Fire Flight

By TinyDuck - Postmortem

Two to four party phoenixes battle head-to-head to light up the sky on the festival of flames, shooting elaborate pyrotechnic patterns in a twin-sticks shooter meets bullet hell game! Who will be the ruler of revelry?

Description

Summary

Fire Flight is a top-down twinsticks bullet hell party game where you and your friends each choose a loadout of firework patterns to shoot at each other and dodge. Essentially, everyone plays as a bullet hell boss trying to knock out other bullet hell bosses. The game is defined by its controlled chaotic nature and a palpable competitive edge.

The basic rules are that each player controls a phoenix in a top-down environment. These phoenixes can shoot waves of projectiles at each other in the form of firework patterns and sparks. Getting hit by an enemy projectile causes you to lose one of three lives; when you run out of lives, you are eliminated. The last phoenix standing wins!

Mechanics & Systems

Player Control

Players can move using the left-stick of a controller (or WASD for keyboard). Any input direction is accepted, and we have manually reduced the deadzone for this action so that certain diagonal inputs don’t end up getting dropped. Player aiming is done using the right-stick (IJKL for keyboard) and like moving can be used to aim in any direction, however, the player turns towards the input direction with a maximum rotation speed. This means that a player cannot instantaneously go from aiming down to aiming up, but must wait for the player object to rotate towards the direction they wish to aim.

Offensive Tools

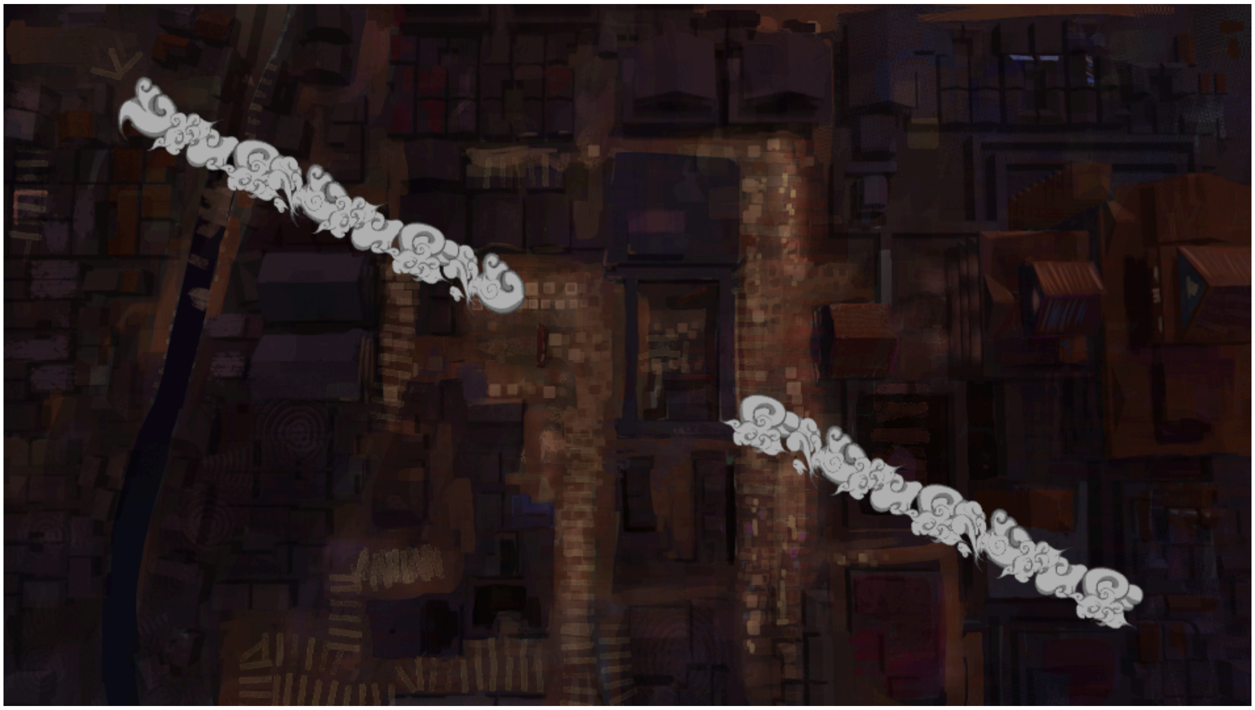
Players are able to select two different attacks before each match. Each attack has unique properties like varying spin, spread, speed, etc. attacks are fired in waves with each attack having an initial resource drain for the first wave and a drain over time effect for firing consecutive waves.

Defensive Tools

Players have two defensive tools that are mapped to the left bumper and right bumper. The first of these tools is the self-slow which allows the player to slow down their movement speed while holding the left bumper. While slowed, the player's "hurtbox", or the collision area which causes them to take damage, is shown brightly while the rest of the player sprite turns a darker color. The second defensive tool is the smoke bomb. The smoke bomb is mapped to the right bumper, and when pressed, causes all enemy bullets within a certain radius to disappear. This area of effect is shown through an animated swirl of smoke.

Environment

Clouds are environmental hazards which when overlapping with a player slowly drain their resources, damaging the player if they run out. Players in clouds are slowed and attacks are blocked by clouds. Each map is surrounded by clouds which after a period of time close in on the players forcing close combat situations. Maps also may contain moving cloud formations that scale, translate, and rotate in unique ways adding variety to the environment.



These maps are designed to provide interesting cover while attempting to avoid "camping" strategies. Some of the maps divide the play space into smaller distinct areas, while others have one large zone with changing constraints, such as the walls getting tighter or cloud cover shifting positions constantly.

Powerups

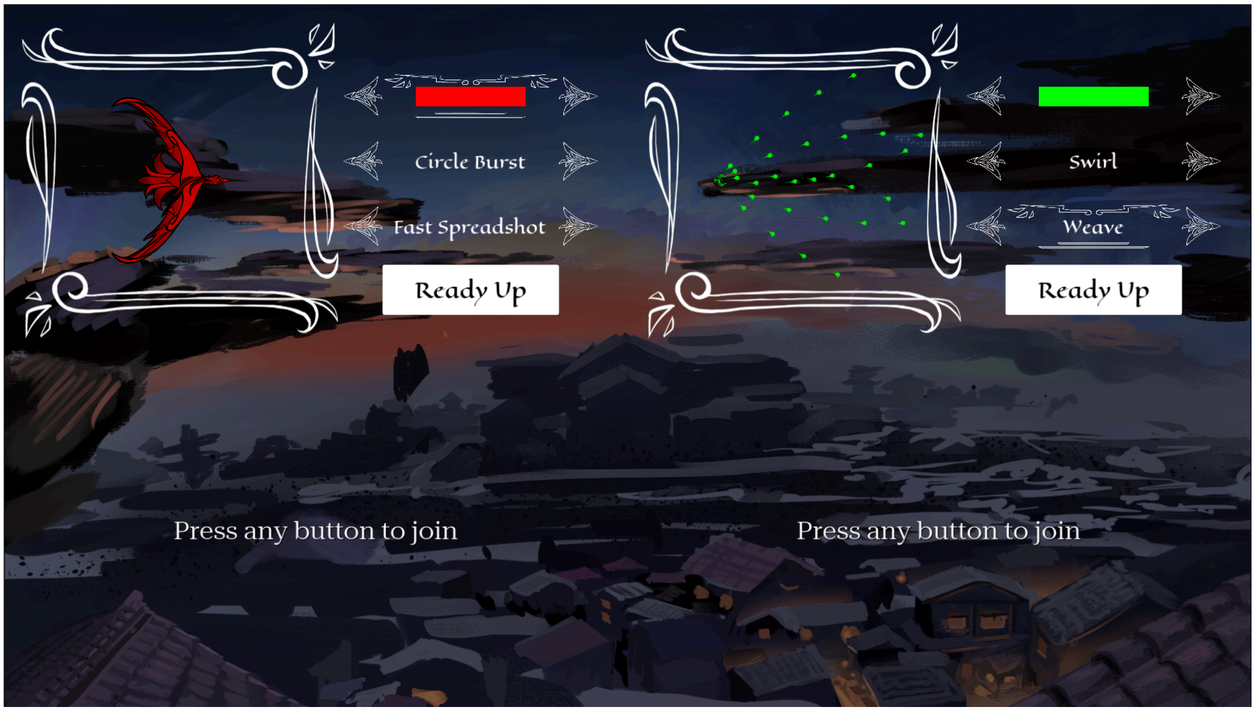
Power-Ups are pickups that are designed to give some benefit (over their opponents) to the player that picked them up. Only one powerup pickup can be on the map at any given time, with the previous one disappearing after pickup or a new one being spawned - whatever happens first. In the current iteration of the game, a new powerup spawns every 10 seconds.

At this time there are 9 unique power-ups. Most are passive effects that either affect player movement, firing speeds, or player abilities; these might apply for either the current player or their opponents. Most are active over set time spans, though a few may be instant and have an immediate upon being picked up. Many of these are immediately obvious, but for those less so, a unique symbol will appear below the player's energy meter, signifying which powerup is currently active. On top of this, popup text will appear over the player conveying the powerup name upon pickup, giving the player the immediate ability to understand what they do. The few powerups not associated with player abilities all have evident visual effects that accurately convey their function to the majority of players.

UI

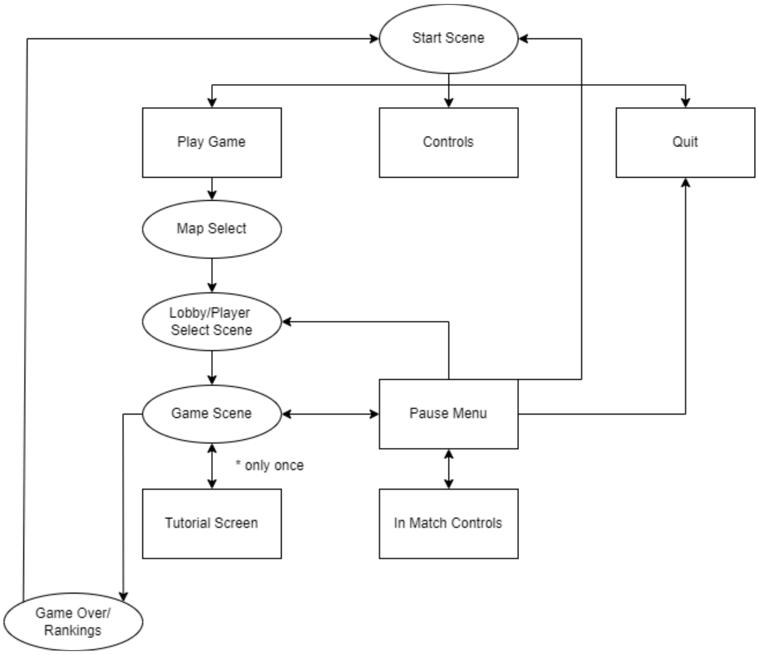
Design

There are two sides to this part of design: the UI and the UX. Starting with the UI, we decided to utilize a gestalt-based artstyle for the UI elements, with buttons and indicators drawing elements from the phoenix designs and flowing style. This decision combined minimalism with intricate patterns in a surprisingly effective way. The gestalt approach meant we could avoid littering the screen with visible buttons and excess bordering, while the intricate patterns of the few UI elements that do exist make up for the empty space of the rest of the UI.

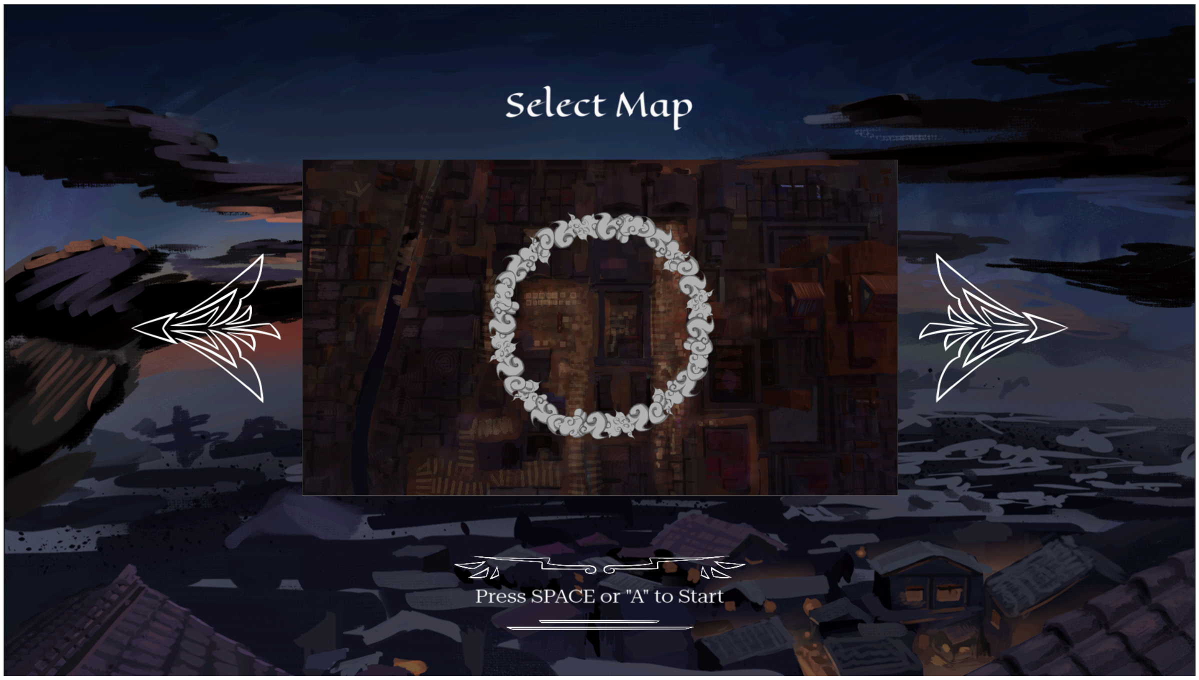


On the UX side, the simplicity of the menus with their minimal buttons and streamlined nature allows for players to get into a match as quickly as possible after launching the game. This is important because, as a local party game, players likely want to get to the action as fast as possible, since they're here for a fun time with friends. While working on the UI and UX design, we made sure to keep the menus as unobtrusive as possible so it never feels like the game is getting in the way of the players having fun.

Navigation



When the game first loads it opens to the Start Scene which has the game title, background art, and a menu with 3 options, Play Game, Controls, and Quit. Selecting "Controls" will open a pop-up diagram with a game controller showing all the game controls. Selecting "Play Game" will progress to the Map Select scene, where you can select a map to play on through a single image carousel menu.

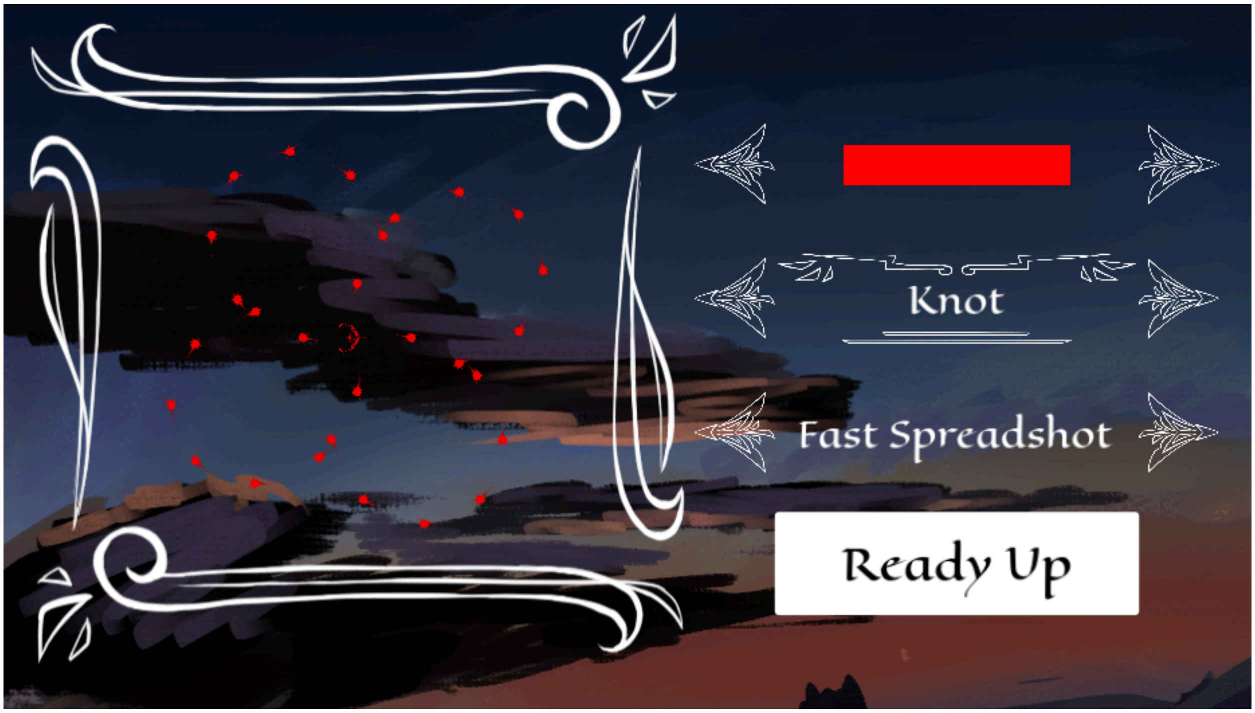


After selecting a map the game will transition to the Lobby / Player Select scene. In the Lobby players can join by pressing any button on the controller which will generate an individual player selection screen, where a player is able to select their color, and 2 different attack patterns. After all players ready up a prompt will show up allowing the players to start the match. Upon entering a match for the very first time a tutorial will pop-up, this will teach the players about the live system, the effect of clouds, how the heat meter works and the smoke bomb ability.

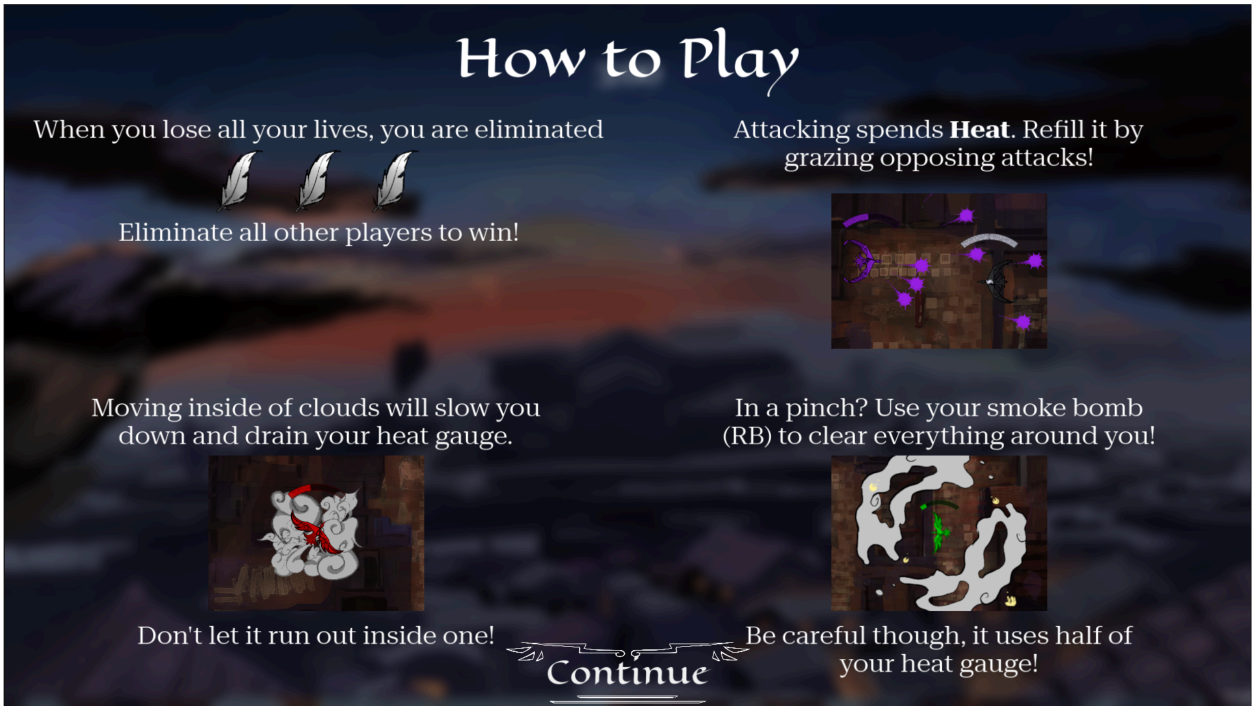
After a player has read the tutorial they can confirm that they’ve read it and once all players have read it pop-up will disappear showing the match screen. Inside the patch players are able to fight each other, they can also pause the game which will open the pause menu, which has the options to Resume, view Controls, return to the Lobby, return to the Main Menu, or Quit. Pressing ‘Resume’ returns the players to the match. Pressing ‘Controls’ generates a separate pop-up that shows the controls like in the start screen, the players can then return to the Pause menu. Pressing ‘Lobby’ returns to the lobby where players are able to change their loadouts. Pressing ‘Main Menu’ returns all the way back to the start scene, and Quit closes the game. Once only 1 player remains the game ends which transitions to a game over/rankings screen, showing the ranks for surviving the longest, and allows the players to loop back to the start screen.

Onboarding

The onboarding in Fire Flight aims to be as unobtrusive as possible in order to allow for players to quickly pick up the game and start playing ASAP. To this end, we tried to “show” rather than “tell” where possible. The best example of this would be the attack previews in the loadout selection screen. Without any text, these previews allow the players to learn about their attack options at their own pace.



Unfortunately, some of the more subtle mechanics require more clear explanations. The best solution we found for explaining these is through a quick tutorial screen with an emphasis on images that shows up before the first match the players start each time they run the game program. Given that Fire Flight is a local party game, it’s very likely that each time someone runs the game they’ll either be with someone who hasn’t played before and/or haven’t played in weeks or months. This one-time tutorial both introduces new players and jogs the memories of seasoned players and then gets out of the way for the rest of the play session.



How to Play

When you lose all your lives, you are eliminated

Eliminate all other players to win!

Attacking spends **Heat**. Refill it by grazing opposing attacks!

Moving inside of clouds will slow you down and drain your heat gauge.

Don't let it run out inside one!

In a pinch? Use your smoke bomb (RB) to clear everything around you!

Be careful though, it uses half of your heat gauge!

Game Loop

Pre-Match

After pressing “Play Game” in the start screen the players transition to the map select screen. This screen is a simple carousel like menu where players can see preview images of each map. Any player is able to select the map and choose to move forward to the lobby.

After map select is the lobby screen. During this screen players can join the game and customize their player by changing their color and each of their two attacks. When hovering over each of these choices the player animation and attack animations are displayed to give the player an idea of what their color and attacks look like. After every player that has joined has readied up a prompt appears on the screen which allows for any player to press enter or start. After pressing start or enter the scene then transitions to the main match scene.

Match

Games start off with up to four players spawning in each corner of the map. The UI will show that all players have three lives, and a three-minute timer will begin counting down. In the opening seconds of the match, there will be no damaging attacks/objects on screen and no powerups spawned yet, allowing for players to strategically position themselves and test out the controls and their chosen weapons before waging flashy war against each other; though this time will be brief.

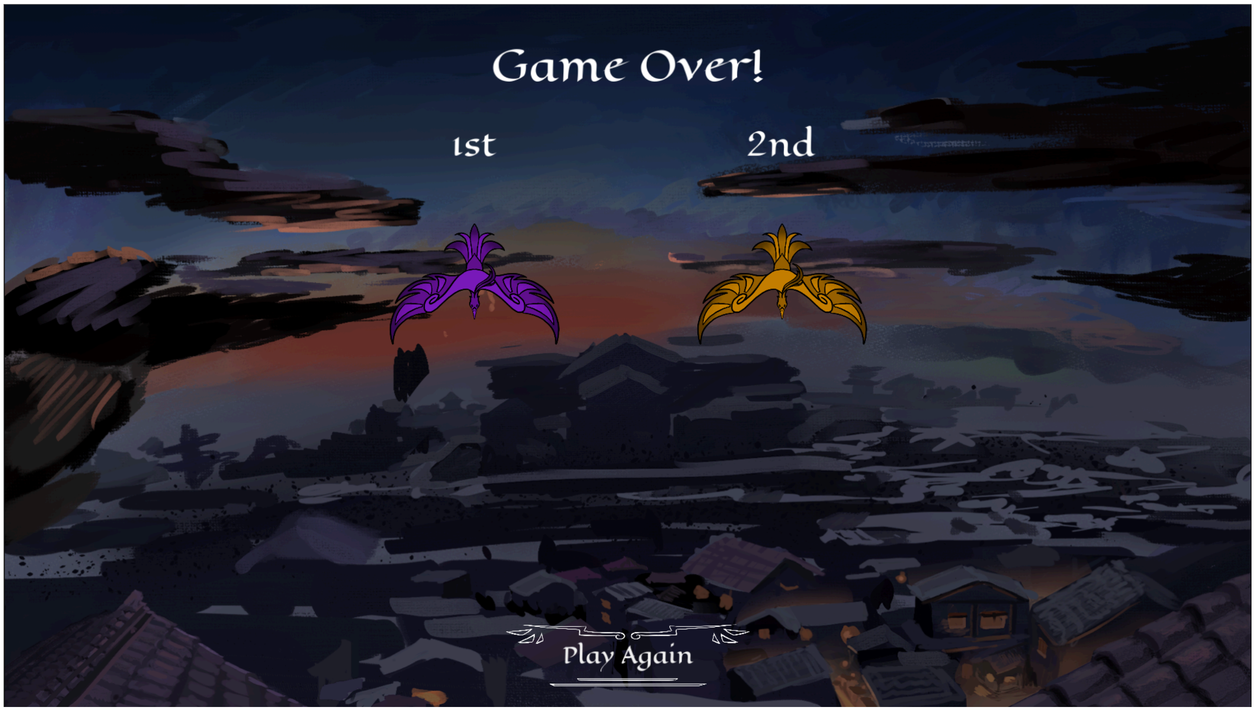
The length of the game will see players using their weapons, which launch fireworks in unique patterns, to damage their opponents. When an opponent's firework hits a player's core, which will become visible as a firework nears them, they will take damage and their ui will show them losing a life. After losing 3 lives, a player will disappear and be out of the game. In order to fire their weapons, players will need to expend heat. How much heat they have can be seen in a meter above their heads. Different weapons have different costs of heat, changing the viable strategies depending upon which two weapons the player selects. Heat will also be expended if players use their smoke bomb ability; This ability allows the player to clear several enemy fireworks around them, allowing them to better defend themselves. Players will slowly regenerate heat over time.

Aside from other players, maps may have elements that players also need to keep track of. Some arenas will have clouds in them. These clouds will block projectiles, and may move, rotate, or resize depending on the map. If a player is inside a cloud, they will be slowed significantly, and their heat will be slowly drained. If a player arrives at 0 heat while within a cloud, they will take damage; this is the only other way players can take damage outside of fireworks. The border clouds on all maps will function similarly to the aforementioned clouds. Also within the arena, power ups will occasionally spawn. Power-ups will give unique benefits to the player who collects them. (For a more in depth explanation of Powerups, refer to the power-up section above).

The game will only end when there is one player left alive. If the timer reaches 0 before this state is reached, the game will enter sudden death mode. During this time, the border clouds will close in, forcing the players towards the center of the arena. The reason for this is to up tension and cause games to end quicker. With players funneled towards the center of the map, they will have less time and room to dodge attacks or use defensive strategies, allowing for a culling of the remaining players.

Post-Match

The results screen displays each of the players from left to right in order from player 1 to player 2 and so on. Above each player is the position that they came in during that round, with the player who died first being in last, and the player who survived till the end being in first. We keep the data which tells us which players came in which position is a script that transfers between scenes that is updated every time a player dies and when the game ends to keep track of which player deserves which position.



We decided that players may really enjoy the loadout they used in the previous match and may want to use that loadout again. The game will save the player loadouts between matches and when returning to the lobby will pre-populate the player select menu with the previously selected options, meaning that if you join the lobby after a previous match the defaults will be the loadout you used in that match. This is done but not clearing out all of the loadouts, instead right before the match is loaded in it will check if there are any loadouts saved that aren’t assigned to a player and then clear those loadouts to prevent “ghost players”.

Project Goals

Chaotic Fun

As a local party game, fun is the key form of engagement we’re trying to evoke in the players. More specifically, we aimed to create fun by combining the satisfaction of dodging difficult attack patterns in a bullet hell with the frantic atmosphere felt in many of the best local multiplayer games. Combat should feel overwhelming with all the bright colors and dozens of projectiles all moving in unique ways, and mastering and overcoming that chaos should feel rewarding.

For the most part, Fire Flight succeeded in creating this atmosphere, particularly in three-to-four player matches. However, in two player rounds, the game could feel a bit slow, with playtesters finding it difficult to hit a sole opponent and too easy to dodge attacks when they’re only coming from one opponent, especially with three lives to work with. In retrospect, this makes sense since the main challenge is from overcoming the challenge of dodging overlapping attacks from multiple opponents at once.

Skill Curve

With the many patterns and aspects of battle within the game we wanted to allow players of differing skill levels to compete with one another. As such, the many attack patterns of the game allow for both safe and aggressive styles of play. Narrowly dodging bullets is rewarded with extra heat regeneration, but players also have the safer option of the smoke bomb which clears bullets around them at the cost of energy. Game knowledge is helpful and gives you an edge but is not necessary to win.

The varying attack choices also presents room for strategy and skill expression. Early on, players tended to latch onto fast attacks or large attacks that take up lots of space. However, for testers who have played more, sometimes they would try out more unconventional attacks which, when used properly, could throw off opponents with confusing patterns and lock opponents into disadvantageous situations. When players get better at the game, aspects such as unpredictability and area control become more important to them than the simpler attacks, since those more straightforward attacks end up being easier to dodge after playing against them more.

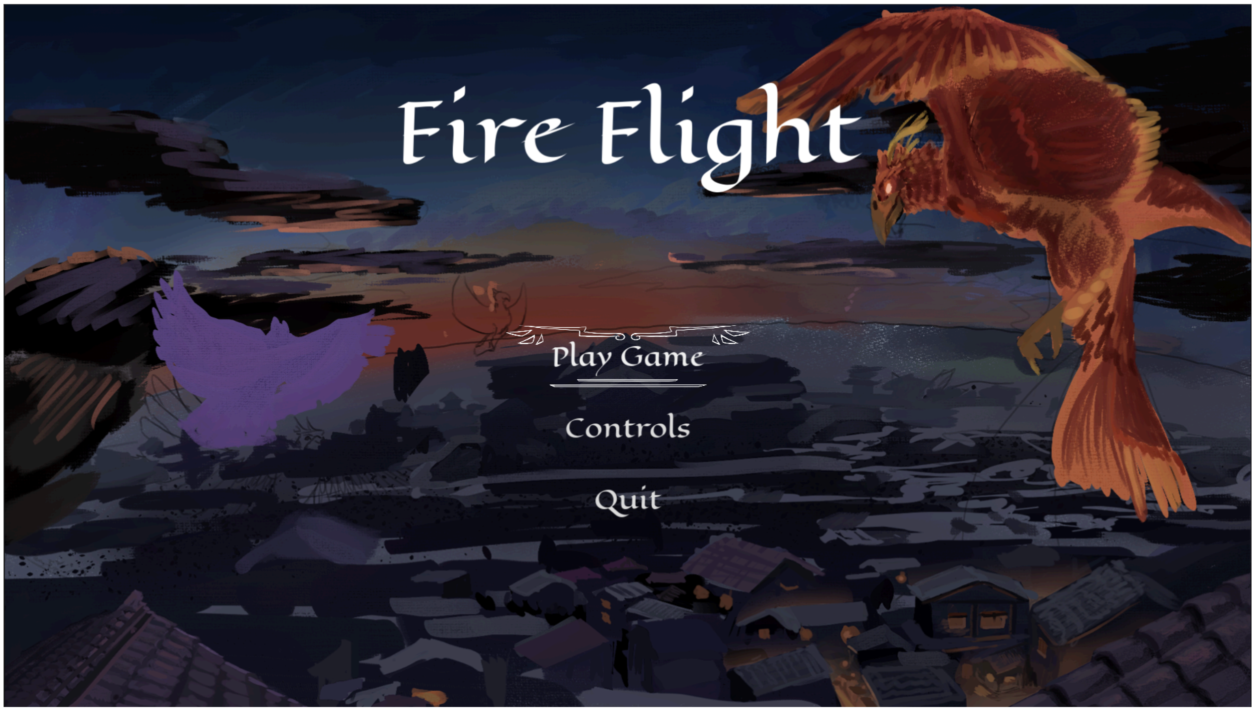
Competitive Edge

From the start, we wanted there to be a strong sense of competition in Fire Flight. Early data from playtesting even showed that the vast majority of players have a strong competitive urge when playing local multiplayer games with friends. With this in mind, we tried to balance the chaos of the game with the stability and consistency needed for highly competitive games. To accomplish this, we created the sense of chaos with overwhelming yet consistent attack patterns rather than randomness and maximized the sense of player agency whenever possible.

Before we got the loadout customization working, each player had a hard-coded set of attacks. The moment we added in that lobby customization, playtests took a distinct turn with participants getting much more absorbed by the strategy and gameplay. Allowing players to select their own weapons removes the innate advantages that some players had in earlier playtests. Likewise, as all spawn positions are functionally identical and power ups aren’t a privilege of any given player, the maps are completely fair as well. The only changing element, outside of player skill, that may give advantages to players is weapon choice. Through playtests, we have determined that some weapons are much easier to use and may even be more effective outright than others. Patterns like weave, which was fast moving and hard to predict, often damaged players much more than one like swirl, which was slower and much more predictable. Many of these were tweaked throughout development, from changing cooldowns, heat requirements, speed, or even the number of bullets in the patterns. It’s clear that there's still room for improvement in the balancing of patterns. This isn’t necessarily a bad thing. Notable is the fact that choosing patterns is a matter of player choice. Players may choose to have the same options in order to eliminate these issues or may find that having different weapons, even if unfair, may be more fun. In some scenarios, this may even act as a handicap for some player groups, allowing them to account for a difference in skill that will naturally be found between players.

Unique Aesthetic

The thematic concept behind Fire Flight is one we’re quite proud of, acting as a type of glue that helps tie the gameplay, medium, and external context together into a cohesive experience. The festival setting and fireworks fit with the intent of Fire Flight being a party game, which are often played when people are gathered for special events. The bright and colorful nature of fireworks and phoenixes sells the energy and fun we’re trying to capture while also making attacks and players clear and distinct from the background and each other. Lastly, fireworks and bullet hells are a perfect match, with both having flashy and intricate patterns of projectiles.



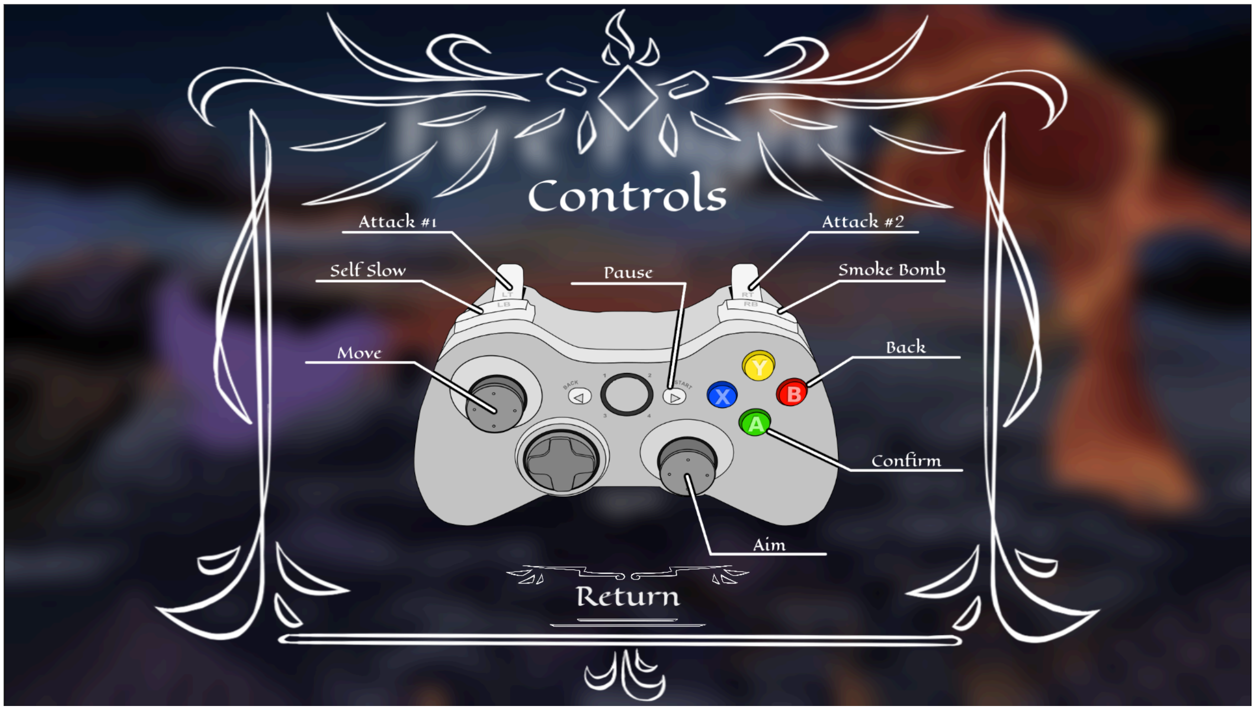
Additionally, Fire Flight’s aesthetic stands out when compared to other games in the local multiplayer medium. Most of the popular games that come to mind are styled around pixel art, simple 3d models, or simple flash-like 2d animation. This distinction could help us stand out in our market, though it could also make it harder to recognize as a local versus game due to not matching the general aesthetic used in those games.

Replayability

The game has a lot of opportunities to be replayable. First we have the 5 different maps which provide unique twists to the core gameplay forcing the players to move around or attack in unique ways to avoid hitting the clouds, or even use the clouds as cover to avoid incoming attacks. Furthermore through the lobby players are able to customize their character with 8 different colors and 7 different attack patterns. This allows for a lot of unique combinations to face off against your friends allowing unique opportunities each match. Finally the pick-up system allows for more replayability as they randomly spawn throughout the match making sure no two matches are the same.

Polish

As a team, we believe that a game’s polish is as strong as its weakest link. With this in mind, we were able to keep a solid standard of quality across the game to minimize the risk of anything looking out of place or unfinished. From the art side, this meant completing a breadth of visuals to minimize the use of placeholder assets, even if it meant leaving some visuals in a more rough or unpolished state. The dev side was similar, with the team making sure to fix anything that felt awkward or that playtests found to be buggy or otherwise incomplete. Much of our success in this area can be attributed to proper scope management and scheduling.



Of course, there will always be a weakest link regardless of how much effort goes into something, but we find it harder to pin down what that link is for Fire Flight. Even the main contenders for the spot are still solid enough that they don't break immersion or stick out too much. If we had to list some weaker areas of polish though, it would have to be the remaining placeholder assets in the controls screen and status indicator icons and/or the text popups when picking up a powerup.

Resources

Design

Game design for Fire Flight started out as a one-man show, though it quickly became evident that this wasn't sustainable. While one person still acted as the design lead and made the majority of the design decisions, especially on overarching systems, the members implementing any given feature were often given the task and discretion to make design decisions regarding what they were implementing. This was especially the case for the attack patterns, since the one programming the bullet behavior would have the best ideas for how best to utilize those development tools and levers.

Additionally, inspiration for maps, attacks, and other features were often drawn from other games in the genres we were working in, such as borrowing attack pattern ideas from *The Binding of Isaac* or the smoke bomb mechanic from *Touhou's* bombs.

Development (ALL minus Dan)

Godot was a new game engine for most people working on this project. Overall, learning was quick and adaptation happened within the first few weeks, thanks to the amount of tutorials online. The decision to use Godot's C# version did add a bit of difficulty in regard to finding documentation and finding code snippets online, but likely saved time that would have instead been spent learning GDScript. Additionally despite Godot being a relatively new game engine the developers that use it are very passionate about it leading to high quality tutorials.

Art & Sound

Out of the in-class members, we had one game designer and four programmers, one of which also acted as the sound designer. As such, it was of utmost importance for the team to recruit and onboard some artistic talent onto the team, as art and animation are resources our team would otherwise be sorely lacking in. To solve this, we presented the project at the Call for Game Artists event in MAGIC. After some networking at the event and reaching out to interested artists directly, we were able to recruit three dedicated artists to the project: two animators and an environmental illustrator. Having two animators was especially important since animation was the most in-demand resource for Fire Flight and, as non-class team members, each of the artists weren't able to work on the project full time. Having multiple animators helped ensure neither of them were overloaded with work while also providing great help to each other in the form of bouncing ideas between each other and checking each other's work.

For the sound designer, Music and Sound Effect creation needed to be balanced with programming obligations. This was possible by separating the audio work into manageable chunks to be worked on in parallel to programming. Sound selection and music composition took place at the same time allowing for cohesion between aspects of sound.

Project Assessment

Positives

- Most playtesters enjoyed the game
- Felt that it was a good mix of competition and chaos
- Core mechanics of the game were easy to learn and adapt too
- Strong polish for most of the game
- More-or-less a complete package. If it weren't for the few missing/incomplete assets, this game could be ready to ship.
- Accomplished all major project goals

Negatives

- Trouble conveying information such as controls and game mechanic functionality, and hitboxes to players
- Difficulty figuring out a good/satisfying UI layout that wouldn't distract or take players out of the game to view
- Small amount of difficulty in providing feedback for events, such as being damaged or cooldowns
- The attacks in the game are all pure damage with no utility other than the implicit zoning control that having an active hitbox on screen creates. This can lead to attacks on the whole feeling "samey" despite each one having different properties and uses.

Success

Overall, we view Fire Flight as a great success in its current state given the 15 week timeline. The game is playable, fun, stable, and very presentable. As a team, we have strong hopes to continue this project into the future, expanding on our current systems and adding more unique attacks and abilities. However, even if we have to leave Fire Flight in its current state, it's a project we can all be proud of and showcase to both friends and employers.

Pitfalls

- We got some really good feedback about changes to the attacks or additional attack types in the last few weeks where it was too late to make any large changes
- Missing or incomplete assets drag down the sense of polish
- Underestimated the time it takes to make player animations as opposed to simpler effects animations
- Misjudged lengths of tasks, particularly in milestone 2
- Lack of solid Godot documentation led to lots of headaches
- Difficult to convey the more intricate game mechanics in a non-intrusive way

Team Assessment

Workflow

At the start of each milestone, we'd review the previous milestone's progress and determine what tasks we aimed to accomplish in the upcoming milestone. Then, in our Sunday meeting, we'd assign tasks and due dates for the sprint and update the trello timeline appropriately. Outside of class, we had two weekly meetings. Because of the size of our eight-person team, we didn't require every person to show up to every meeting since that'd be nearly impossible to schedule. Sometimes we'd do some work outside of meetings as well and check in over discord. Essentially, our workflow was very flexible, but with expected goals to meet by certain deadlines. Even those deadlines could be flexible though, since we had scheduled buffer periods into our progress timeline to account for any blockers or unexpected issues.

Once a task was marked as done, the project lead would review the changes, test them, and either request further changes or send the trello card into the “completed” list. For changes the project lead made, he would get someone else on the team to double check his work or, for minor changes, simply see if anyone raised issues before marking the trello card as completed.

Positives

- Organization of Trello led people to be working on tasks and plan for future tasks without difficulty
- Artists joined project, allowing less time searching for assets and more time for programmers to focus on programming the game
- Team members helped each other solve problems and develop solutions

Negatives

- We needed to communicate better, occasionally had people working on the same thing without realizing it which caused merge conflicts/loss of work
- Could be better at commenting on code while working on it, instead of going back afterwards
- Code often wasn’t programmed with future changes taken into consideration, causing some things to need to be completely rewritten
    - Particularly an issue with ‘if’ statement parameters
- Misunderstandings with Godot’s git integration led to unnecessary information in commits midway through project, making it harder to keep track of changes

Lessons Learned

- Godot is a really useful game engine with a lot of unique features.
- Autoloads are very useful (Godot's Singletons)
- Planning in advance to have extra time dedicated to working on feedback and bug fixing was really helpful.
- Task delegation is very important. Making sure no one person has too much or too little to do is critical, and each side of that coin is just as bad as the other.
- Dedicating more time to developer tools early on would have saved us time in the end (such as being able to spawn specific power-ups)
- Being more careful with debug statements in order to not lag others games