ReBoot Postmortem

Team Unknown Space

Table Of Contents:

Things That Went Well:

Personal

I'm proud of the programming I did at the time even though looking through it now I see a million ways to optimize it. I was just starting to commit to the path of being a more tech focused designer. I was responsible for the particle system and all the audio systems and I think they add a lot to the game. I had never used the in engine audio effects and mixer before and I'm proud of the functionality I made, especially for someone with no prior experience. I'm proud of the amount of music I made for this game. I made 5 of the 6 tracks in the game (Leo Robles Gonzalez made the song Stikerz). I also made a lot of sound FX for the game that gave it an additional layer of feel that helped make the game more enjoyable.

Design

While we shouldn't have let it get to the point that it did, we were able to think on our feet and close the game loop with very little time. We put the self imposed deadline of the Champlain Game Fest on ourselves. While it was a nightmare at the time, it put us in a position where we absolutely needed to have a solution to close the game loop. While it was nowhere near what we wanted at the beginning of development, it ended up working for us.

Art

Our color pallet was very pretty and readable. We made some adjustments so that I can be able to see everything (I'm red green colorblind). The visuals are probably the strongest part of the game. One of our artists was a tech artist and her expertise really sped up development and made iterations very quick and easy.

Programming

The programmers really held everything together on this one. The designers kept throwing new problems at them every week and the programmers really did a great job keeping up. They also were able to communicate effectively when there was a feature that was out of scope and why we wouldn't be able to get it done. The lead programmer and I worked together closely to prioritize features to get the game in the right shape to be presentable. We communicated well to balance what was important for the design and what we would be able to accomplish for our dead lines.

Overall

Despite the hell that this game was to develop and the many problems we had, whenever we showed the game off in class, to peers, to testers, or at festivals, people liked it. They usually said that the game was fun and pretty. We were able to pull the game out of a

tailspin from something buggy, unplayable, with no core experience. Despite the game being very small and not what we set out to make initially, people liked the game and that makes me glad.

Things That Went Poorly:

Personal

I ended up siloing myself a few times to work on audio assets which, while making them was fun and they ended up being very valuable, I wasn't around enough to give adequate leadership to the team. The team needed a lot more big picture direction early on which wasn't there. I recognized this problem eventually and we had a team discussion in order to correct but it would have been better to avoid the problem in the first place. I wish I took more control over systems and asserted myself in that area. The other designers on this project and I talked 7 or 8 months after this project concluded about how we should have gone with the direction I wanted where movement was more focused on momentum, gravity, and using the curves of the bowl to gain speed. I should have defended my points more. That's more my fault than anyone else's

Design

We often ended up going with decisions we hated least rather than decisions that anyone really liked. Additionally, we tended to get hung up on things we didn't know. For example, players were having problems finding the speed surfaces in the map. We thought of a bunch of different solutions but didn't know which one was best. instead of just implementing one of them we were more confident in and moving from there, we implemented none of them. We learned too late that it's better to quickly go with a solution that will *probably* work and then testing and iterating on it.

Art

Designers didn't have a level planned out until very late in development. This gave artists very little time to make art assets to populate the space. This led to a map that feels empty. Additionally, one of our artists wasn't really given enough space for her to be creative as they were over managed. There was a really lengthy process to get art assets in the game. Assets had to be modeled and textured, then reviewed and critiqued, and then finally put in the game. There is a whole folder of art that got rejected that, to my layman's eyes, look great but were never put in the game.

Programming

The programmers really did a great job but they still have room for improvement. There was a lot of time wasted early on. We spent a lot of time looking at how to do shaders in unity. At the time we didn't know that it was more trouble than it was worth for us. Additionally, there

was a lot of stuff in the game that should have been prioritized first. While I'm always the person to push for polishing as you go, we really should have focused our efforts on the mountain of bugs we had. We also had times where programmers were going unmanaged. This led to wasted time and resources.

Overall

Our team had the philosophy of "wait and see" for this project. We ended up not taking action out of fear that we would make mistakes. This led to nothing getting done. My teams after this project have a strategy we jokingly call: "fail faster". We aren't always 100% sure we're making the *best* choice but we're always pretty sure we're making a *good* choice that can be iterated and improved upon. Making a choice that doesn't work perfectly is never punished. We just work hard to improve things.

We had 2 couples on our team which led to dynamics that were hard to navigate. Having to worry about people's relationships and also manage a team added additional challenges to working with the largest team any of us had ever worked on at the time.

Our onboarding process was essentially nonexistent. Our design documentation was not where it needed to be. Part of our design team was going through a very difficult time in his personal life that very often got in the way of him completing tasks he was assigned. Many of these tasks were documentation. We also had no safety net or back up plan for these tasks not getting completed.

Not having these docs done, a lead design team that was often unavailable, and no adequate design process resulted in a team that was unguided, didn't know what they were making, and had no shared vision. We eventually fixed all of this but it should have been prevented and planned for.

Takeaways:

1) Understanding that making mistakes is part of the process.
2) A developer that looks out for problems and is proactive about fixing them is a good developer.
3) You should plan your project timeline at least a few weeks if not months out. If development doesn't go perfectly to plan that doesn't make your initial plan worthless, it just needs to be adjusted.
4) Being in a managerial role means you need to budget less time doing hands on work and leave enough time to help and manage other team members.
5) Design with intent backed up with research.
6) Budget extra time for mistakes because they will happen. This helps avoid crunch.
7) Communication is key.