Spaced Out Justification Document

By Amy Liu, Yelena Pham, Chris Pike, and Shane Schulte

The initial concept of our game was conceived during one of the in-class brainstorming sessions, where we essentially played telephone with game ideas. I think the first idea was a western style dodgeball game, which was adapted to an asymmetric 2v2 dodgeball game where 2 players placed walls and 2 players played dodgeball in a closed area. From there, we made the leap to a different asymmetric 2v2 game, with a pilot and engineer battling another ship in space. From that single brainstorming chain, I believe 3 of the final projects were conceived.

The first major iteration happened before we even began development; we were discussing the initial idea, and the topic of balance came up. The initial concept was more of a dogfight-esque battle, where the goal was the destroy the other team's ship. However, trying to balance the game between the engineer repairing and pilots dealing damage seemed like a daunting task. To remove some of the necessity from this, we opted to give the game an alternate goal - a race. This allows us to somewhat control how long the gameplay will last, regardless of how skilled the players are.

Since we were a bit behind some of the first few deadlines in class, we missed out on precious in-class feedback. To make up for this, we playtested the game together, and got feedback from various friends. We realized one of the biggest problems with the core mechanic of our game was that the engineer player had nothing to do at all if the two ships weren't relatively near each other. To fix this, we added a fairly strong rubberbanding mechanic - the second ship to go through each ring gets a slightly larger speed boost than the first. This closes the gap between the two teams slowly but surely, as long as the losing team can play well. We tried to come up with a design where a skilled team would win maybe 7 or 8 times out of 10 to a slightly inferior team.

Another problem we did not run into until we had other people playtest our game was the accessibility of our game. We fell into the trap of having done an extreme amount of internal playtesting and having done little external playtesting. We then finally had people who had never seen our game before playtest what we thought was a well-designed game. It was then that we realized that concepts that were plainly obvious to us, were not completely obvious to our players. Some of the things that changed were the general start of the game, aspects of the ship's interior, and mechanics for helping the pilot fly.

At the beginning of our development there was no real start to our game. The players assigned themselves their role and the action just started. After feedback from Austin and from playtesting we expanded the start of the game to ease players into their roles. After the players picked their roles, we started the game off which short explanations of all the mechanics in the game and what they looked like. These worked extremely well when players decided to read them, but was not extremely helpful when they sped through them. Since a majority of players would not take the time to read the explanations at the beginning of the game we added a short tutorial before the action takes place to force the players into learning how to play the game. First, the engineer needs to turn the ship on before anything happens. The engineer is locked in a room and is pointed at the engine they need to start. The engineer starts the engine with a button sequence that we use on two other systems inside the ship. This opens the door out of the room and starts the countdown for the pilot. To let the pilot figure out how to fly, we created a Y-shaped start to our race track. This gives the pilot time to figure out the controls for flying and shows the pilot to fly through the rings. Since each pilot starts off at each prong of the Y, they get this solo tutorial time before merging into the race track.

When the game first started out the ships' interiors were rather large and mildly confusing. Since we had played the game so much, this did not bother us while playtesting. When we had new players play the game we saw this as a pain point since our new players were getting lost in the hallways and did not know where the broken ship systems were located. Some solutions to fix this that did not make it into the game were painting hospital styled direction lines in different colors to guide the player and having a mini map that exposes what was broken and where it was located. Both solutions did not make it into the final game because we felt that the actual problem was the map was too large and these solutions did not simplify anything for the player. Our solution was to drastically shrink the ship's interior and place floating labels on each ship system that could break. The smaller ship made navigating easier for the player and the floating labels can be seen anywhere from the ship so the players know where to go. The labels would also change color from green to red to indicate that the system needed a repair. Another issue for the engineer was not being able to locate the enemy ship when they had pulled ahead. Not being able to find the ship that was winning led to that ship continuing their lead and almost always winning. Adding trail renders to the ships allowed the engineer to spot and target the ship with their turret, damaging the enemy ship and closing the gap between them.

Piloting was another issue we found that new players of the game were struggling with. Due to the inverted y-axis, players who were inexperienced with video game controllers would fly off course easily and have a difficult job of getting back on course. At first we decided to add arrows that would direct the ship back to the nearest ring the players needed to fly through. This was a pretty good solution, but even this helpful mechanism was not enough when players drifted way off course. If both pilots ever became this far-off course, the game could last indefinitely. We decided that we needed to add a mechanism that would bring players back to the course, much like being brought back to the track in Mario Kart. We let the player reset themselves when they too far off course. This allows the players to make progress in a situation where they would not be.

In conclusion, our game has come a long way through all of its iterations and playtests, but we are all happy with the final product.