. Pressing the F1 button anywhere in the game opens a form to submit a bug or feedback directly to our database. Relevant technical information and a screenshot are automatically included in the report. Users even have the option to draw on the screenshots to highlight problems (yes, we received penis drawings; no, they were not many).

The response was immense. Closing duplicate and erroneous issues took some time but still there were tons of valid bugs and issues reported by our users during Early Access. In the end, Victor Vran turned out more stable and bug-free on release day than any of our previous titles that didn't have the luxury of an Early Access beta test.

The feedback and bug reporting feature was so useful to us that we kept it even after official release.

Early Access Cheat Sheet

If you consider Early Access for your game, here is a recap and some extra food for thought:

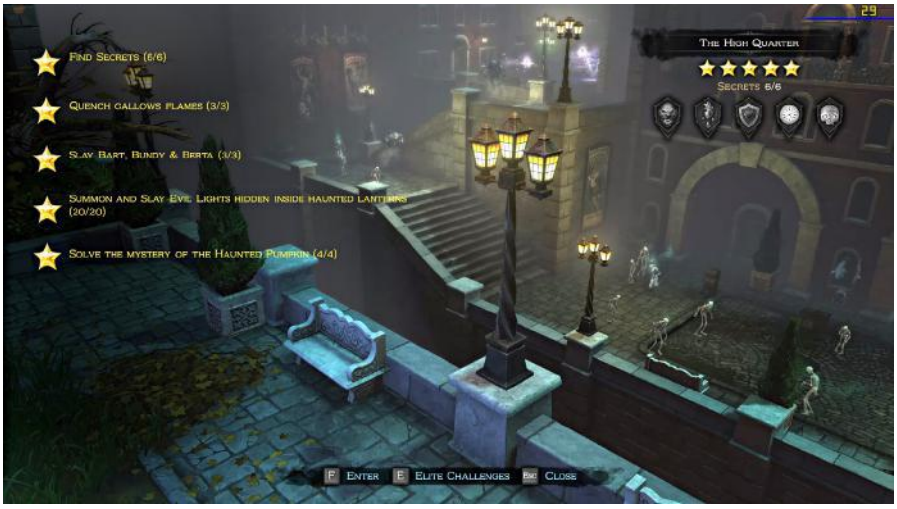
- Start with a plan. Release with a roadmap and share it with the community. Schedule some time to develop new features based on user feedback. Update the plan as priorities change and post the updates publically.
- Build the core of your Steam community. Your first players should feel welcome and a part of the development process; this is why they are buying an incomplete game, after all. Naming a boss monster after a particularly active community member may not be a bad idea.
- Talk with the community! Encourage the dev team to join the discussions, even if this costs them some time during work hours. Announce upcoming major features in advance to »test the waters«. Ask the players for their suggestions and ideas – »How do you think this system should work instead?«
- Establish a pipeline for receiving and analyzing feedback and bug reports. Volume matters here – if many players are bothered by an issue it is highly probable that it is a major one and you should address it promptly. Monitoring Steam forums is a must, but remember that not all players participate there. Integrated in-game tools for bug reporting, feedback or data gathering will expand this front.
- Identify marketing messages that work to use them near the all-important release day. Consider using your favorite Steam reviews as testimonials.
- Iterate fast and update often, even if this means breaking your game at times. Maintaining a constantly changing live game is harder than developing a project that is not yet released to the public, but the benefits are numerous.
- Don't be afraid to experiment – players will be more forgiving at this stage and radical changes will be accepted much more favorably during Early Access than after release. If you make a mistake, just admit it with the proper argumentation and revert the change.
- Read all user reviews under a magnifying glass, and pay particular attention to the negative ones. Try to provide a developer reply to each negative review, even the ones with objectively inaccurate claims.
- Watch how the players play your game. Play with them if the game supports multiplayer. Watch them on the Steam streaming service otherwise. You will discover tons of confusing mechanics, interface and usability issues that way.
- Have fun! The last few months before actual release when the game really »clicks together« are, in my opinion, the most rewarding period of the game development cycle. Experiencing them together with thousands of players while your game is live in Early Access can be truly special! **Boian Spasov**

About Haemimont Games

Haemimont Games is an independent games development team brought together by a common passion: to thrill, engage and provoke gamers worldwide. The studio is based in Sofia, Bulgaria. Since 1997, Haemimont Games developed 11 original titles and numerous add-ons.



Even the looks of the main character subtly changed. You couldn't imagine how many hours were spent discussing his name, beard and hat. And Haemimont still didn't get the name right at the end!



Challenges were so well received that Haemimont decided to introduce a set of Elite Challenges unlocked after main story completion.

IMPROVING USER HAPPINESS AND LOYALTY



With an ever increasing amount of users of mobile games, the demand for a solid customer support increases likewise. Nordcurrent's approach for this is to unify customer support and quality assurance.



Ricardas Jascemskas is Communications Manager at Nordcurrent.

Ricardas is a former journalist, with more than 10 years of experience covering games and tech industries. Since 2014, he is involved in Nordcurrent's social media, PR, community management and support activities.

Cooking Fever was released in August 2014 and since then amassed more than 80 million downloads. Starting with iOS, the game subsequently was released for Android, Kindle and Windows Phone platforms, adding not only users, but more complexity and the chances for unexpected situations. The in-house customer support team had to adapt to ever increasing numbers of users and mix of platforms, redesigning the processes and structures involved with helping customers and improving their experience with the game. Here are the main things which worked well and left more players happier with the game.

Two Teams in One

First and largest impact to overall process was having a united team of customer support and quality assurance. Nordcurrent has a long history of working with consoles, handhelds and mobile, which required a different approach to quality assurance, with direct customer support starting in the last mobile era. The solution to have a united team improves many aspects of customer support activities and speeds up the process of dealing with each situation.

It is especially evident working with technical issues, which occur despite of a huge amount of polish and attention to smallest details. As the same people run the quality assurance processes who after the release talk directly with players, it becomes much easier and faster to pinpoint the problems from mostly vague descriptions by customers and offer right solutions.

Working with Players

Second very important condition is to have a clear and comprehensive database of different possible situations and »rules of engagement« with players. The aim is to have as little steps dealing with each user complaint as possible, because with every email and every question sent to clarify the situation, he or she typically becomes less cooperative. So instead of only asking to describe the problem with text, it's really helpful to ask for a screenshot or even a video file. Of course, with typically less technical casual players it's a challenge by itself, but with detailed explanation, and, more importantly, incentive for them to help, it really works.

Of course the database would not be useful if it's not always updated, so each situation and solution must be properly logged and become a part of general knowledge and reference for forthcoming cases.

Improving the Game

Having both, a united team of quality assurance and customer support, and the always growing database of users' situations, translates into one



Feedback from players can directly impact game design changes, such as extra steps in the tutorial of Cooking Fever.



Many players are eager to offer their ideas for game updates, like the Juice Bar location in Cooking Fever.



The latest Cooking Fever update was based on FC Barcelona football club.

more crucially important step – improving the game itself. It's the old truth that players don't always experience the game the same way as developers do, especially when talking about a vast casual audience. So it's important to have the ability to smoothly implement the ideas and changes coming through the customer support.

The example of this could be the case with the Sushi level in Cooking Fever. After this content update was released, we started to get questions related to seaweed placement – some players could not see where it was. In the end, the best solution was to add an additional step to the in-game tutorial exactly at the point when the seaweed is introduced in the game.

Give back more than expected

In Lithuania there is a saying: »small gifts increase friendship« and it's a good way to improve the relations with users. It's not enough, to only solve the player's problem, as then he or she just gets back to usual mood and hopefully is not so angry or sad anymore. But to really make someone happy you need to exceed expectations. As it was mentioned above, when we ask our players for help figuring out the situation, we always compensate with extra in-game currency. But it's not limited to that – in any case when players are frustrated, even if it's not the game's fault, it's worth to cheer them up with some in-game gift. You could even offer your other games for free or with extra currency, if there is such an option. In such way the players would hopefully become more loyal and help spread the word about new games.

It's not only about the gifts, though. Usually, it's enough just to show attention and remember the players who were contacting you. For example, after Cooking Fever was released on iOS, a lot of people were asking when the Android version will be available. After the Android release, we contacted many of those users informing them about the release and got lots of positive comments – same for updates etc.

The Customer is not always right

Admittedly, there is an opposite side of having friendly relations with players and acting generously in various situations. There will always be a small number of people who will try to abuse such opportunities and try to gain unfair benefits. The usual situation in Cooking Fever's case is the claim that purchased in-game currency has not been received or has been lost due to some technical glitch. In most of the cases we immediately compensate that and give extra coins, but it happens that some players start to ask for this repeatedly. In such cases, despite making them angry and risking 1 star votes in app stores, we ask players for some kind of proof of purchase.

Of course, it would be best to have some automated solution for such cases, and currently we are preparing for the in-game currency data to be reported with each ticket to customer support. We'll see if this helps to smooth out this part of dealing with players.

Social Media and proactive Support

It is worth to consider adding social media to customer support activities. It has obvious benefits such as being publicly seen as the company caring for the customers and might also be an effective way to reach new players. But before actively doing so, it's important to answer several questions. First, how it is going to work alongside traditional channels. With Nordcurrent games we mostly use social media as an additional FAQ channel. If the problem or situation is well known and has a clear universal solution, then it's of course beneficial to list it publicly in social media channels. If not, then we just use it to direct users to our customer support page.

Another important thing to consider is the expectations of social media users regarding the customer support there. Various surveys suggest that customers expect faster responses on social media than through email. For example, Lithium Technologies study indicates that 53 percent expect to get the answer in less than one hour and 73 percent during the same day. But the extra effort and resources put to social media might be worthwhile, as then it becomes possible to actively monitor and react to customers responses online, which might help identify possible problems and situations in need of attention earlier than just relying on traditional email based support systems.

Ricardas Jascemskas

Nordcurrent

Nordcurrent was founded in 2001 in Vilnius, Lithuania, and for many years worked as a developer of handheld and console games. Since 2010, Nordcurrent moved to a mobile first approach and currently is developing and publishing casual and Free-2Play games. The studio is mostly known for mobile game series »101-in-1 Games«, »Happy Chef« and »Cooking Fever«. Nordcurrent has offices in Vilnius and Warsaw, with more than 100 employees. The company is also working with external studios from Central and Eastern Europe.



PORTING BASTION TO PLAYSTATION VITA

BlitWorks is specialized in porting games. But when the team started to adapt »Bastion« to run on PS Vita, they quickly realized that porting C# to C++ would need a lot of work and new proprietary tools.



Miguel Angel Horna is Founder & Tech Lead at BlitWorks.

Miguel Angel Horna started out in the videogames world after feeling curious about how the arcade machines, he used to play with as a child, worked in the inside.
His curiosity led him to the deepest heart of technology and he redirected this knowledge in the creation of emulators of all kind of gaming machines and consoles: from the mythical Model 2 to the MegaDrive or the Dreamcast.
In the year 2011 he co-founded BlitWorks, a videogame porting and publishing company. BlitWorks has become the trendy porting company between indies in just a few years, because of the impressive catalog of games that passed through their studio (»Bastion«, »Don't Starve«, »Super Meat Boy«, »FEZ«, »OlliOlli«, »Spelunky«, »Badland«, »Age of Zombies«, »Jet Set Radio«, »Sonic CD« etc.).



Once the team managed to get Bastion running on PS Vita, the game was running horribly slowly, at around 30fps.

Bastion is a game that was released for Xbox 360 originally and it was developed using XNA, .NET and C# technologies. As a porting company, we are used to receiving tough assignments, but this is probably the toughest one to date. We've learnt a lot of things that may be of interest to C++ and C# developers.

Most ports are isolated efforts, mostly unrelated to each other, where you start with the original code and when the game runs fine on the target machine, the job is done. This is not the case with the port of Bastion for PS Vita, as it was strongly related to our previous work porting C#/XNA games. Let's start with a brief review.

Three Years working with C# on PlayStation

It all started when we were contacted by Sony to port »FEZ« to PS3/PS Vita/PS4. After evaluating the options for porting a C#/XNA game to PlayStation, we opted for using a generic C# to C++ conversion tool which required a lot of manual assistance.

Meanwhile, Xamarin ported Mono to PS4 and we thought it would be a good idea to port Mono to PS Vita ourselves, so our next C# ports could be done without conversion tools. This sounded like an unbeatable idea.

Note that on PS Vita there was some kind of C# support via PlayStation Mobile, but it had a series of drawbacks like severe memory limits that rendered it completely unusable for most XNA games.

So we finally ported Mono to PS Vita and a while after demoing it with small games, we got our next port request: Bastion for PS4 and PS Vita. Wow!

We started porting Bastion to PS4 using Xamarin's Mono (this would need a separate Post Mortem), and we eventually succeeded. So we continued with the PS Vita port as planned, and this is today's story.

Bastion with Mono for PS Vita

Before trying to run Bastion with our Mono and MonoGame ports, we had only tested simple games with only one thread, and we were quite happy. But Bastion was a different beast and we ran into unexpected trouble:

Garbage Collector and multi-threaded code

Mono's Garbage Collector (SGen) is generational, but it's still using a »stop the world« approach. This means every now and then it needs all threads to be stopped in order to do its duties, but we were missing a very important primitive on the PS Vita for this code to work, so we started looking for alternatives. We first added a global flag so when the Garbage Collector wanted to run, all the threads would check periodically and would let the

Garbage Collector do its work. But when tested in real code, we found a lot of edge cases that turned into deadlocks.

At this point, our priority was porting the game, so we just removed the threaded functionality in Bastion and we were back to the safe and happy world of single threaded programs. After all, you never know what's coming next, so it's always better to have something running with known issues than having nothing running at all.

PS Vita memory limit

The PS Vita, being a portable console, has less usable memory compared to a non-portable one, so a very important thing we needed to do before seeing the first rendered frame on the PS Vita, was to rescale all the textures to fit in such limited budget. Luckily, Supergiant (the developer of Bastion) had their own asset pipeline we could hack in order to rescale the assets while repackaging them.

Sounded like it would be a straightforward change, but as we added the exact restrictions for the hardware, we soon realized that we were not getting the size reduction we expected and needed. By ocular inspection, we found a lot of texture atlases more than half empty. The original game had all the atlases nicely packed, almost using all of the available texels, so what had happened? The introduction of PS Vita's PowerVR texture constraints (square and power of 2) suddenly forced the atlas code to grow texture size four times each time it couldn't fit the sprites in a given texture size. So if the sprites couldn't fit in a 256 x 256 by a slight margin, the atlas code would switch to a 512 x 512 size instead, leaving a big empty space in there. The sprites are grouped by usage, so if we just filled the empty spaces with sprites from other groups, we would need to keep more textures in memory, which would actually defeat the purpose.

In order to regain this precious texel memory, we split the largely empty textures in smaller groups, as it's better to have two 256 x 256 textures instead of one 512 x 512 texture. This increased the batch count a bit, but it was worth the memory saved.

We also encoded music and effects in Sony's compressed audio format, which as a bonus put to good use with the PS Vita audio hardware acceleration, freeing up some precious CPU resources.

Performance

Once we solved the threading issues and managed to fit the textures in memory, we were finally able to see the game running on the PS Vita, for real! But our first impression was less than relieving. The game was running horribly slow, at around 30fps in the very beginning of the game which is the simplest scenario, with the added bonus of the stalls caused by the Gargabe Collector. Panic!

Time to fire Razor (the PS Vita's profiler) and see what was going on. No luck. All we could see is that one core was completely busy running the Mono entry point, meaning there was no evident waiting or sleeping. But the C# code was completely out of the Razor's reach, so it was not showing up nicely charted in the timeline as we were used to see any other game's code. This didn't come as a surprise, as we knew it would happen, but when you need to use a tool and you can't this is when you realize how much you rely on it.

We manually inspected the assembly output of the Mono AoT compiler and it looked like the code was completely unoptimized. We were unsure if Mono's JIT was doing an optimization pass on generated code that was omitted by the AoT, but the reality is that the performance we were getting was completely unacceptable.

Even worse, if we progressed slightly through the game, the memory would run out very quickly. All the overhead introduced by the .NET Framework was leaving the game with even less memory to run. The use of a »Compact .NET framework« on the X360's XNA suddenly came to our minds and it made a lot of sense, but there was no equivalent counterpart in Mono.



One problem the team did face was the fact that texture atlases were automatically increased to 512 x 512 if the sprites did not fit into a 256 x 256 texture, which created a lot of empty space in there, wasting almost half the memory.





The team was very proud of the performance they had achieved with the port of FEZ thanks to the use of C++ and the same great performance was their aim for Bastion, which lead to the creation of their own conversion tool.



After a few months of hard work and thanks to the self-made tool, Bastion was finally running at 60fps.



Bastion's code was big enough to completely discard the possibility of moving on without the possibility of debugging or profiling. Tough times at the office back then. All our effort porting Mono to PS Vita was for no good, and we didn't want to repeat the huge effort it took us to port FEZ.

But one of the things we were proudest of our FEZ port was the slick performance we attained, thanks to the use of C++. So as a desperate measure, we decided to try yet another approach: to create our own conversion tool. We had the knowledge on how to convert most of the C# constructs to C++, and this way we would be able to reuse this tool for other ports. This would be yet another big amount of work, but at least we knew it would lead to a viable solution. Supergiant were supportive enough to accept yet another delay on Bastion for PS Vita, so we started what would become our semi-automatic tool for converting C# code into C++ called Unsharper (www.blitworks.com/unsharper).

Unsharper

You may have heard about IL2CPP and it may look like Unsharper should be the same or very similar. Far from it, both are two very different approaches to solve the same problem. The key differences between these two technologies are summarized in the table in **figure 1**.

IL2CPP	UNSHARPER
C# ▸ IL ▸ IL2CPP ▸ C++ ▸ Executable	C# ▸ Unsharper ▸ C++ ▸ Executable
Garbage Collector	Reference Counting *(on steroids)*
Requires Virtual Machine	Doesn't require Virtual Machine
Complete .NET assemblies linked into the executable	Minimal .NET Framework linked into the executable

Figure 1: The table shows the key differences between IL2CPP and Unsharper.

As you can see, the first difference is that Unsharper does not rely on IL. This has a few very deep implications, most of them due to our focus on performance:

- Instead of relying on a VM, we're converting all C# features to straight C++. This strains the C++ compiler but also allows it to produce more optimal code as it allows for static analysis.
- Generated code resembles original C# code as much as possible, so you can debug with native tools. The C++ compiler has a lot of work to do, this means long compilation and link times and hence slower iteration through the game.
- Reference counting is less efficient than Garbage Collector, but does not produce hiccups which is something you normally prefer in a game.
- Reference counting is not able to automatically find and release cycles, this has to be solved at coding-time.
- Thread management on the PS Vita is much easier and safer due to the lack of Garbage Collector.

We're basically trading run-time for compile-time and coding-time.

The last difference is very illustrative of the different approaches followed by each project. Instead of using Mono's implementation of .NET Framework, which is quite big and can't be easily broken in small pieces, we implemented a small subset of .NET Framework in C++. This takes significantly less run-time memory and as an added bonus, it reduces the amount of C# features needed to be implemented, which is always welcomed. It's a completist approach versus a more specific and focused approach.

Bastion for PS Vita with Unsharper

So a few months later, we got Bastion for PS Vita running for the second time, in a much better shape.

60fps at the beginning of the first level was much more encouraging but of course, it still didn't mean that the port was going to be easy.

But before we proceed, let's explain how this speed boost was possible. The first thing to note is that game code is now properly optimized by the C++ compiler. There are no wasted registers and unneeded load and stores in the code. Also the code is much smaller, both because of the optimizations and because we're using a smaller framework, so the CPU cache is more effective. On the other hand, Unsharper uses reference counting for memory management, which is traditionally slower than a fairly good generational Garbage Collector such as Mono's SGen. Bastion is a memory intensive game, so we couldn't overlook this

weakness. But since the game was already converted to single thread when we were trying to run it with Mono, we could entirely ditch the atomic increment and decrement from the reference counting code, which is quite an expensive instruction on an ARM CPU. This way, our reference counting on steroids gets a comparable speed to a good Garbage Collector, and as a bonus, it does not cause hiccups.

The effects of not using atomic operations on the reference counting code can be seen in the table in **figure 2**.

Now that the game was converted to C++, we could put the PS Vita's profiler to good use. A profiling session would now show the game code instead of a black box filling a whole core. And interesting to note is that debugging was also possible as the code was now natively traced.

Performance revisited

The first thing we usually do once we get a port into a workable state is optimizing the game loading time, as it would affect the iteration time of everybody involved in the project.

The profiler showed a lot of time spent on XML loading, so we drilled down to see the cause. Our framework implementation used TinyXML, but the .NET API mapped quite poorly to it, so we just changed the C# source to be able to use a more TinyXML-friendly API. Then we used the profiler again to find which files were the slowest and translated them to a binary format so they'd be even faster to load.

It possibly took more effort optimizing the startup time than the savings it reported in clock time across all team members, but having long iteration times is so distracting that the benefits can't be just measured in seconds.

As we fixed issues and allowed the game to progress more and more, we started to find levels that performed poorly. Again, PS Vita's profiler came to the rescue and it helped us finding a lot of micro-optimizations that made almost all the game run at 60fps. The biggest gains came from fixing just two performance bottlenecks:

The first one was in the code for retrieving game data (weapon attack damage, for example), which was searching its name in a dictionary. That lookup was not particularly slow, but it could be very frequent and it was straining the cache, so we replaced the name for a simple index and mapped the dictionary to an array. This super simple fix was enough for this code to stop drawing attention in the profiler.

The second biggest bottleneck was trickier to fix. In the collision detection code, the game had to remember which elements had been already tested to skip them in some steps. That was done with a dictionary, too, and we tried to make a custom dictionary to speed that up (you can always beat the standard implementation of a data structure if you know how your data works). This solution improved the game performance, but it was not enough. Then we tried a completely different approach. Instead of saving which objects had been tested, we made the objects themselves remember that. It was a huge and somewhat scary change, but since it helped reduce the cache pollution, it worked like charm and that solved the remaining performance problems.

Memory Limits revisited

Although the game was almost absolutely memory leak free (something we discovered while doing the PS4 port), the converted code had a lot of memory leaks. This is because reference counting is not able to deal with object reference cycles, so that's a task that had to be addressed manually.

The first thing we tried in order to find the memory leaks was to use Visual Studio's Memory Usage Tool with the C++ version of the game running on Windows. Since this tool was tracking absolutely all the game allocations, it used tons and tons of memory and playing the game under those conditions was not very reasonable.

So we built a special debug mode in Unsharper's reference counting code where you could register certain objects to be tracked. Each time a new reference to that object was created, the entire call stack at that point was saved. When the reference was deleted, the trace was removed as well. Then at a strategic point in the code, you could print all the references that were still alive.

Mono's Cooperative Mode for SGen GC

A cooperative mode for Mono's SGen was unveiled late 2015 by Miguel de Icaza in his blog. This mode allows to support Garbage Collector in non-preemptive platforms. That is really good news as anyone trying to use Mono on PS Vita in the future will be able to skip the trouble we had with multi-threaded code. If we had been able to use the cooperative mode in Bastion for PS Vita, we would have certainly been able to improve the framerate by reworking the C# code, with the exception of the Garbage Collector stalls, which are unavoidable. However, as we weren't, it's up to the reader to speculate whether that would have resulted in a solid gameplay experience.
For more information please visit:
www.makinggames.biz/MonoSGen



	With atomics	Without atomics
Simple Scene	50fps	60fps
Stressful Scene	24fps	36fps
Really Stressful Scene	12fps	25fps

Figure 2: This table shows the effects of not using atomic operations on the reference counting code.







As an example of how we used this tracing mode, let's imagine we were to put a specific enemy object to be traced. We would play a certain level for some time and, after leaving it, we would print each alive enemy instance, alongside where the living reference was created. This would help us pinpoint the code that was responsible for the leak and break the reference cycle that was causing it using weak references.

Tracing an object was as simple as changing the Unsharper generated »make« function for its »makeTraced« equivalent:

```
// code generated by unsharper
reference_ptr<Unit> UnitManager::CreateUnit()
{
        return reference_ptr<Unit>::make();
}

// code modified to trace Units
reference_ptr<Unit> UnitManager::CreateUnit()
{
        return reference_ptr<Unit>::make_traced();
}
```

Then, at a strategic place you could print the info for all alive instances of that specific object:

```
void OnMapFinishUnloading()
{
        [...]
        // print all live references of traced objects
        debug_sharedptr_print_allocs();
}
```

Most of the time it was not obvious what kind of object was being leaked, so we needed to rely on Visual Studio's Memory Usage tool in order to find out a list of candidate classes. We just took two snapshots in strategic places, and after the ten minutes it usually took to compute the difference, we had a good starting point for manually tracing the suspicious classes with our helper code.

This combination of tools worked flawlessly to break loops on the C# side, but it eventually stopped working. Every memory leak was quickly narrowed down until one day we couldn't find one remaining memory leak that was driving us nuts. Then we relied on the unbeatable technique of commenting things out until we figured out the memory was being leaked in our PS Vita dependent C++ code (trophies, savegames, cloud saves), which of course was conveniently disabled on Windows. It may look obvious now in retrospective, but when you're dealing with a big codebase, it's easy to forget about that thin layer of code that it's just calling native APIs and has been battle tested on other projects.

Memory Limits revisited again

Looks like when you have the game running at 60fps without leaking memory, you're done with it, right?

The build started to be tested and the usual flow of issues came in response. We did our usual bug-fixing sessions until we stumbled upon the next headache. The testers were reporting crashes that we weren't able to reproduce. It took us a good amount of time (and several crossed emails) to figure out that those issues were only happening on the testkits, but not on the devkits (even with the memory set to resemble a retail console).

We managed to get a crash dump, and since the game was now in C++, it was relieving to know that we would be able to properly resolve the symbols of the stack trace. But shockingly enough, after inspecting the trace it turned out the crash happened inside system's malloc function, far away from the unsharped code. This was strange as we were using our own memory pool and a custom allocator, so malloc was not even used anywhere in our program. Then we realized it was the system libraries





The action role-playing game Bastion was originally released for PC and Xbox 360 in 2011. About one year later, the game was ported to OSX and iOS and was released for PS4 and PS Vita in 2015.

that were allocating some memory for their own internal use. It turned out we reserved almost all the available memory for our own pool, leaving the system without enough memory for its duties. Unfortunately, reducing our pool size would have been quite troublesome, because the game was already fitting so tightly that new memory issues would surely appear. This seemed a quite big showstopper, as we couldn't get more memory out of thin air, especially at the very end of the project. Then we remembered of a feature recently added to the PS Vita that allowed games to use more memory (at the cost of some acceptable drawbacks), so we decided to use it and our memory crashes fortunately disappeared.

In previous ports we had never stretched the memory to such an extent as to learn that the retail mode of a devkit does not behave exactly the same as a testkit. Yet another example of a bad assumption that could make you waste a lot of time.

Conclusions

C# is a super-good programming language, but you need full tooling support (debugger, performance profiler, memory profiler), otherwise your game turns into a black box. The only major drawback inherent to C# is its reliance on a Garbage Collector, which is a big issue when creating games.

We achieved the possibility of programming in C# with C++ performance, and without porting limits to any platform, with native tooling and without the Garbage Collector hiccups. The conversion is not fully automatic, but it's not as expensive as it may seem. Breaking memory loops seemed to be very difficult to do, but once we refined our technique it was not that hard. Building a tiny implementation of .NET Framework is not a big task as it may seem. Using the C++ Standard Library, and other common C++ libraries, you could quickly replicate the most used classes.

Knowing the hardware limits is always important, even if you are programming in a high level language.

Miguel Angel Horna





OPERATING LIVE SERVICE GAMES

PlayFab promises to be an all in one backend platform for games. Mark Val explains what the free service can do for game developers and how players profit from it.



Mark Val
is General Manager of PlayFab Europe.

Mark ran multimillion user online projects. Entrepreneur, award-winning developer with 17 years of experience. Produced and directed many projects for corporations, private and public organizations around the world. In the gaming industries he built studios for Ubisoft and Unity Technologies, while leading departments and pushing the limits for U.N.E.S.C.O., Marketel, Cossette and others. Clients include Mentos, Air Canada, Cadillac, L'Oréal Maybeline, Rogers and more than 40 others. During those years, he worked in Zurich, London, Boston, Vancouver, Montreal, Samoa, Singapore and now Berlin. Mark is responsible for expanding PlayFab into Europe, winning over both developers and partners.

High performance in online game development and live game operations is no longer optional. The best performers in our industry are not just developing faster, they're also delivering higher quality games with fewer failures while operating as a service, 24/7, across the globe. To compete in that market, you shouldn't be choosing between feature delivery speed and quality; faster features means more valuable feedback you can use to make your games more playable, more popular, and more lucrative.

To help you win the game of making games, we created PlayFab, the ultimate Live Game Operations platform. Since our public launch in mid-2014, we've been helping games like »Loadout« on PSN and »Adventure Capitalist« on iOS to iterate faster and learn more about their players, all on a robust, trustworthy platform that scales. As an added bonus, PlayFab is free.

All the APIs covered in this article are available for you to implement right away for free.

With PlayFab you can Build, Launch, and Grow your online game at no cost. Not only do we offer the tools to build your online game with all the SDKs and documentation necessary, we also provide a unified web interface, a Game Manager, that gives your operations team a single destination for viewing and managing a live game. Operating online games at scale is a challenge and we give you the best elastic backend available for any unexpected surge of traffic.

So how do you get started? We have made everything from making API calls to our backend service, to getting setup in your favorite engine as simple as possible. For example, getting setup in the Unity 3D Game engine could not be easier.

Adding the PlayFab Unity SDK to your game takes only a few steps:

1. Navigate to the PlayFab Unity web page
2. Click Download Unity Package
3. Open your Unity project
4. Click Assets, Import Package, Custom Package
5. Navigate to where you downloaded »PlayFabClientSDK.unitypackage« and select it
6. Click Open
7. In Importing package, select All and click Import





»SteamBirds Alliance« (left) as well as »AdVenture Capitalist« (right) are two games that use PlayFab for example.

That is all that is to it, you are ready to get started making your first API call. Take a moment and complete the rest of the tutorial here: http://api.playfab.com/docs/getting-started-with-playfab

Build

Economy

Making a beautiful game with awesome content is one thing, but managing an entire economy with its own marketplace is another. With items you can rent, trade, consume, stack, unlock, bundle in a group, or use into a »gacha« system; we offer various ways to edit and manage all of their properties and attributes from the game dashboard and APIs. From the fire rate of a weapon or the regeneration rate of a character's mana, your game designers can decide to nerf, add more items or simply change the pricing or description of any virtual good with no further development hours needed.

Let me show you how a simple configuration can highly benefit your game. Let's say that you have a weapon and it has a value for how much damage that weapon does to an enemy. Hard coding that value in your game client is probably the norm. With PlayFab, you can store those values in custom attributes on the catalog item that represents your weapon. Now we can use an API call to gather that weapon's damage value, thus eliminating the need to ever rebuild your client code.

Here is an example of pulling a value from a catalog item via our Unity SDK.

```
PlayFabClientAPI.GetCatalogItems(new GetCatalogItemsRequest() {
    CatalogVersion = "MyCatalog"
}, (result) => {

    var BigGun = result.Catalog.Find(ci => ci.ItemId == "BigGun");
    var attributes = JsonUtiltiy.FromJson<Dictionary<string, string>>(BigGun.CustomData);
    var weaponDamage = attributes["WeaponDamage"];

}, (error) => { });
```

This type of backend configuration programming puts more creative power in the hands of your game designers and will also boost your monetization by offering new possibilities.

You can also easily create promotions and offer different prices based on player achievements (or any other custom variables) by managing separate stores from the dashboard with their own unique pricing. This is the kind of creative freedom that helps monetization specialists to optimize their strategy and make you more money.

Meanwhile, we protect that money by eliminating fraudulent replay attacks via receipt validation. We know you wouldn't trust just anyone with your real money transactions, and neither would we, so security is a top priority at PlayFab. You can even let your players manage the game economy by giving them the power to trade via our trading API, since our escrow system guarantees that each transaction between players is legit.

And of course, trading wouldn't be complete without gifting. The social power of giving to others can tremendously improve your retention by fostering relationships and deepening emotional connection to, and within, the game.

Social

Successful games understand the importance of having a vibrant community and we offer a series of solutions to help your players bond.

Leaderboards are a great way to encourage interactions by providing a platform for friendly competition. You can even create push notifications to let high scorers know their top score are in jeopardy or has been beat, and by whom.

Photon, which provides multiplayer real time services, is already pre-integrated in the PlayFab platform. With Photon Chat, players can talk to each other in channels, and add friends in a list for matchmaking or to participate in a common quest. We all know the importance of grouping together players with commonalities such as language, culture, or player advancements to improve social connection and retention.

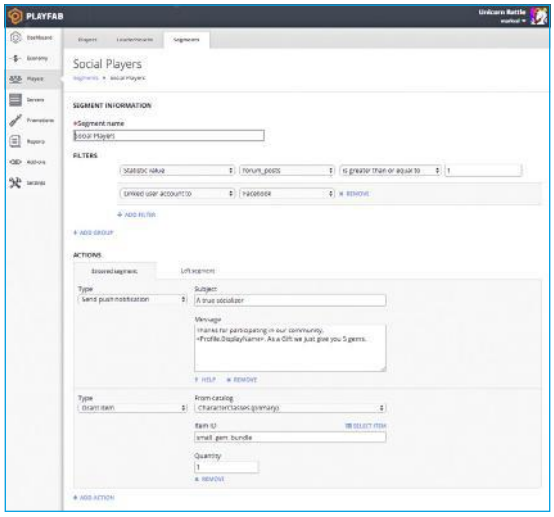
By using the Photon Cloud with PlayFab, developers can effectively power a wide range of game styles requiring server authority, without having to develop a full game server (not to mention pay for hosting). This means that even without a custom game server, titles

Ryan Williams from Spry Fox about PlayFab

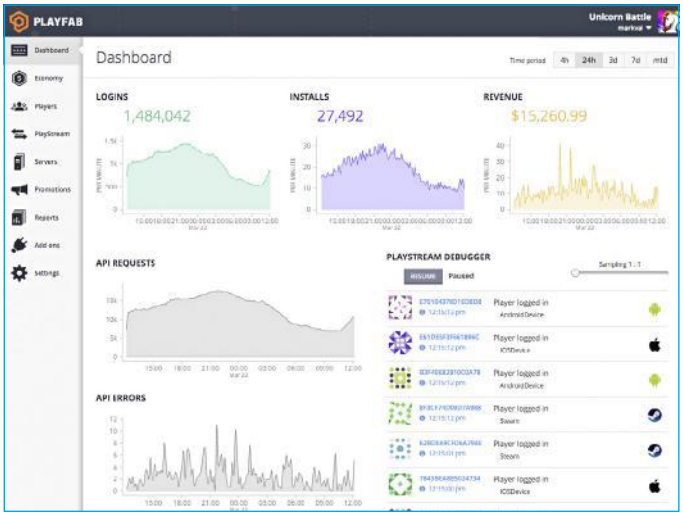
At first we wanted to implemented our own player data saving system but then we found out about PlayFab. Having an API that was doing exactly what we wanted, we could skip setting up server phase and service building. Now we are also using it for in-game chat and other major parts of our game. We chose PlayFab because we trust the team and the experience they have from building live game operations games. On top of that, the customer relations is really good by being extremely responsive to all our questions. Most of the developers forgot about the user support and management features, once they release a game and the PlayFab dashboard solves all of that.



PlayFab supports various engines and coding systems out of which you can choose.



Talking to your players directly helps to engage them even more with the game – therefore you can setup Push Notifications for all sorts of events.



The dashboard gives you a lot of information about your players and your game, most important revenue as well as monthly and daily player numbers.

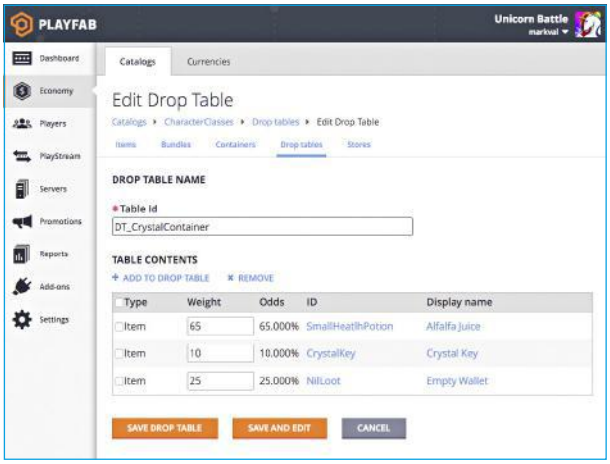
can do cheat detection, perform validation on scores for reporting statistics, retrieve complex datasets in a single call (rather than having to query for components individually), and many other operations which would be risky or inefficient if made from the client.

Here are the standard actions in Photon and the associated event handlers they call in PlayFab's Cloud Script:

Photon Actions	PlayFab Cloud Script Handlers
Create a Room	**handler.RoomCreated**
Join an existing Room	**handler.RoomJoined**
Change a Property on a Room	**handler.RoomPropertyUpdated**
Leave a Room	**handler.RoomLeft**
Close a Room	**handler.RoomClosed**

In addition, you can trigger a custom event on the client to drive any other logic you need via »OpRaiseEvent« in the »LoadBalancingClient«. You'll see examples of each of these in the PlayFab Photon Cloud Script sample.

One of the most essential and versatile game operations tools is the ability to speak directly to all of your players at the same time. You can use our news content generation panel to announce a new update, promote a specific feature, or to gather feedback. Being in communication with your players is a great way to get valuable data as well as increase player satisfaction and loyalty.



Via PlayFab, you can even tweak balancing and items in your game.

Furthermore, you can encourage conversion via push notifications. For example, you can let a mobile player know that a friend just joined a game, or that a special event is coming up. Customizing the Push Notifications per player will also develop a sense of ownership and personalization. The PlayFab dashboard gives you access to create and edit those messages to optimize messaging for each user persona.

Here is a simple example of getting a promotion from Title News:

```
PlayFabClientAPI.GetTitleNews(new GetTitleNewsRequest() {
    Count = 1
}, (result) =>
{
    var promoHeader = result.News[0].Title;
    var promoText = result.News[0].Body;
}, (error) => { });
```

Content and Logic

From the text used in your onboarding tutorial, to the power values of a monster or the icons used in a menu, there is so much to optimize. You don't want to be limited by hardcoding every essential piece of your game and having to work around developer release cycles. You'll want to create custom content for each culture, language, and persona, and you'll want to make changes on demand. Too often a typo slips into a release and you have to live with it. Being able to add or edit content gives you ownership and the ability to improve the overall game experience at any time. Our TitleData dashboard combined with our Cloud Script puts you in the driver's seat, now and always.

Having the capability to extend PlayFab APIs with fully customizable V8 JavaScript inside of Cloud Script means you can create functions that are truly unique to your game without worrying over infrastructure. From rewards systems, player-to-player interactions, on-demand notifications based on parameters, clan creation, or asynchronous matchmaking, the possibilities of Cloud Script are endless. You can also guarantee that there is no cheating happening on the client side by validating player moves based on progression or current internal data. See more about how to use Cloud Script here: https://api.playfab.com/docs/using-cloud-script/

Launch

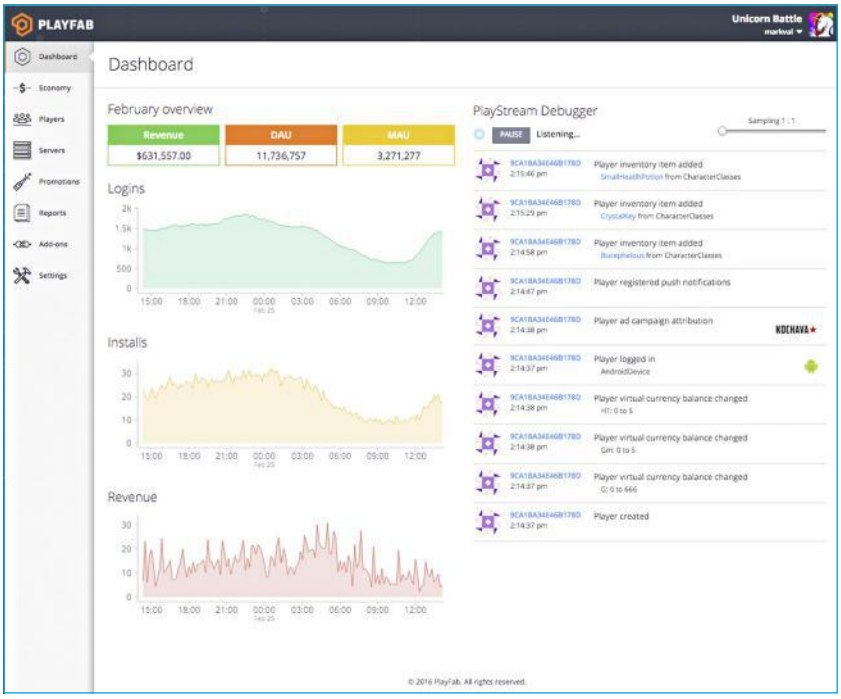
Before any launch comes the soft launch. This is the time to validate your game mechanics for retention, conversion-to-payers rate, and overall active users k-factor. Using our standard report, you can already validate your overall success, but you'll need some custom analytics to improve your onboarding tutorial or see current player progression. Happily, the seamless integration of 3rd party services with PlayFab gives you the ability to create funnels and deep segmentation that will provide the answers you need to hit your goals. Once you have proven your models, it's time for launch. Sit back and watch the magic of PlayFab's elastic scalability in action.

Cross Platform

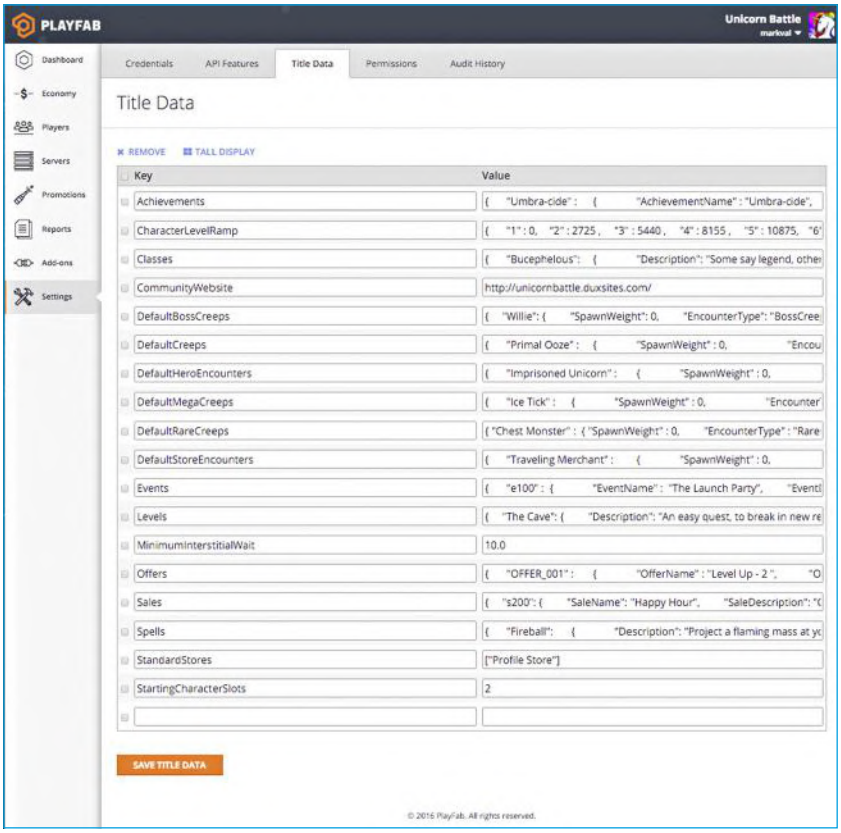
While your players might come from PSN/XBOX, Steam/Kongregate/Facebook and iOS/Android/Amazon or others, you'll have direct access to the heart of PlayFab; the players account manager. Accounts can be authenticated and linked from all third party services, which means PlayFab supports full cross-platform game play. The player dashboards gives you a suite of customer support tools to edit the data of any player. Inventory, leaderboards, emails, username, internal data or history of transactions ... it's all there for you to use to your best advantage.

Coupons

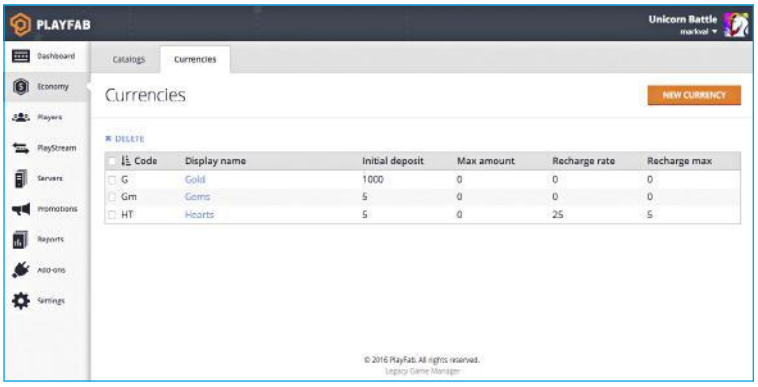
Another great tool in your PlayFab arsenal is promotional coupons. Everyone loves getting coupon codes; it feels like you're part of a privileged group. Coupons can be used to give free items or currency, but you could also make custom coupons that allow players to jump ahead in the game. This provides a great opportunity for players, and particularly YouTube reviewers, to get a sneak preview of what is happening further in the game without a big time commitment up front.

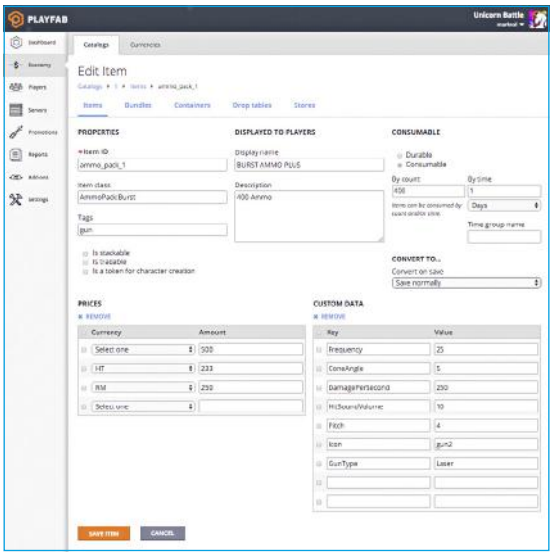


PlayFab helps you to easily keep track of your active users, revenue and so on.



With PlayFab you can edit a lot of features in your game like Achievements, Levels, Events and many more very easily.





Adding a new currency or a new item into your game can very quickly be done with the PlayFab backend.

User Rights Managements

If you have a large team, you may want to customize the dashboard to provide individual and team access to relevant material only. For example, you might only want to allow your publisher to view reports. With PlayFab User Permissions you can assign unique right of access, and see the full history of actions performed by the agents.

Grow

Your game is fun, your players are coming back, and you see the potential of your community blossoming. It's time to orchestrate growth by creating promotions, adding new content, and targeting the global hotspots from whence your next players will come.

Segment In-Game Behavior

The beauty of PlayFab comes when it is time to automate player actions by segment groups based on their in-game behavior. Being able to automatically send push notifications or show a different store to the players who are doing a lot of purchases upon entry or exit of a specific segment will boost monetization. For example, you might want to give a special item for everyone who logs in on Christmas, or send a congratulatory message to players who've hit









Some more games that are using PlayFab as their backend.

Hyper Hippo about PlayFab

If you are willing to make your own backend, you should take into consideration all the time spend working on that service could be spend in making your game a lot better. PlayFab allows you to have all the tools that you need right out of the box on day one. You just have to look through the documentation, download the libraries that you need and if it is your first project, within an hour or two you can be up and running. It gets you past all the hurdles and gets you to focus on making a fun game.

Using PlayFab is like having another person on the team to make your backend. Gaming companies are good at making games not services. Any time spent making services, is time not spent in making games fun and great.

Make the game fun and enjoyable first, then when it is time to go live and expand, PlayFab is there for it.

PlayFab can support AAA games while keeping the iterative approach of indie companies in helping them to get to an MVP.

I would recommend PlayFab because as a Designer you always require a wide toolset. This single tool is changing the whole canvas.

We strongly believe in live operations, which means you can adapt your game as it is running. Each week, we gather feedback from our fans and we can change inputs right away or try a new experiment without having to ship the game every time.

We wanted to run live services events every week or month for Christmas or Halloween by changing the theme and content inside the game.



With PlayFab we were able to use Unity Asset Bundles and push them to the PlayFab Content Delivery Network, while saving player profiles, inventories for different stores in their cloud and using segments with messaging to reach out to different players based on how long they have been playing, how much money they spend or players that are still running free by showing different things in the game. Flipping a switch or changing a value with direct impact makes all the magic behind the scene. The time spent on planning updates is significantly shorter.

a certain level or logged in a certain number of times. The possibilities are endless, and they can all be customized to follow your chosen tracking variables.

Beyond your Game

Creating a solid community base starts with encouraging players to be competitive via leaderboard functionality, and interactive via chat features, but it doesn't end there. You can go further, creating direct marketing campaigns to tell them about upcoming events or sales, and encouraging them to alert their friends. You can further expand and vitalize your community by integrating the friend list from Facebook and Steam. Cross-promoting between games is as simple as flipping a switch inside the news dashboard. Optimizing your revenue by adding new items to your economy and adjusting the price by country helps you stay ahead in an increasingly global market.

Survival is not enough. We want you to shine, and good luck comes to the well-prepared. PlayFab's live game operations platform is what you need to optimize each and every game you release through improved iteration flexibility, speed, and control, as well as increased insight. And stay tuned! We are investing in some exciting new feature areas that will be rolled out in the coming weeks and months.

Add-ons Marketplace

We realize that no matter how big PlayFab gets, we will never be able to meet 100 percent of your backend service needs. There will always be other companies providing valuable service that your game needs – from in-game advertising, to analytics, to payment systems, and even to monetization consulting or customer service.

Until now, if you wanted to use one of these services, you needed to integrate it into your game yourself – typically by dropping in yet another SDK. These SDK's quickly add up, to the point where »SDK fatigue« seems to be a real thing.

Yet here's the thing – many of these services don't actually require a separate SDK. Many of them can be integrated server-to-server, wired up directly to your game backend. So that's exactly what we've done with this new add-on marketplace – make it easy for you to discover, signup, and start using these new services immediately, in many cases with just the push of a button. We've already done all the integration work necessary for it to start working with your game.

We're really excited by this Marketplace because we think it's a true win-win-win for everyone:

- You win by getting a curated list of the best services, available in most cases at the click-of-a-button – no engineer needed.
- The add-on providers win because it's much easier and faster for them to sign up new customers.
- We win by getting cool new features that makes PlayFab more valuable to you, as well as a new revenue stream that allows us to drop our price to free, making our platform even more attractive.

Mark Val

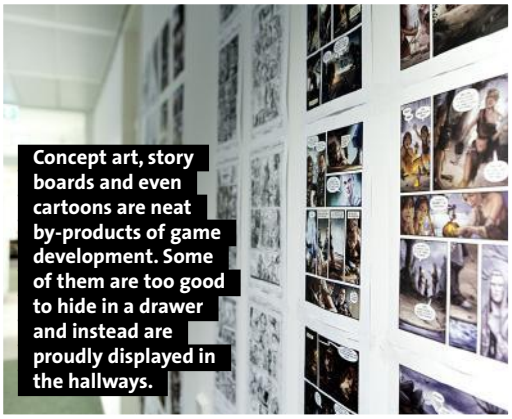


Via the add-on marketplace, you can easily integrate a lot of well known services into your game without an engineer needed.

A day in the life of ... CRYTEK



Over the years, Crytek has received several awards for their games, e. g. for »Crysis 3« and its protagonist Prophet (left).



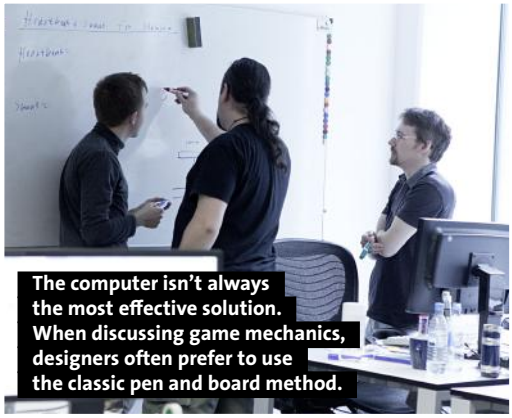
Concept art, story boards and even cartoons are neat by-products of game development. Some of them are too good to hide in a drawer and instead are proudly displayed in the hallways.



Audio designers need their own office, so they can mute any background noises and don't annoy others when they sometimes need to turn up the volume.



The philosophy from the first part of the »Crysis« series is part of the tough daily office routine.



The computer isn't always the most effective solution. When discussing game mechanics, designers often prefer to use the classic pen and board method.



2016 is an important year for Virtual Reality. With their own CRYENGINE and a game for both Oculus Rift and PlayStation VR, Crytek wants to secure its position in this new area.



When a game is released, every team member gets a framed disk (old school!) with the signatures of all co-workers who worked on the game.



The movie theater can be used to test in-house projects on the big screen with full surround sound or simply to play a game during lunch break.



Stand-up meetings are important in order to keep everyone on the team up-to-date. Sometimes more than 100 people are working on a project.

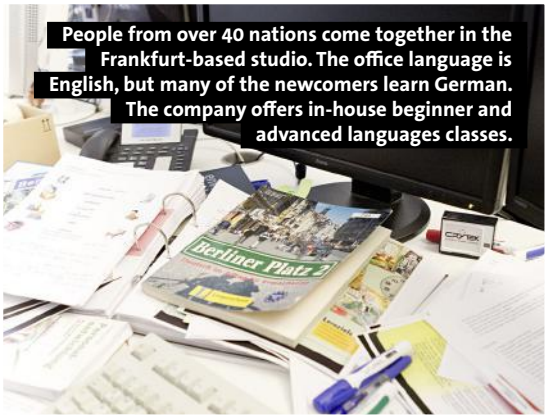


Fans can follow Crytek streams on Twitch.tv, which are mainly produced in a special recording facility of their Frankfurt studio.

There's no question: Crytek is one of the most well-known German developers, having already made their first appearance in the AAA league with »Far Cry« back in 2004, while today they focus on VR. In Frankfurt, where Crytek chose to set up their headquarters, they opened their doors for us.



Not all design concepts make it into the game.



People from over 40 nations come together in the Frankfurt-based studio. The office language is English, but many of the newcomers learn German. The company offers in-house beginner and advanced languages classes.



Crytek was founded in 1999 in Coburg, Germany, and attracted international attention with titles like »Far Cry«, the »Crysis« series and »Ryse: Son of Rome«. In 2007, the developer moved its headquarters to Frankfurt am Main.



Marius, the protagonist from »Ryse: Son of Rome« published in 2013, has by now swapped his Roman helmet for a sombrero.



Crytek makes no secret of where the game developer and publisher has its offices.

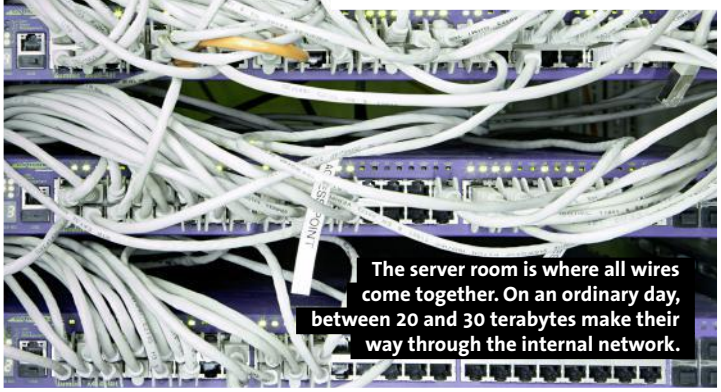


In addition to their German headquarters, Crytek owns studios in Bulgaria, Hungary, Ukraine, Turkey, China and Korea.



The technology demos »Back To Dinosaur Island 1&2« will soon be followed by the two full-fledged Virtual Reality experiences »The Climb« und »Robinson: The Journey«.





The server room is where all wires come together. On an ordinary day, between 20 and 30 terabytes make their way through the internal network.



Motion capturing allows for particularly realistic animations by recording real motion sequences. To do so, actors need to wear a skin-tight suit whichcan be captured precisely by special cameras.

TOOLS

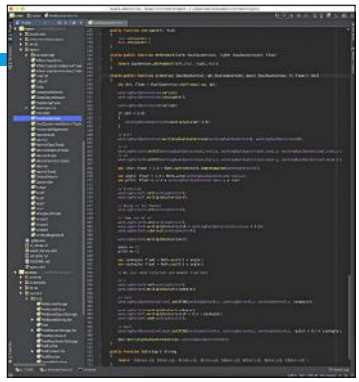
You need great tools in order to build great games. Experts from the industry share their view on recent updates as well as new software solutions and reveal their favorite little helpers.

HAXE WWW.HAXE.ORG



Sven Otto
Lead Software Developer
at GameDuell

Anyone who develops games for several platforms will love Haxe. Haxe is an open-source toolkit based on a modern, high-level programming language. It contains a complete standard cross-platform library in order to address a number of target platforms, from mobile platforms like iOS and Android to HTML 5, Neko, Node JS, Flash, C++, Python, C# to PHP and Java. Our team at GameDuell has written its own game engine based on Haxe and made it available on Github including its libraries (https://github.com/gameduell). In late February, we published »Grand Gin Rummy«, our first globally available mobile game for iOS and Android, which is based on Haxe. The HTML5 version for Facebook is nearly finished, too, and will follow shortly. Publishing additional versions in the future like for Apple TV or Flash would be a piece of cake thanks to Haxe.



The toolkit Haxe makes programming for different target platforms easy.

PIXEXIX PIXEXIX.SOPHIEHOULDEN.COM



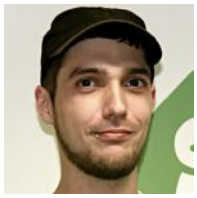
Pixexix allows you to edit textures, for example in Photoshop, and view the results live on the 3D object.

Pixexix allows artists to texture 3D models by dynamically mapping images onto the model. The models just have to be available as OBJs and include UV coordinates. You select a perspective and click to make a snapshot; a PSD file which can then be edited directly in Photoshop or any other graphic program. Whatever is drawn there is projected directly onto a 3D model; you may repeat this step as often as you like and draw an entire model this way. Pixexix is being developed by indie game programmer Sophie Houlden and is currently in Alpha phase. The only thing missing is an option to draw onto more than one texture at a time. A limited version can be downloaded free of charge; the full version is available for 30 dollars.



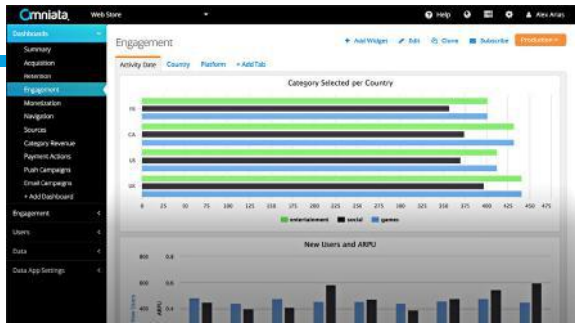
Friedrich Hanisch
Programmer and
Game Designer
at Rat King

OMNIATA WWW.OMNIATA.COM



Markus Nitter
Producer
at Sproing

Omniata is a data platform used by a number of top publishers in the mobile sector. The benefit of Omniata is the enormous variety of supported features. It even covers some parts of marketing like automatically sending push notifications based on user behavior. Especially the possibility to carry out A/B tests or cohort analyses makes Omniata a valuable tool in the world of virtual goods. The comfortable user interface even allows beginners to create complex evaluations. Graphics and charts are refreshed in real-time, which allows for quick and easy reviews of query results. The integration in one's own app is easy thanks to the good documentation. Standard KPIs are also pre-defined and explained in detail. All this makes Omniata the perfect solution for everyone with a centralized way to work and an aversion to dealing with tons of tools.



Omniata is an easy-to-use data and analysis platform equipped with comprehensive features.

MAKING GAMES
THE NEXT ISSUE

04|2016 MAKINGGAMES WILL BE AVAILABLE ON JUNE 17TH, 2016

Culture Shock

This time around we will keep the promise: Steam, various App Stores as well as PlayStation Network and Xbox Live are easy toolsets for developers to publish their games worldwide, opening up markets like China, Russia or Brasil for example. Those were not reachable without distribution or publishing partners in the old days, but this new freedom brings along a lot of problems. Our experts will give advice on which countries prefer what kind of genre or which platform to play on, they will explain how games have to be adapted for different territories or which legal restrictions there might be as well as how to find the right partner in an unknown culture.

Other topics

A critical look at the MDA-framework

Wolfgang Walk has analyzed the MDA game design framework and explains why this approach is outdated nowadays.

Best Practice Brand Management

Blue Byte will give advice how developers can establish a brand and why this might be important for a game to be successful.

Interview with Ron Gilbert

The creator of »Monkey Island« and »Maniac Mansion« talks about the work on classical adventure games and challenges on Kickstarter.

IMPRINT

Webedia Gaming GmbH
Redaktion Making Games
Ridlerstraße 55
80339 München
Tel.: 089 / 244 136-0, Fax: 089 / 244 136-118
General managers: Nicolas John
Editor-in-chief: Heiko Klinge (v. i.S. d. § 55 Abs. 2 RStV) - see address of the publishing house
GameStar GmbH is a 100% subsidiary of
Webedia Gaming GmbH
Ridlerstraße 55, 80339 München
Tel.: +49 (89) 244 136-600, Fax: -89600
General Manager: Nicolas John
Register court: District court of Munich
Commercial Register Number: HR B 218 859
USt-ID Nr.: DE 300 33 09 61
Shareholder of Webedia Gaming GmbH are the majority shareholders Webedia GmbH (District court of Charlottenburg, HR B 114531 B), as well as PietSmiet Holding GmbH (District court of Cologne, HR B 84294) and Connected 4 Media GmbH (District court of Munich, HR B 218234)

DISTRIBUTION TRADE CIRCULATION

MZV GmbH & Co. KG
Ohmstraße 1, 85716 Unterschleißheim
Telefon: 089 / 319 06-0, Fax: -113
E-Mail: info@mzv.de
Web: www.mzv.de

EDITORIAL TEAM

Chefredakteur
Heiko Klinge

Managing Editor
Sebastian Weber

Editors
Yassin Chakhchoukh
Markus Schwerdtel

Design
Sigrun Rüb (ltd.)
Manfred Aumaier
Alexander Wagner
Eva Zechmeister
Anita Blockinger (Freelancer)

Editing (Freelancer)
Marion Schneider

Translation
Bettina Wilding

Contributors:
Friedrich Hanisch, Travis Hoffstetter, Miguel Angel Horna, Ricardas Jascemskas, Caroline Martin, Markus Nitter, Sven Otto, Patrick Plourde, Boian Spasov, Mark Val, Timm Walter

We thank our interview partners:
Olivier Dauba, Chadi Lebbos, David Lightbown, Olivier Palmieri, Alex Parizeau, Gabriel Robert, Jessica Tams, Julien Varnier, Sadia Rahman Zain

Office Management:
Isa Stamp, Anita Thiel, Annie Weissenberger

CONTACT US

Advertising Sales:
Ralf Sattelberger 089 / 244 136-730
ralf.sattelberger@webedia-group.com
Director Sales:
Ralf Sattelberger 089 / 244 136-730
Article Suggestions and Comments:
editor@makinggames.biz
Recruiting & Projects:
projects@makinggames.biz
Address changes and Subscriptions:
kundenservice@makinggames.de
Changes company register:
info@makinggames.biz
Betreff: Company register
Ad approval for print issue:
Manfred Aumaier, 089 / 244 136-668
anzeigendispoprint@gamestar.de
Ad approval Online:
werbemittelonline@idg-eas.de
Media rates:
www.makinggames.biz/mediadaten
Printing:
SDV Direct World GmbH
01159 Dresden

CUSTOMER SUPPORT

Subscriptions and single issue orders:
Making Games Kundenservice
ZENIT Pressevertrieb GmbH
Postfach 81 05 80, 70522 Stuttgart
Tel.: +49 (0) 711 / 7252-275
Fax: +49 (0) 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 per year (includes 6 issues)
Please contact us for foreign prices and bulk discounts
Payment:
Postbank Stuttgart, BLZ 600 100 70
Konto-Nr.: 311 704
How to reach the team of Making Games:
Webedia Gaming GmbH
Redaktion Making Games
Ridlerstraße 55
80339 München
Tel.: 089 / 244 136-0, Fax: 089 / 244 136-118

LIABILITY INFORMATION

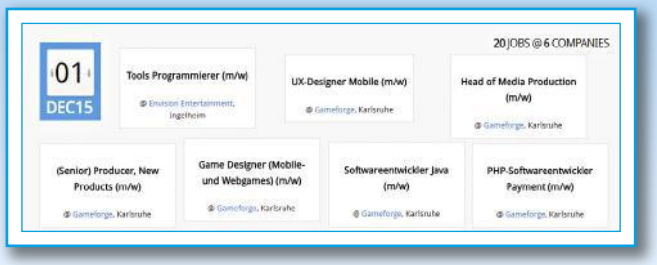
© Copyright Webedia Gaming GmbH

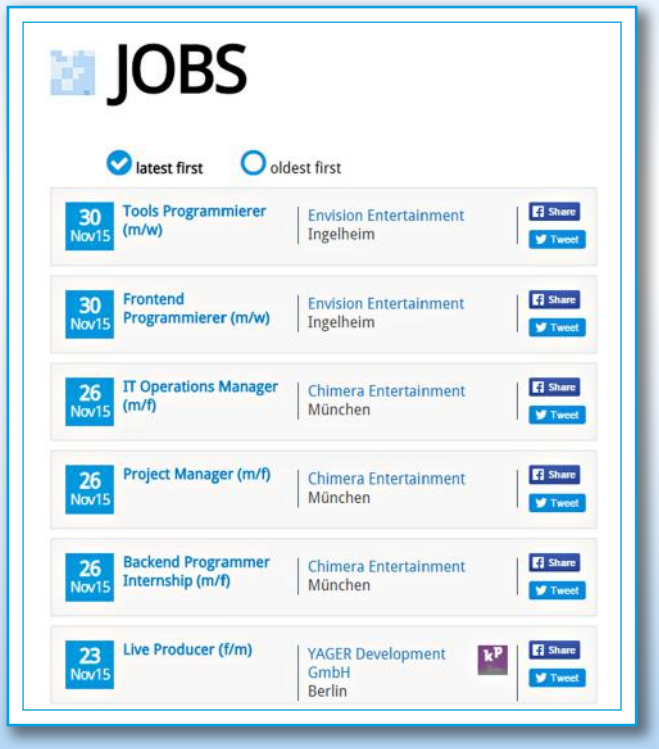
Liability: Despite review neither the editorial team nor the publisher are liable for the validity of any publication. Any paper in Making Games is published without consideration of possible patent protections. Any names of products are used without warranty of free utilization.

making games JOB EXCHANGE

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

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







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



Or get in contact:
Sebastian Weber
Phone: +49 (0)89 / 244 136 650
E-Mail: sebastian.weber@webedia-group.com


GAMEFORGE
SPIELEND KARRIERE MACHEN!
UNSER STANDORT: KARLSRUHE
UNSER SPIELFELD: DIE WELT
WIR MACHEN ERSTKLASSIGE ONLINE-SPIELE.
JOB-LEVEL-UP GEFÄLLIG?
DANN KOMM ZU GAMEFORGE!
http://corporate.gameforge.com/jobs/
Adresse:
Recruiting Office
Albert-Nestler-Str. 8
76131 Karlsruhe
Tel: +49 721 354 808 244