Spaced Out Research Document

Amy Liu, Yelena Pham, Chris Pike, Shane Schulte

Intro

Spaced Out is a 2v2 multiplayer racing game in space. Each team is composed of a pilot and an engineer. The pilot navigates the spaceship through the race course, shoots the main guns, and communicates damage reports to the engineer. The engineer is located inside the spaceship and is in charge of firing the secondary guns and repairing damaged ship systems.

In order to win the race, each team will have to navigate through asteroids and try to slow the other down by shooting at them. As a ship gets hit, a random component will break, and different damaged parts will have different negative effects. For example, if the engines are broken, the spaceship will stop. If the gravity generator is broken, the engineer's movement will be affected. If there is a hull breach, the ship will be slowed down.

Spaceteam

In the mobile game Spaceteam, the players are working together to crew a spaceship in order to escape an exploding star. Each player is assigned a random instrument panel and has a command prompt on their screen. The command prompts displays a time-sensitive command which has to be fulfilled to prevent the spaceship from crashing. The commands can correspond with any of the other player's instruments panels so the players have to shout the commands at each other, making the game extremely hectic as four people are all yelling technobabble while also trying to listen and follow the commands that match their instrument panels. It can get crazy, but all of the yelling makes the game a fun, albeit intense, experience.

That frantic communication between players is something that would benefit our game, and we brainstormed ways to incorporate it. Our game ended up with a similar mechanic in which the ship damage report is displayed on the pilot player's screen so they will have to tell the engineer which components are broken. This will result in more interactions within the team, with the engineer having to decide which components are higher priority than others.

Lovers in a Dangerous Spacetime

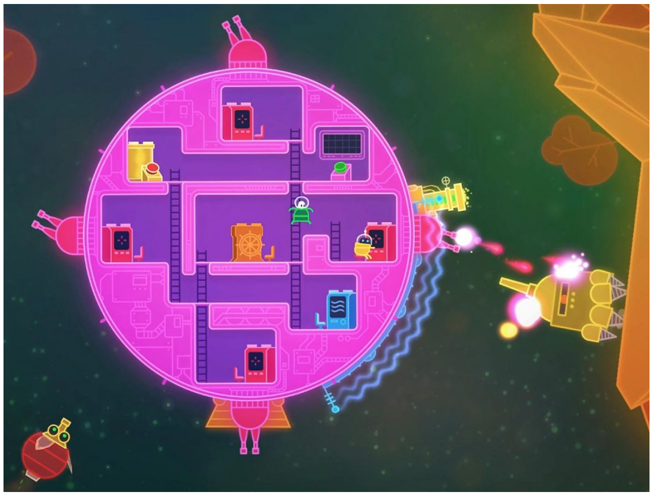

Figure 1. An example of ship design in this game.

Lovers in a Dangerous Spacetime is a 2-player co-op game in which you shoot robots and save the universe from evil. While designing our game Spaced Out, this was one of the first examples that came to mind that affected our decision-making. We used it to consider ship design (both internally and externally), map design, health, and the overall goal of our game.

First of all, you can see in the above photo that the game is 2D and shows a slice of the whole ship. We decided to make our game 3D because we believe that will make more of an immersive experience for our players. That decision led us to move away from the design in Lovers in a Dangerous Spacetime because with a 3D game, having a bird's eye view of the ship doesn't support the experience we're looking for. So even though that ship design has benefits in its simplicity for the user, our game actually works better by giving our engineers a first person perspective and allowing them to explore the ship, and our pilots a third person perspective.

Secondly, our decision to make the game 3D also plays into effect for our overall map design. Lovers in a Dangerous Spacetime uses the whole monitor, as the two players can never leave one ship. However, we decided to split the monitor into four screens, one for each player, as there are two ships with two people controlling the interior and the exterior. This design might make our game a little more confusing to understand where in space everyone is, but it does make it more realistic.

Furthermore, our decisions in measuring the overall health of the ship contrasts with this game. In Lovers in a Dangerous Spacetime, the ship itself has health and as it takes more hits, you can see the shell of the ship begin to crack. There's no way for the players to directly increase this health except by completing mission goals or by finding planets with bubbles of love. On the other hand in Spaced Out, a random component will break and the engineer needs to fix it in a timely matter. There is no measurable health component and no one in the game can actually die.

Finally, the goal of Lovers in a Dangerous Spacetime is to defeat evil and rescue your friends. It's a cooperative game. While there is a co-op aspect to Spaced Out, it's ultimately a racing game and you want to beat the other team.

Sea of Thieves

Sea of Thieves is a pirate game that has mechanics based on both players cooperating and players competing. The game forces players to cooperate by making it impossible for one player to sail the ship alone. For example, when the sails are let down, the player steering the ship can't actually see where the ship is going and has to communicate with another player standing at the bow in order to not crash the ship.

Additionally there are elements of player versus player since different crews can engage in naval battles. In a naval battle, crew members have to divide up the tasks of steering and sailing the ship, shooting cannons, and patching holes in the hull. This creates many interesting player choices since the players have to decide who and how many people are manning the cannons versus bailing water and fixing holes. The player experience of the team cooperation during a naval battle is quite different to spaceteam since it lacks the frantic energy of spaceteam due to the slower nature of actually aiming and shooting the cannons rather than mashing a button on the screen.

Also the player experience in cooperating with teammates in Sea of Thieves can be much more rewarding since victory over the other crew is possible whereas Spaceteam always ends with someone messing up and everybody losing.

Captain Sonar

Captain Sonar is a team based hidden movement, resource management, party game. Four players on each team assume the roles of the Captain, the First Mate, the Radio Operator, and the Engineer. The Captain shouts which direction the submarine will be moving, decides when the submarines systems will be used, and keep tabs on what the other players are doing. The first mate will decide how the sub's systems will get charged and when they will be available for use. The Engineer must repair the submarine since each movement will put wear on one of the ship's systems. The Radio Operator will listen to the opposing team's captain tracking their every move in hopes of finding out where they are.

Every player has their individual role but each job requires interaction with your teammates. The captain cannot use a torpedo if the first mate has not been charging it, the engineer hasn't fixed it, or the radio operator doesn't know where the enemy is. The game avoids the problem other cooperative games face with one person quartering backing decisions for the whole team. The team truly needs to work together to defeat the opposing team.

A major design decision in Captain Sonar is the pseudo hidden movement aspect of the game. This hidden movement is the setup for the radio operator role, but also helps shape the tone of the game. When you are hidden from your enemy, the feeling of the game is not hectic but it is suspenseful. In the early rounds of the game, you can kind of ignore your opposition while you take care of your early game plan. I think this kind of non-interaction does not work well in a video game. We want the teams to be interacting right from the beginning. We want your decisions to matter as they are being made instead of being delayed. This should add to the hectic, urgent feel we are trying to bake into the game.

The hidden movement aspect in this game also shapes the interaction experience between the two teams. The Captain needs to audibly communicate their movement or use of the systems, but that is all that is required. If the players wanted to get into the game, their table talk should be kept quiet or if they wanted to play the part, intentionally misleading. This inherently changes the interaction of the teammates as well. The players will talk in hushed whispers, talk at a normal volume to throw of the opposing team, or just not talk at all and infer what the player wants to do by looking at their board. No communication or limited communication is something we absolutely do not want to have happen in our game. We the players to start working together from the very start of the game all the way till the very end. This way they can share the same experience and share their accomplishments or failures together.

Having teams of four on Captain Sonar sounds like a great idea. Eight people playing all at once having a great time. I wish this could be explored in our game, but sadly it cannot. I do however believe that trying out an eight-player game would come up short. I think two players per ship in our game is the correct number. This gives two very fulfilling roles to our players. These should keep our players extremely occupied and in constant communication. I think the four players in Captain Sonar dilutes the fun for each individual role, but makes up for it in the sheer number of players. If you look at each role in Captain Sonar, there is not a whole lot of substance. The players are kept occupied because they are trying to complete their task in a reasonable amount of time. Each role is not required to interact with the three other roles. The player can talk to another player to their left or right for the whole game and still have a successful game. I think our game is going to more engaging and rewarding since we aren't diluting the roles for our players and we are trying to keep the players in constant communication.

Guns of Icarus

Guns of Icarus is online multiplayer airship combat game. It is set in a steampunk universe where your airship is a combat-ready hot air balloon. The three roles in this game are Pilot, Engineer, and Gunner. The main difference between these roles are the number of equipment slots each player gets for their class. Pilots get more pilot slots, engineers more repair slots, and the gunners get more ammunition slots. The player will choose their class and their equipment loadout. The pilot will fly the ship, the engineers will repair parts of the ship, put fires out, and buff the ship, and the gunner will shoot at the other ships while keeping their weapons working. These roles do not need to be strictly followed. Anyone may leave their current post to go help anywhere on the ship.

The ships are divided up into the helm, weapons, the engine, the balloon portion and the hull. The helm of the ship is where the steering wheel is and where the player who wants to navigate the ship will need to go. The engine is made of up a large engine that controls the velocity and two smaller engines for turning. These can break and will affect movement and turning if broken. The balloon portion affects the ship's vertical movement. If the balloon is broken, the ship will start sinking and cannot gain altitude until it is fixed. The hull is the health of the ship. The hull has a shield that needs to be completely depleted before taking actual damage and the shield is repairable. If the shield is depleted and takes more damage, the actual health of the ship is lost. If the ship loses all its health the ship is destroyed, and the health lost cannot be repaired.

I think this game most closely represents what we are trying to accomplish. A minor difference between these two games is the theme. I usually disregard theme as an element of the game that has little impact. Many games have been re-skinned or re-themed that still play like the original. But theme can creep their way into the game design.

Here we have steampunk with the vessel of choice being a large monolithic hot air balloon. Thematically these cannot be quick or agile. By using space as our theme, we choose to have spaceships as our vessels. Using spaceships allows for our game to be faster paced and quicker than Guns of Icarus. We want to have a faster pace to help facilitate teamwork and communication. Our game would miss the mark for being a great cooperative game if our teammates played their own single player games in parallel with any interaction.

One aspect of Guns of Icarus I noticed that I thought was done perfectly is the interaction between the gunner and the captain. The weapons on the ship do not have the greatest mobility. They overlook the side of the ship they are on and cannot be moved roughly more than 30 degrees left, right, up or down. The pilot is perched on the upper most portion of the ship, and can only reasonably guess what can be aimed at. This restricted aiming for the gunner and the restricted worldview of the pilot forces the gunner and the captain to work together to help line up shots for the gunner. After watching a few games of people playing, this design and the most profound impact on the player's interaction. I hope we can emulate this design decision because I think it is done perfectly.

Mario Kart Double Dash

Mario Kart Double Dash is an interesting spin on the typical Mario Kart game, which also has several elements that are similar to our game. The ‘spin’ is that each kart has 2 characters instead of 1 - a character in the front, driving, and a character in the back, using items/powerups. Additionally, there is a mode to the game where each character is controlled individually, allowing a teamwork/cooperative aspect to the game.

One of the places where this game fell short was in the cooperative mode. The roles of the players were asymmetric, similar to our game, but they are also able to switch their roles, unlike our game. The main problem that this creates is added complexity - the player has to know how to play both roles in one single playthrough, which steepens the learning curve. In a game like Mario Kart, you don't really want to have any barrier to entry. It's intended to be a party game that anybody can pick up and play with very little understanding, and potentially win.

The other big problem with the cooperative mode in this game is the fact that it required very little communication to complete the goal. This succeeds in simplicity, but it makes it feel far less like a team game and more just like 2 different games for the different roles. There's no real sense of collaboration, which is something we need to be careful to avoid in the design of our game.

One obvious shared aspects between the two games is that they are both races. In Mario Kart, you race to reach the finish line first, picking up various items/powerups along the way. This is a tried and true game model, as Mario Kart games are widely successful. However, moving the environment to space presents a level design challenge of enclosing the 'track' without actually enclosing it physically. The typical way to implement a 'track' in any sort of flying game is to use a ring checkpoint system. You have to fly through various rings in order before you get to the goal. However, another aspect of Mario Kart to keep the party feel is the forced comeback mechanic - players near the rear of the race have an advantage over the players in front of them. Implementing this with only 2 ships and a ring based flight system is quite the challenge. One solution is to lock some of the better weapons to be front facing only, and make sure the track has a general forward progression; this would make sure that the team that is behind has access to better weapons to use on the ship in front of them, therefore allowing for comebacks to happen more easily.

Finally, Mario Kart has a system in place for when players fall off of the track: they are picked up and respawn, while being handicapped for a few seconds to penalize them. This is hard to implement in space while keeping with the 'comeback' sense, because what really defines going off of the track? One idea to create a similar effect is to calculate the distance the ship is from their current checkpoint goal; if they reach a certain threshold, they can be gently nudged in the correct direction. If they go even further, perhaps their controls can be revoked entirely while the ship corrects their path. This would help ensure one team with a bad/learning pilot can still play the game without becoming unfairly behind (to the point where there's no longer any interaction between the 2 ships).

Keep Talking and Nobody Explodes

Keep Talking and Nobody Explodes is a multiplayer cooperative game where one player views the screen, showing a virtual room with a ticking time bomb they must defuse. The other players are bomb ‘experts’ who aren’t allowed to view the screen, and must communicate with the defuser to identify the components of the bomb and work together to defuse it using a paper manual. This separation of information creates an environment where players are forced to communicate verbally to win. Our game seeks to do something similar, although to a lesser extent. This mechanic super interesting, and adds a layer of depth and teamwork to the game by inter-connecting the player’s jobs. Obviously, Keep Talking and Nobody Explodes takes this to an extreme end of the spectrum by separating the players physically and limiting their view. The primary method they use to create fun is by overloading the players with information, and making them sort through it quickly in order to win. The pressure of being able to see a clock ticking down adds to the suspense of the game.

There are, however, other methods of implementing necessary communication. For example, creating a scenario where 2 players have to do something simultaneously can be an interesting mechanic, especially if only one player is told about it. Requiring an input from the pilot during a sequence to fix a component could be interesting - imagine during a repair, the engineer’s screen suddenly says “Have the pilot reboot power systems (press X) in 5… 4… 3… 2… 1…”. The engineer would need to process this information and quickly relay it to his teammate in order to complete the repair; otherwise, it would fail. Another example would be only showing the health readout of various systems to the pilot - he’d have to tell the engineer when the engine health was dangerously low so the engineer could go and make repairs.

Another interesting aspect of this game is how easy to pick up it is. It has various levels of difficulty, but the idea of the game itself is super intuitive. There isn’t a lot of strategy involved - it’s very clear how you’re meant to play, it’s just a matter of skill and execution. This is something we’re hoping to achieve with our puzzles for repairing the ship - very straightforward and obvious, just a matter of performing them accurately and quickly so that you can move forward to your next task. The whole challenge of Keep Talking is that none of the tasks are inherently difficult - it’s just a matter of performing fast and making sure you are correct, because in that game the consequences are much more severe if you make a mistake (boom!).