Making Games is Hard… No Seriously, You Would Be Surprised.

Game Design is one thing we learned as a company to be surprisingly complex. At a glance, being an avid gamer myself, one would think that is the only prerequisite besides having the aptitude to program the game, but game design is an art. It's an art that involves mixing all mediums that need to work in concert.

This became glaringly obvious on Red Piston's first large internal game, Banzai Blowfish, where we felt we learned enough on client projects to make an ambitiously polished game.

The concept for Banzai Blowfish was a simple one: make a game that sort of played like a mix of a platformer like Mario brothers but with a classic physics-based pinball game mechanism. The idea was that each level has a series of pinball like bumpers, louvers, and other mechanisms that can manipulate to get the hero Banzai Blowfish to ultimately break out his family members from a bamboo cage somewhere in the level.

We started with rough sketches, brainstormed, and refined until we had the hero and villains sketched out. We were ecstatic. It was coming together.

Soon we realized, playing other hit iPhone games, how far we needed to go. Even at that time which one would consider the heyday of mobile gaming, the hits were polished. They all had great music, sound effects, graphics, and most important, gameplay.

Even with our small studio size, we were confident we had the right team to create amazing art in its many forms.

Levels were sketched in rough form on blank 5x11 papers, approved, and then rendered in Photoshop. This method of making levels proved to be onerous and a real thorn in our sides as we forgot the most important aspect of game development, the gameplay.

Essentially, after a couple months of creating music, sound, and even visuals we were all happy about, we started to find the gameplay to be challenging to nail.

This is where we learned the hard way how deceptively difficult it can be to make a great game; sure, you can make a game with cool sound, graphics, and to look polished, but is it any fun?

We were fighting with the physics engine the whole time as levels that seemed to work on paper ended up impossible to beat when Banzai would unpredictably find himself wedged in the environment, sometimes literally between a rock and a hard place.

We also realized how hard it is to holistically create a fun experience; we had level progression that was all over the place. Early testers complained they can breeze through to level 6 which suddenly was impossible, but then the next 5 levels after were a breeze again.

These were teachable moments, and we worked hard to course-correct using level components where we could instead of re-doing levels from scratch. This made for a mobile game that was good but could have possibly been better.

Lesson two would be test early and often on all devices; we gasped in horror as on our lower bounds targeted devices, frame rates dropped to a totally unplayable 5 frames a second.

This left the team feeling deflated, and personally, myself on programming, literally feeling sick. I felt of course personally responsible for a product that was now un-shippable after months of everyone’s hard work. I had failed the team, failed to make a game that runs at all on our targeted devices, which were iPhones.

After getting past the initial panic of my errors, I resolved myself to use every waking moment I had to figure out how I can fix our frame rate issue and not let everyone down.

Messing around one day with an early level that did not have rendered backgrounds or interface elements, I noticed building to device an astounding bump of the frame rate to a totally playable 50 fps. I now had some clue; there was an issue with the rendered levels and UI which were basically just a graphic mapped to what’s called a quad. To explain, a quad is basically a flat piece of geometry. We have the whole game essentially graphically mapped in 2D on these quads that live in a 3D game engine. Each layer is a different set of graphics with the look of the level itself being the bottom-most layer with a giant transparent graphic painted on a square piece of geometry.

Doing research now focused on how the game engine rendered textures, I stumbled upon a realization that it’s computationally expensive to throw a graphic with a ton of transparent area on a quad. This is because in order for the camera to render the transparent parts, it needs to essentially go through each pixel to see if it needs to be blended at a certain transparency. This isn’t a huge issue with small things like a health bar in a game, but we were using it on the whole level. The answer was to cut out the geometry to match the graphics instead of using quads. So, if there was a rolling hill and plenty of empty sky above, we cut the geometry around the hill to discard the empty sky part.

We tested with one level, and it was a moment of joy as a completed level ran at 45 frames per second on a low-end iPhone! We were back to the excitement of having a game to launch. We fixed all ‘fill rate’ issues as it became known with all levels and everything else while we were at it.

The takeaway from this game was we learned a lot about the technical aspects of how to make a game run and be playable, but the most expensive lesson was how to actually make a game fun and balanced.

Banzai Blowfish was and still is a proud moment for our company as we all pulled together and made something that was just a dream. It wasn’t easy, but we saw it through.