Zachary Fredeman, Nico Esposito, Noah Decambra, Zoe Paulus,

Mac Comeau, Anthony Nisi, Adrian Taylor

Blades Of Balance

Postmortem

Written by: Zachary Fredeman, Nico Esposito, Anthony Nisi & Noah DeCambra

Version # 6

Date: 3-4-22

Credits

Producer

Zachary Fredeman

Programmers

Zachary Fredeman

Mac Comeau

Nico Esposito

Liam Haggerty

Artists

Noah Decambra

Adrian Taylor

Anthony Nisi

Zoe Paulus

Table of Contents

Document History

- Version 1.0
    - The original version of this document. Document was created from a template.
- Version 2.0
    - The brainstorm, collaboration, and tool sections were filled out as well as the basic information regarding the group and the game.
- Version 3.0
    - The summary and what went well/poorly sections were added to.
- Version 4.0
    - General section edits as well as the addition of concept art.
- Version 5.0
    - Added Possible Future direction section, rewrote game summary.
- Version 6.0
    - Final document edits and wrap up.

Game Overview

Genre: Competitive QTE(quick time event) game **Platform(s):** PC

 # Players: 2 Players

Setting: Fantastical forest

Look: 2d Isometric

Summary

Blades of Balance is a 2 player game where you compete against your opponent to see who is stronger, Life or Death. You will each choose a side, Life or Death, and engage in a sequence of quick time battles to determine a winner. Your goal is to gain control of all the land and defeat your opponent. For each round you win you will gain control over more land, and upon a loss you will be forced to retreat and lose some land. Upon achieving 3 more victories over your opponent you will have gained control over all the land and are declared the winner. The Story follows the final battle between the gods of life and death, where only one will come out the winner. As the 2 gods battle, the background changes to represent who has the lead.

The game was designed with a target audience of teenagers to young adults, and those who seek a small competition or test of skill. The game presents a competitive environment in which you can challenge your friends to a duel of skill. The game relies on simple mechanics of quick time events and key combinations to create a low barrier of entry, making it accessible to people of many ages and skill levels. It is a fast paced game with short rounds enabling those with small amounts of time the ability to enjoy a quick game. While the game is simple in design

it remains very replayable due to the fact that each person you play against will have a different level of skill keeping the game interesting and different each time you play it.

Tools

We made use of the Unity game engine to create our game, as that was the game engine our group had the most experience with. Unity has a bit of a learning curve to it as would any other game engine but there are plenty of resources online to aid in learning any new skills and to help solve problems encountered throughout the design process. The 3D models used for the project were found on Itch.io's free asset page. The assets were Gekkou and Kasa from a cyberpunk samurai package. After getting the 3D models they were remodeled to fit our game's theme of duality. This was done by modifying the Kasa asset to look more menacing and evil while the Gekkou asset was modeled to represent peace and life. Even their animations favored their style, Gekkou fights calmly and with a simplistic respectful style, and Kasa fights more brutishly and with aggressive attacks.

The Development team made heavy use of visual studio and unity to program and create the game. Visual studio very easily supports integration with unity and was an application most of us had already on are computers. Github was also used so multiple people could work on the game and push the changes they made to everyone else to help streamline efficiency. However, GitHub caused quite a few problems with merge errors and not working well with some machines, so it was sometimes easier to just email the changes made to someone else to add to GitHub.

The art team used lots of physical paper to come up with design concepts and then converted them over into digital media that could be integrated into the game. This was done through various programs, such as Photoshop and Illustrator. While Photoshop is great for image editing it isn't the greatest for creating art from scratch and importing it over to unity. There were also issues with not removing the backgrounds of images, forcing people to reupload them with transparency. They also used various free, browser-based websites, notably Pixilart and Pixkel for creating certain assets, like the title and background. For rigging and animations, Maya was used. Maya is an industry-standard 3D computer graphics application and is an often required skill for game designers who want to work with 3D modeling. First, the characters were imported from Itch.io into the Maya software and re-modeled to fit the theme of our game. After remodeling, they were rigged, skinned, and animated to work inside of Unity.

Collaboration

The team most often worked in person with each other, this allowed for quick and easy communication. It also allowed others to provide help easier when people encountered an issue or got stuck. Discord was also used to ask any questions when we were not in person as well as set up meeting times, and share some files with each other. It is a good idea to keep meetings in person as it allows rapid communication. It's also very helpful having a way to communicate effectively when not in person, which makes discord or slack very useful.

Brainstorming

To start the brainstorming process we created a Google Doc of all possible ideas to fit the theme of Duality. Once we had a bunch of ideas written down we went through each one and

elaborated more upon what the idea is and how it functions as a game, as well as eliminating any similar ideas and combining some ideas to make a more cohesive game. The next step was to look at the complexity of each game in the list from a programming point of view as well as an artistic point of view. With the allotted time of 48 hours to finish a game, it was best to go with a simpler idea instead of being widely complex so we could achieve a more polished game. We removed ideas that were a bit too ambitious to create within 48 hours. After we were left with only a few ideas we voted on which one people found the most interesting and went with that idea.

What Went Well

What went well was the ability of the group to be decisive on decisions within the game, what would work and what would not or what we did have time to implement and what we did not. This also let us decide on a topic very early on in the process to allow us to start working sooner rather than later. We also found it easy to meet up and work within the hours of the jam and most of the team members were available and there were no issues with dragging people in to do work.

The game itself also came out relatively alright. More things went well than was anticipated. The title screen and its various buttons did work as intended and would send you to where they stated (so the Credits button would take you to the credits screen), and we'll likely utilize that formula for any other future projects we take on. Some of us were surprised that the game did load, and the Tutorial UI worked as intended (although the Continue button was really small and not in the right spot). We weren't sure on the day of the presentation (Sunday) if

everything would work together in such a way that the game wouldn't immediately break (which thankfully the game, at the very least, did load, though it didn't work as intended).

What Went Poorly

Mainly what went poorly was the misinterpretation of how long certain things would take to complete. Many of the artists completed their work early and were unable to contribute further due to a lack of knowledge on unity, as learning the program in a weekend is not easy. This left a lot of room for the programmers to work, but since we did not make many builds of the game throughout development we did not know that the game didn't run outside of the unity editor. The usual issues with github also slowed the process as some members lost work and had to retrieve it with the help of others which caused more slowdowns, as well as some members doing things that they had never done before or had little familiarity with so much of the work took longer than anticipated.

The game itself did load upon pressing start and then going through the Tutorial UI, but after that, nothing worked as intended. Only one of the two-player characters was animated, and the character didn't animate like we expected it would when performing an attack. None of the UI prompts worked (so none of the button prompts would load into the UI, so you would have NO idea what button to press). Adding onto this, the background didn't work as intended (it did load, but it didn't shift to whoever was losing as we had planned). None of the tree sprites in the background worked, either, and didn't change to their respective side (so if Death was winning, the trees on Life's side didn't become decayed or withered).

This was likely because, as mentioned previously, the game wasn't really stable outside of the Unity editor, and since we didn't really do a lot of thorough playtesting with the finished build, we weren't able to gauge how well the game would work. To be fair, a lot of this was because we only had three days to build the game, but for future projects, we will try to do some more extensive and frequent playtesting with more complete builds.

Future Direction

Currently there are no further ambitions for the game moving forward. We believe that if the team was to continue working on the game it would look much more different than it currently does simply due to the conflicting art styles that clash in the game, not making it the most visually appealing.

However, this game can be taken in many directions from where it was left off. The game is fairly close to being fully completed and there haven't been any major bugs encountered so far. The game still needs the final art assets to be implemented as well as the animation of the Life character connected to the game script, and Background transitions to represent who is winning. Once the game is finished it can go many places. The current game configuration is set up for computer use, but with some minor tweaks it can be deployed to mobile applications. The game suits a mobile platform really well with its simple controls and low skill floor, making it an easy game for anyone to play. With the games' short round times, it's a nice game to play when sitting in a waiting room or you have a couple of minutes to kill with a friend. To make the game more user friendly it would help to add an online component so you can play the game when on your

own. As the game is currently only playable with 2 people, this will enable people to play the game whenever they want.

Individual Team Member Remarks

Zach/Programmer - More in-depth familiarity with GitHub and its uses for unity as well as the initial setup. Help formulate more concrete ideas and be able to effectively communicate them to the team to ensure everyone is on the same page.

Nico/Programmer - Help lead the team and stay on track, as well as get manageable goals from the beginning to finish the game on time.

Noah/Artist - I would continue to work on the rigging and animations of the imported 3D models, and possibly create our own models from scratch as opposed to redoing someone else's work. And be able to do minimal programming for animation triggers in Unity.

Liam/Programmer - With more practice I would be able to more accurately determine how long a certain code will take depending on my familiarity with the type of code.

Mac/Programmer - With more practice I would be able to complete certain types of code faster.

Zoe/Artist - I can practice more to get familiar with unity tools to be able to help more besides some of the basics.

Adrian/Artist - I could have learned how to code and help out the rest of the team. I plan on eventually learning how to better use Unity and other game making resources to better help my teammates in the future.

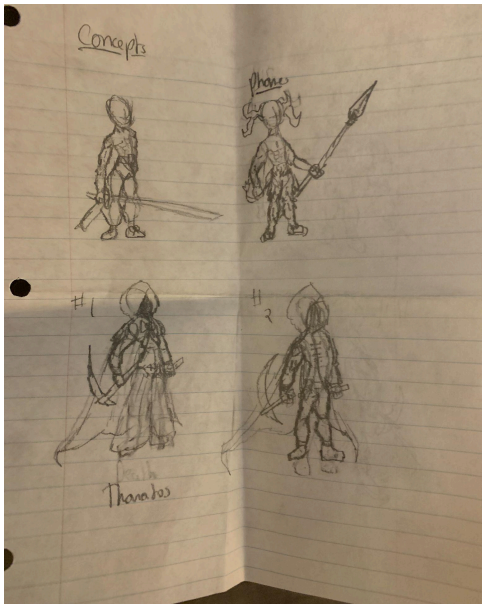
Anthony/Artist - I plan on learning more about Unity and other engines sometime soon, so as to be more independent (this did cripple me when we were forced to go virtual for a large part of the second day, and I wasn't able to help as much as I had wanted). I am personally happy with how a lot of things went regarding this project, though I am sad the game didn't fully work as intended. Still, I am happy with everything we did accomplish, and I do plan on expanding my skill set beyond art.

Reflection

This was the first game jam for well over half the group and was the 2nd game jam for the rest. It very much acted as a learning experience for everyone in the group. The time pressure to complete a project forced everyone to try out something new and learn it as quickly as possible. It also exposed the group to what work needs to be put in, in order to make a game, and familiarize them with the process. It was also a good way to help people gauge how difficult a task can be and that certain things like art can take lots of time to complete.

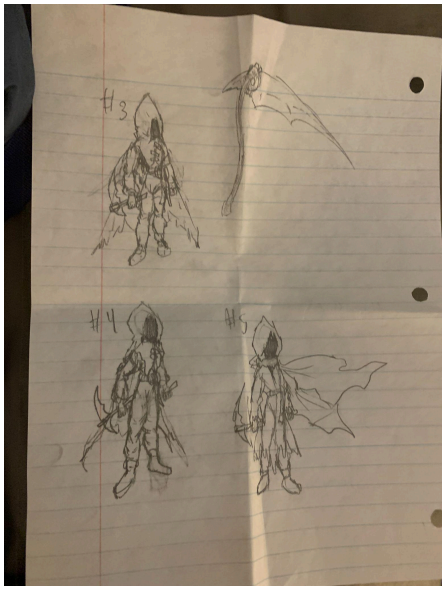
Concept Art

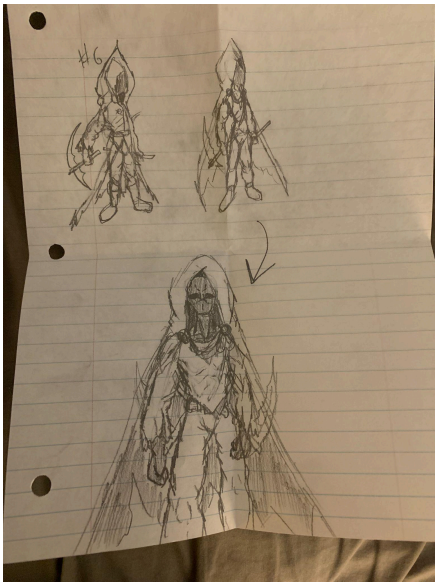
Here is some of the concept art that the artists made up when working on the game. They were initially going to go for a 2D-inspired game, and they created various concepts for both player characters. Since the theme was the idea of Life and Death, and their eternal war, they figured the player avatars should represent one of the sides (one for Life, and one for Death). So, they drew up some designs for the two characters.



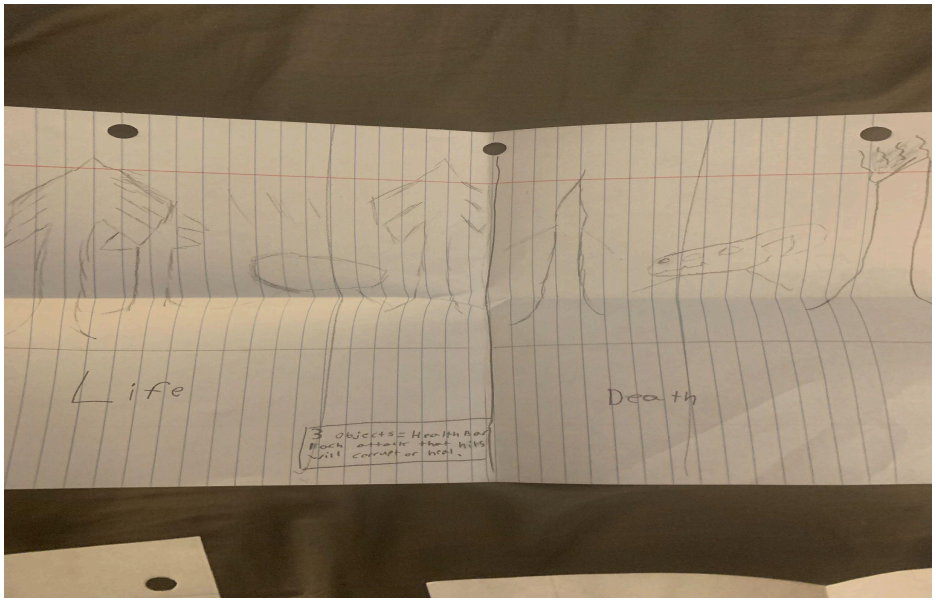
Life was very easy for them to conceive. Anthony wanted a Greek-inspired satyr warrior, but was stuck on the name. He wanted to name him Pan, but Adrian came up with a better name; Phanes, the primordial Greek god of creation in the Orphic tradition.

Death, however, was a lot harder for them to decide on a design (particularly his cloak and tunic). Anthony went through various designs, changing how the cloak would be worn, initially going for straps holding the cloak together, and then changing it to a scarf with a hood. He didn't like it, though, and instead changed it to a hooded tunic. He didn't like that either and changed it to have it loosely hang on his shoulders and held together by a chain.

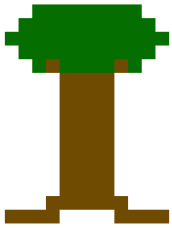




The final full sketch was the design they were really happy with. Anthony and Adrian came up with the name Thanatos (The Greek god of death) for the character. These designs would never be used, though, since they decided to switch to a 3D game with premade assets for the characters, so the two character designs never got used but their aesthetic was implemented in some of the changes to the imported models.



Early concept art for the background of the game that Adrian drew. This served as a baseline/foundation for what would become the background. However, certain things (such as the boulder on Life's side and the skull on Death's side) were excluded from the final design (going more for a dark forest in the background).



These were the early concepts for the three tree sprites that are in the background. Like with the early background, these served more as a simple and primitive sketch/early design (since Adrian was having issues with conceptualizing the trees), so Anthony created these trees as a baseline/foundation that Adrian would then expand upon for the final design.

Summary

This was the first game jam for many and acted as a great learning experience. There are several key things to keep in mind for future game jams as well as game development. While it is important to brainstorm many ideas, it's important as a group to determine which ideas are the most interesting and feasible with your current time limit, and individual skills. A game doesn't have to be super complex to be entertaining and it's usually better to have a more polished

completed piece than one that is missing many parts. Many of us are definitely going to be more resourceful and use other programs so as to be more autonomous and also be able to work outside of the main group should the situation arise where that is necessary (like if the weather gets bad).