

## Project Post-Analysis and Afterthoughts

---

**Project:** Explosive Dawn [First Playable Build]

**Date:** Fall 2010

**Genre:** 2D Shooter

---

### What is it...

---



Explosive Dawn, at its First Playable stage, was a much different-looking game than it is now. The concept is still very much the same however. It's a 2D shooter wherein players can get power-ups whose behaviors combine. For example, if you have the laser power-up and the spread shot power-up together, your gun will be a spread laser. The style was retro, featuring pixel art, 8-bit noises, and chiptune music.

This is my first game project working on a game team as a designer. Our first semester was absolutely chaotic. Our team began as two designers and three programmers. Our tech director was the sort of leader for the team. (He was the designer for my Freshman game, REMBlocks.) His technical interests were for something that involved particle systems and influence maps. That was his idea, and my fellow designer and I set to work on a prototype for his idea. By Engine Proof, we had a flash prototype of something along the lines of what he wanted. Our game engine was suffering, however, and it did not do very well when presented.

Immediately following that, our tech director left the team, and the other game designer got an internship, leaving a team with one designer (myself) and two programmers. After a meeting with one of our professors, we decided on doing something basic as possible—a 2D shooter like *Geometry Wars*. It was just something that would get us through the next milestone with the broken engine we had. We made it to our Prototype milestone with the very rough beginnings of our idea. It was called Fusionihilate, named for the main mechanic of weapon fusing, and our game name was Explosive Dawn, which one of our professors really liked.

Following that, we gained another programmer (one of the programmers from my REMBlocks team), and began expanding the game, which we named Explosive Dawn. We then changed our team name to Little Black Dress Productions (because it sounded sexy). By First Playable, we had a 2D shooter where you played until you died. There were three different weapons which could be combined and four or five different kinds of enemies, though they didn't have particularly great AI. It wasn't anything huge, but it was something.

## What went right...

---

We went through a bunch of twists and turns at every corner, and at every one of them, we bounced back. We just kept going, and we didn't let them break us down. Our ability to keep the team together and actually manage to produce something was probably the best thing that we did that semester. Despite everything that had happened, we still managed to get a very solid B on our project.

The game itself was an amalgamation of all of our desires for what made a good shooter. Also, we did a very smart thing and spent a lot of time researching before we did a single bit of actual design work for the game. That helped me a lot since I wasn't particularly familiar with the genre. It also helped us weed out which ideas we wanted in the game and which we didn't want.

Ultimately, we had a very basic, simple game that had a cool, somewhat unique mechanic and was somewhat fun and addictive, so even though it wasn't at all what we were expecting, it was definitely something worthwhile.

## What could have gone better...

---

Our tech was definitely weak, but we made up for it by scoping our project down. As much as I'd like to say that our tech would have been better if we had more people, I can't because I know that's not how it works. We had weak tech even before we lost our tech director. It made our presentations weak, and it definitely made us feel not very confident about our product.

The other issue that we had was that our initial particle idea never had a clear vision for exactly what we wanted out of our final product. We had TWO designers on the product (which in hindsight may have made things worse), and between the two of us, we couldn't get anything clear at all nailed down. It was always this amorphous sort of, "It's a particle...thing!" idea. Letting our vague tech ideas run our design was definitely something that didn't help us any.

Another thing is that once we decided that we weren't going to do the particle idea anymore, we decided on doing a game project from a genre that is very heavily saturated with all kinds of games. Also, I'd never really played many 2D shooters before, so it was a largely foreign genre to me. I'd originally been hoping to do something more original than just a 2D shooter, but we had to work within our tech and time constraints.

## What I learned...

---

One of the most important things I learned was just to not panic. We went through a lot of serious changes, and we didn't know if our team would hold together or not. However, everything worked out in the end, despite weak tech and team instability. Through it, just being rational and calm about everything and not letting all of the fears and worries get to the team and destroy it was a huge thing for us.

I also developed my skills in working with an idea that was not my own, which is something vital to my success as a designer because I'm going to be doing that in my career. If it were up to me, I wouldn't have done a 2D shooter, but the fact that we were doing that as a team made me learn to develop research skills and just general learning skills for picking up on design aspects of unfamiliar genres. Communication throughout this entire process was paramount because of the fact that it was a team generated thing, and we learned how to express our ideas and how to brainstorm, which I'd never thought needed to be learned.

For actual design-related things, I learned about how balancing a system works in a one player game as opposed to a multiplayer game. In a multiplayer game, balance is key, but in a one player game, it really just needs to be balanced in a way so that players win—though not always necessarily easily. That was something I'd never really thought much about until actually having to design a game like this myself.

Then, I learned about a lot of art things that I didn't know about before, like what sprite sheets are and how animations work. I had to not only learn how these things worked but also how to do them myself because we didn't have an artist on our team, so I had to do it all.

## If I were doing this project again...

---

If I had to redo that semester again, I'd probably need to break my self-imposed caffeine ban. It was stressful, but I learned so much. I would probably have tried to get myself more involved with the tech to help out the team, since I do know how to program. I maybe would have been able to program the game logic or some of the AI or something, allowing for the team to focus more on the engine stuff like graphics and physics.