Postmortem: Vivid Games' Real Boxing

PRINT

By Tomasz Strzelczyk,Grzegorz Brol,Krystian Komisarek

Here at Vivid Games, we've been making mobile games for nearly 10 years. We've produced several sports titles around ski jumping, speedway, and BMX. We've even tried our hand with the future of sport, by bringing the classic *Speedball 2: Evolution* to both iOS and Android.

Whilst deciding on our next game, we saw a great opportunity to bring a realistic boxing title to mobile and tablet. With EA's *Fight Night Champion* being the only real contender for the boxing title, we thought that we would enter the ring as the wildcard outsider.

We've noticed that in the last year, something has fundamentally changed on mobile and the App Store. The bar has been raised massively in terms of gameplay and production values, with standout titles such as *Infinity Blade II*, *Dead Space*, *CSR Racing*, *Sky Gamblers*, *Dead Trigger*, and *Bastion*.

The increasingly competitive world of the App Store means that you've got to invest heavily and have triple-A titles that will grab the attention of the media, and, most of all, Apple and Google -- with all developers hoping to be featured each Thursday.

That's why we made a point of never compromising our vision of the game, *Vivid Boxing*, and we invested a huge amount of resources into the project, using the best technology available. The cornerstone of this was Epic's Unreal Engine 3, to bring a console look and feel to the game. We risked a lot and crossed a lot of barriers, but our initial vision remained unchanged.

What Went Right

1. The Team

People are definitely Vivid's most important and valuable asset. It's something you have to keep in mind when organizing their work. Whilst working on *Real Boxing*, rooms and furniture were rearranged so that people wouldn't have to run across the building searching for one another. Team meetings were being held so often that we had to use our social room, because the only conference room we had was not enough.

Researching boxing through movies and games was part of getting ready for *Real Boxing*. We also had gym sessions, martial arts training, and a punching bag hanging in the heart of the office -- all that constantly reminding us how important this project was for the company.

One of the key decisions was to bring people from different teams together to work on the game. Developers, designers, and everyone working on some of the other projects ended up helping with the production of *Real Boxing* in the last weeks before release. We're extremely proud of everyone's involvement, which extended far beyond what was expected of them. For example, game testers not only reported on errors, but also suggested how to fix them. Their knowledge of boxing also proved very resourceful. Most importantly, everyone was highly motivated throughout

the entire production.

2. The Fighting System

The fighting system was always going to be our greatest challenge. When production started, we tested different types of touchscreen control. The first version of the control system used two buttons to dodge left and right. Tapping and holding with both fingers led to a blocking movement. Making single-tap or slide gestures performed different types of punches.

However, the first focus test clearly proved that most people were unaware that boxing involves a lot of dodging, so we decided to implement a single dodge button which would trigger the appropriate type of dodge depending on the attacks received. This way, the player only had to pick the right moment to dodge.

Simplifying the gameplay mechanics went even further. Someone suggested dividing the screen into two areas: the left side being defensive -- where taps would trigger a dodge, whilst tap-and-hold would start blocking, and the right one being offensive -- where taps and slides would trigger punches.

We soon discovered that this required too much interaction and killed all the fun. Players couldn't decide which hand to use for the next attack and it wasn't natural. There was even a moment when we were thinking about dividing the screen into four areas so that the player could decide between throwing punches at the head or at the opponent's body but we abandoned this idea since it overly complicated the gameplay.

Eventually, we returned to the previous idea, which was pretty similar to what we have in the final build with the control system deciding if it's better to hit the head or the torso. This gave the player time to attack properly with the addition two buttons on the HUD for blocking and dodging.

We then had to make a difficult decision about the flow of the fight. We needed something that would make the fight less chaotic -- less like a street brawl and more like actual boxing. Implementing smart AI was an important part of the process. The effectiveness of each blow was limited by the amount of stamina each character had. Each punch lead to a decrease in stamina, so the AI should not only decide when to attack the player, but also when to slow down for a bit to regain stamina. This is how "floating turns" were born.

The duration of each turn depends on the actions of both fighters: if the player's fighter has a full stamina meter, it means that the player may attack soon, so the AI becomes defensive, and uses blocks and dodges; if the player

decides not to attack, despite the high stamina, the AI will attack -- if his stamina meter is full.

In addition, each turn can interrupted with a dodge. A successful dodge grants the player a stamina boost and a great opportunity for the counterattack, which then causes greater damage than a regular punch. Transitions between offensive and defensive "floating turns" were marked with text displayed on the screen: ATTACK and BLOCK or DODGE.

We found that sticking with this “natural” flow of the fight was the best way to beat down the opponent. Just like in the real ring, it's wise not to rush at your opponent from the very start. A wild and furious charge drains all the stamina and your moves become less effective --offensively and defensively.

Finally, to make the player feel truly like they're in a boxing ring, you also need to have a camera that works in the right way. We wanted one that would shake in a realistic way during attacks and whilst taking damage, and our first attempt involved point interpolation and the built-in shake mechanics of Unreal Engine 3. This turned out to be more chaotic than we anticipated, and we decided to go with a camera sporting smoother transitions. We placed the camera at a fixed point in relation to the boxer, and then simulated movement using particle physics affected by several forces.

For example, one of these would be the force stemming from animations, an elastic force turning the camera back to its default position in relation to the boxers, or the resistance force, which prevents the camera from moving at excessive speeds or shaking too much. Assigning a single three-dimensional vector to each animation allowed us to create a fully animated gameplay camera, with each attack making it respond in a unique and satisfying way every time.

---

3. Unique Features: Gesture Control System and Multiplayer Mode

The idea for the gesture control system was put forward by our CEO Remigiusz Koscielny at a meeting with Apple. We put forward the idea of using the camera on Apple devices to enable the player to control their fighter in a Kinect-like style. This immediately sparked Apple's interest, but the challenge was that we hadn't actually developed the system and no other iOS application featured anything like this, so we so then had to figure out how to make it work!

The development of our innovative motion control scheme (later dubbed V-Motion) consisted of three phases. In the first we acquired the necessary data, which was then processed in phase two, the longest one. This processed data became the base of our technology for the Gesture Control System.

In the third phase, movements were transferred into the game engine. The initial results weren't very good. The system was unstable and needed to be continuously re-tested. However, three weeks before the project's deadline, we finally had a breakthrough which meat that it worked. We were all surprised and excited by how the motion controls turned out. People at Apple, who were given an early build of the system, were also very impressed.

At the time, we noticed that each player moves differently, and that the system didn't always register the gestures correctly. It was therefore necessary to design a tutorial for the player to learn how to move, so that their boxer behaved how they wanted him to. The work on the gesture control system lasted three months. Admittedly, the motion controls still had some limitations, working best in a brightly lit environment with the player dressed in a contrasting color over a plain background such as a wall.

True multiplayer rarely appears in iOS games. When it does, it is usually in a turn-based form. Our goal was to create a fully-fledged, real time multiplayer experience for two players, but first we needed to see how the Unreal Engine worked with iOS and the Game Center system.

Most of the problems we encountered whilst developing *Real Boxing*'s multiplayer were due to the need to optimize the performance across different devices, because one player may be running the game on an iPhone 4S and the

other on the new iPad. Making it work meant facing many challenges.

The multiplayer mode was created by Jaroslaw Dziuk and Lukasz Purcelewski, and the whole process took more than three months in total. We're are still working on further optimization for the online experience, with future updates expanding it with new social networking functionalities and even game modes created by Bartosz Biniecki -- one of our game designers.

4. Outsourcing

Both the scale and degree of sophistication of the project were enormous, which is why we decided to outsource some of the work to other companies. The most important thing for us in any collaboration is receiving a finished product, ready to be implemented into the game itself.

Dash Dot Creations, one of the best animation studios in Poland, did the animations for Real Boxing. This encompassed the teaser trailers, game trailers, game intro, animations, and computer processing.

In order to achieve a high degree of realism, we decided to use motion capture sessions with real boxers. This was undertaken in the state of the art Alvernia Studios, which has been used for *The Witcher 2: Assassins of Kings* and *Bulletstorm*. Voice acting for the game was recorded in London, at the famous Studio OM (*Driver: San Francisco*, *Risen 2: Dark Waters*, *Tales of Monkey Island*). Working with these companies allowed us to maintain high quality standards and also to finish the game in a very short time.

5. Visuals

Using the Unreal Engine has brought a lot of praise for the game's visuals. It's one of our best achievements, and it shows just how far one can push current mobile devices. Just some of the graphical effects we implemented are depth of field, color grading, bloom, rim lighting, normal mapping, and environment mapping.

The boxers are the most important element of our game and that is why they received the most polish. We made highly customizable characters with players being able to change not only their name and hair color, but also skin color, tattoos, and clothes. We want to highlight this because implementing tattoos, as simple as it sounds, required us to modify the game engine. What's more, during the fight every punch has a visible effect on the boxer: face and body deformation, facial expressions, blood, bruises -- we wanted to create boxers as close to the reality as possible.

Real Boxing is a game for iOS, but Wojciech Michalski, created 3D models that could be used in games for PlayStation 3 and Xbox 360. Each one consists of 10 million polygons, which wasn't an easy thing to do, considering some of the limitations of the iOS version of Unreal Engine -- poor lighting being one. In order to obtain good results, we had to resort to a number of tricks (for example, the light source is positioned at the point of the camera, which gave good shading and good effects on the skin of boxers) as well as work around compression issues, because you'll not find clearly pixelated textures like in many other games on iOS, yet our boxers show great detail right down to their muscles.

What we managed to do with the audience, however, is on a different level; it was our mini Mission Impossible. One of our graphic artists, Bartosz Rydel, created the most advanced animated audience on mobile devices and based the models on our own Vivid Games' people. Sadly, everything comes with a price. Performance optimization was very important; hence the iPhone 4 version of *Real Boxing* doesn't have these animated character models, but a texture with the audience instead.

The 3D menu is another jewel. The background is an interactive training arena, from which you can go straight to mini games (mini-bag, speed-bag, jumping rope). Creating games for mobile devices is in many ways much more difficult than a PC. On a smartphone you can't just add motion blur with a press of a button, but you still need to implement it. That is why we are very happy *Real Boxing* was appreciated in terms of graphics.

What Went Wrong

1. Training Mode

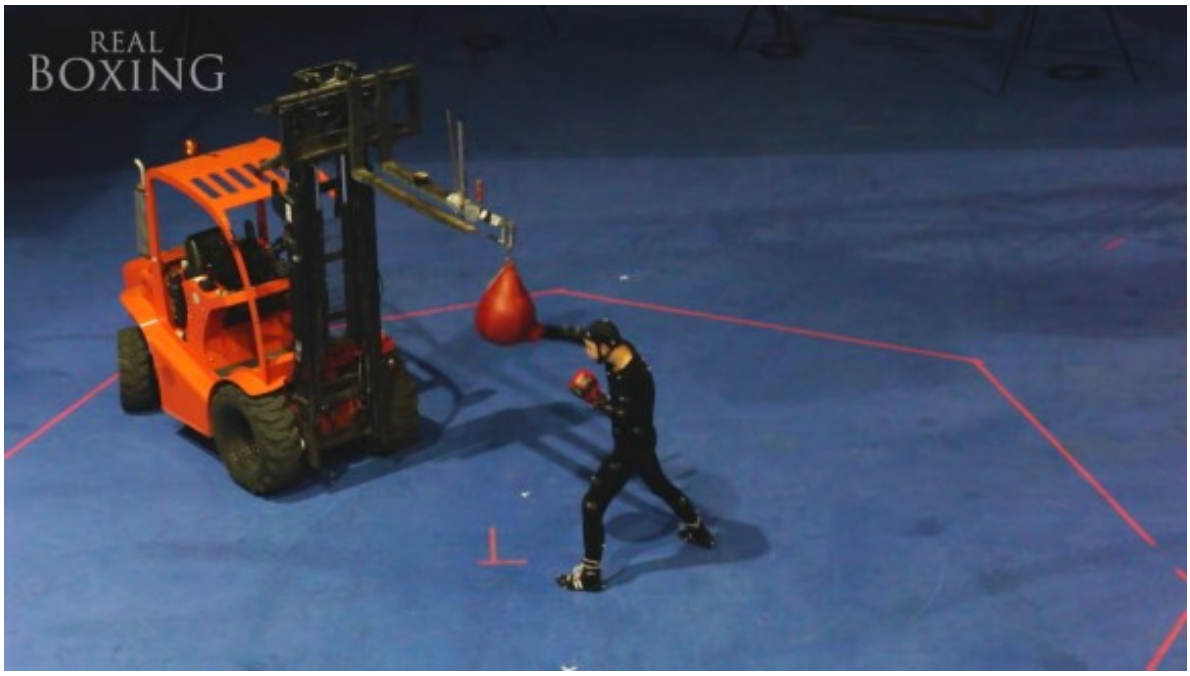
The challenge for any game is to launch as bug-free as possible. This is especially crucial on the App Store, where a minor error can result in poor reviews and a lowered star rating. Testers reported 1403 errors -- including both the smallest ones that were corrected in five minutes as well as 193 fatal errors. The team worked almost nonstop, and the infamous record belongs to Jacek Szarejko, senior tester, who worked 46 hours in a row -- just like doctors in hospitals (he also recorded 3.13 GB of gameplay videos and photos).

Despite all of this, the first version of the game contained two critical errors, which meant that it failed to end if the player completed the final task ahead of time. The error slipped through testing because of some changes implemented in the version after the final tests, which theoretically weren't supposed to affect the training in any way. We immediately eliminated the problem in the first update, but for many players their first impression was associated with this error, resulting in a mix of one and five star reviews.

Early in the development, we were well protected from such errors. Once a week, we had a control of the files and code review. The document reported bugs and fixes, and attributed them to specific developers. Our goal was primarily to eliminate the problems of further development of the code. The advantage of this arrangement was that one person had an insight into the whole project.

We also had also several additional documents for our developers, thanks to which a new programmer could quickly get in the draft, for example one about the setting-up tools (configuration, commissioning of the project), technical docs (technical aspects of the implementation), or Unreal script best practices (code tips). Of course, the preparation of documents and full control required a lot of time. Krystian Komisarek, our lead programmer who dealt with the code reviews, earned the name of "doc writer."

However, in retrospect, we can still see some imperfections and shortcomings of our system, for example naming and sorting of reported comments in reviews which caused the use of the review to be very difficult.



2. iPod Touch (4th Generation)

Literally a week before the premiere, there were problems with support of both iPod Touch 4th Generation and iPhone 4, and we thought we had overcome them. After the release, though, it turned out that despite our efforts, there were still problems. We immediately informed players, adding a message to the App Store description of the game and apologizing to them. The error has been eliminated in another update, and players who purchased the game on these devices were rewarded with in-game coins.

We handled errors via FlySpray, whose advantage is the ability to add tags. It allows you to quickly search for specific errors, set the priority and specifying the percentage of its amendments. We added attachments (print screen, videos), which allowed us to fix or verify bugs quickly.

Testers also connected errors: if one error was due to another there was a chance that one patch will work for both. In addition, we always added the following comments to all errors: who and what improved, version, device. This allowed fast transmission of information to the right people. Unfortunately, even this system didn't save us from those two big mistakes. We're always striving to improve processes and learn from mistakes, that was a good lesson for us.

3. Documentation

The preproduction phase was rather short, and we didn't have much time to discuss some key game features. The main game design document and other additional documents (i.e. SFX and music list, voiceover document, game economy sheet) were finished as the project was moving on. Updating documents every week was a great way to monitor progress. For each documentation update, we sent a short outline of the changes to the team members with bullet points in the email giving them a glimpse of key changes in the project.

The information flow may fail a bit from time to time. We decided that the best way to improve the information flow was to organize meetings with the people involved in the implementation of specific features. Sometimes, spontaneous talks took place in the team's room saving us from lots of additional work or helped to solve the major issues.

---

4. Kismet and Animation

We had problems with Kismet, which was used by designers for the prototypes. Their job was well done, but a problem appeared in the final version of the game. We saw real monsters built from the blocks, and gargantuan links between independent elements of the game. It was a source of extreme errors.

We also had to deal with the framerate and drops in performance. For example, the reaction to the attack was associated with face and body deformation, facial expressions, blood or bruises. Problems ended after we transferred those events to the Unreal Script -- programmers rewrote what level designers had created in Kismet on the scripting language. It gave us better control and improved performance.

We also encountered serious problems with the animation. Perhaps we didn't spend enough time on research and training, because part of it was incorrect. Particularly troublesome was the distance between players. We recorded three different distances for each hit, so that devs could afford to choose the best one.

However, we still had the unusual situation when the boxers began to attack at the same time. We had to use a couple of tricks to overcome this including collision detection, inverse kinematics and selection of the most appropriate action.

5. Crunch Time

The decision to cooperate with external companies was correct, but we found out that reporting errors and bug fixing became much harder. Direct contact always shortens the path and speeds up the process. It was the right decision, but passing the data to the new programmers who later joined the production almost killed our server on the same day when we had to finish a valid build!

Like most developers, our team was working flat out at the end of the project to deliver on time, as Apple had indicated they would possibly get behind the game in a big way. During this time, one of the rooms literally turned into a repository of energy drinks and local restaurants provided meals on a daily basis for those working extra hours. The ambitious nature of the project put massive pressure on the team, and each developer was aware of the need for it.

There is no simple recipe for a solution to this problem. We can't pretend we found one either -- yet -- but we will change few things in the future. To name but the smallest one, for example, as we experienced occasional delays due to lack of complementary equipment -- graphic designers required additional monitors and had to wait for delivery, so we'll surely optimize the work stations earlier next time.

In the case of a tight schedule, every short downtime can cause people to stay longer. For our next high-budget title we will spend more time planning, researching and will invest even more resources into the project preparation. For *Real Boxing*, we knew that we were working on a triple-A title, but that's easier said than done.

It is certain, however, that every team needs to rest and regenerate after such a hard work, so everyone got additional paid vacation after the release of the game and began to receive overtime. We also prepared a huge “Thanks for the hard work” banner for the office.

Final Verdict

We feel that with *Real Boxing*, we achieved a victory, one that was confirmed by reviews and gamers alike. The investment in quality shone though, and we were awarded the coveted Editor's Choice by Apple, and received almost universally great reviews from all the major media that we targeted.

We also learnt a lot of lessons, as developing high-end games for mobile devices is new territory. *Real Boxing* has by far and away been the biggest venture for our studio and brought invaluable experience. But this is not the end. We plan on supporting *Real Boxing* by providing updates, releasing new content (arenas, boxers, items), and expanding the multiplayer mode. We're also looking at the possibility of releasing our game on other platforms. Of course, *Real Boxing* is just the first of a few high end titles we've got in the works, so watch this space!

Data Box

Developer: Vivid Games
Publisher: Vivid Games
Release Date: 15 November 2012
Platforms: iOS
Number of Developers: 25 full-time, 8 subcontractors
Length of Development: 5 months
Budget: $300,000
Development Tools: Unreal Engine, Scaleform
And: More than 500 energy drinks, few boxing training sessions, 25 pairs of boxing gloves, about 200 dinners, one punching bag, a few thousand coffees, one big Thank You banner for our team.

Return to the full version of this article