Fire Flight

By TinyDuck

The Team

- Patrick Doherty - Programmer
- Zander Herbert - Programmer
- Connor Higbee - Programming Lead, behavior programmer, sound engineer
- Dan Kowalski - Project manager, game designer, programmer
- Donovan Scullion - Programmer

Out-of Class Members

- Alex Ni - Environmental Artist
- Gia Romero - 2D Animator
- Sophia Wyse - 2D Animator

Trello Link

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?

Summary

Fire Flight will aim to create a chaotic top-down battlefield of sparks and smoke. Players control their phoenix, choosing a color to distinguish themselves, move around with one joystick, and shoot out bullet patterns in the form of firework displays with the other joystick, firing one of two types of patterns using the left and right triggers. Depending on scope, the game would also feature different firework patterns, either as random pickups mid-match or as a quick loadout option in the game's lobby.

Inspiration

Some of the most fun times Dan's had in his life have been when sitting with friends with a hodge-podge of different controller types plugged into a laptop and connected to the TV, playing all sorts of local co-op and competitive games. While most of these party games are casual in nature, the ones that stuck with him the most were those that still had a hint of a competitive edge, where layers of depth and mastery are hidden behind a deceptively simple first impression. Games like TowerFall, Spider Heck, Screencheat, and so on. Fire Flight aims to capture this essence, a game that's easy to get started in mid-party almost instantly while providing room for skill expression with flashy and impressive plays.

The other half of the puzzle comes from bullet hells, especially Touhou. Some of the games in the Touhou series are made of PvP fights that work in a similar way to Tetris99, where defeating enemies will send hazards to your opponent and vice versa. The concept of a PvP bullet hell stuck in Dan's mind while playing Astral Gun, a top down PvP twin-sticks shooter. However, unlike a bullet hell, the focus of Astral Gun is on aiming rather than dodging, and the attacks are relatively tame. In contrast to this, Fire Flight aims to capture the essence of bullet hells, littering the screen with projectiles. The end goal is to make a local versus game where each player feels like they themselves are a bullet hell boss.

Moodboard

https://docs.google.com/document/d/1vfFKAZ6kLhYQmhmWBy_RgyPIiw8ZjYiK1Il46J3Dg4g/edit?usp=sharing

Mechanics

Controls

The game is intended to be played with a twin-stick style controller. The left stick is used to move the player around the map, while the right stick allows the player to aim their attacks. Players may hold down one of the bumpers (left vs right is to be determined) to reduce their movement speed, allowing for more precise movement to avoid incoming projectiles. Each player's aim will be indicated with either a small arrow or chevron to indicate direction, or by rotating the player sprite in the direction they will fire. If the stick is neutral it will use the most recent aimed direction.

To keep the controls simple for players to learn they will only have 2 different firework types or patterns to use during a match, these will be chosen before the match starts and will use the Left and Right Triggers respectively. Launching your fireworks utilizes a "heat" meter, so players will need to manage their attacks instead of endlessly firing. Another use of this resource is the "smoke bomb" ability. This is tied to one of the bumpers and uses a large amount of your heat to create a burst of smoke, instantly clearing all projectiles in a radius around you, similar to bombs in other bullet hell games.

Movement

- Controlled by left stick
- Velocity is normalized
- Instant acceleration/stop
- No drifting movement. Ice physics in a bullet hell would be awful
- Movement direction is independent of aiming direction
- "Slow down" button drastically reduces movement speed while held down. Start with making it half speed, then change it based on testing.
    - Bound to left bumper

- Can’t move through clouds/walls. Holding diagonally against a wall should still keep the velocity that is parallel to the wall (i.e. should be able to “slide” against walls rather than just being stuck)

Aiming

- Controlled by right stick
- Phoenix visibly faces the direction it’s aiming
- Aiming is independent of movement
- When there is no aiming input, retain the most recent aim direction
- Aiming direction has a turning speed, so it takes time to change direction
    - Rotation speed is fixed. Instant acceleration/stop
- Attacks fired are centered on the aiming direction

Attacks (Fireworks)

- Each player will have two attacks, mapped to the two triggers
- Attacks will originate from the player and be fired out from them, with any patterns centered on the player’s current aim direction
- Attacks will fire and drain the player’s heat continuously for as long as they hold down the attack button. While doing this, the player’s flight and turn speeds are brought to a crawl, such that VERY minor adjustments can be made.
- Attacks will have a short built-in cooldown in order to prevent mashing.
- Attacks will consume an amount of the player’s heat meter based on the type of attack. This will be one of the balancing “levers” for us to work with in the future.
    - There is both an initial cost (to encourage committing to an attack) and a prolonged cost that drains the player’s heat as long as they channel the attack.
- Bullet objects (sparks) will be controlled algorithmically, applying movement values by the bullet pattern it’s a part of

Smoke Bomb

- Spends half of the player's maximum heat meter to delete all enemy sparks within 2 player-lengths of them
- Bound to right bumper
- Activates on button pressed, and not on held/released
- Add a very brief cooldown period to prevent accidental double-inputs in the event a player is mashing in a panic

Resource Meter (Heat)

- Internally tracked as an float from 0 to 100
    - Displayed as numberless meter
- Each player has one independent of all other players
- Consumed by fireworks and smoke bomb
- Refilled passively over time (10 per second)
- Refilled when a player hits an opponent with a spark (15 per hit)
- Refilled when grazing enemy projectiles (5 per spark grazed)
    - Graze hitbox will encompass the player's entire sprite while the damage-taking hitbox will encompass only the player's core/center
    - Cannot graze on the same projectile multiple times
    - Grazing does not delete the spark, but being hit does
    - Grazing is checked when a spark leaves the player's graze hitbox
    - Heat regained from grazing is reduced while attacking to help prevent endless attacking
- When trying to use a firework or smoke bomb without enough meter to use it, the heat meter will visibly flash black and the player's controller will vibrate

Objective

The way that a player wins is by being the last player left alive, all players will start with the same number of "lives" or times they can be hit, after each hit the damaged player will "rekindle from ashes" giving them a brief invulnerable period at the

cost of not being able to attack. To help players from falling too far behind from repeatedly dying, they will gain a damaging flame trail that follows them, starting after their first death and increasing in length for each additional death.

Lives

- Each player starts with 3 lives
- When a player is hit, they lose a life
- When a player is at 0 lives, they are eliminated
    - Projectiles that player created stay on the screen until they disappear naturally
- When a player loses a life, they enter a temporary invulnerability state where they cannot take any non-movement or non-aiming actions and cannot be hit
- While invulnerable, the player will be dimmed and flashing darker

Fire Trail

- After a player's first post-damage invulnerability period, they start leaving a fire trail that increases in length/duration based on how many lives they've lost this game
- Fire trail damages opponents who fly into it

Environment

The other non-player centric mechanics in Fire Flight include a variety of systems to enhance gameplay. The first is the Map, the main area where players will be able to fight each other. All players are on the same screen at the same time rather than in a split-screen environment. The walls of the map will be visually represented as clouds, limiting player movement and forcing them to fight each other while also providing a source of cover, as projectiles can't go through walls.

The final group of mechanics are random pickups, which will randomly spawn throughout the map and give players a powerful, yet temporary, buff. These vary from simple boosts to the main mechanics such as a heat refill, or a barrier blocking the next

projectile to hit the player, to changing how their attack works with a triple shot, or firing an additional projectile backwards. Items may also cause an instantaneous effect, such as making a delayed lighting strike on top of each opponent.

Map

- There will be multiple battle arenas to choose from after players select their phoenix color and firework/attack types
- Maps in general should be symmetrical on at least one axis
- Maps are differentiated by their wall layouts and possibly by background images

Walls (Clouds)

- Clouds that block projectiles
- Walls on some maps may be able to move in a set path/pattern
- Players can fly into clouds (and clouds into players). While inside, players move at half speed and lose 20 heat per second (making it a net of -10 per second when combined with the passive regeneration).
- If a player is in a cloud with 0 heat remaining, they lose a life (entering the invulnerability period as per usual). While invulnerable, clouds do not slow players.
- The dark clouds at the edge of the screen are impassable by any means.
- After 10 minutes (or a number of minutes chosen by the players in the settings), clouds start creeping in from the edges of the screen, effectively shrinking the playable area until covering almost the entire screen, forcing the game to end shortly after due to knocking out all of the players.

Pickups (Red Envelopes)

- Items that spawn randomly on the map.
- Flying into an item picks it up and immediately applies its effect, showing an icon above the player briefly to indicate what the item was.
- The game will spawn items one at a time at random intervals between 15-45 seconds.

- If possible, items will not spawn inside of clouds or direction on/next to players
- Pickup Effects:
    - Heat refill
    - Barrier (blocks 1 hit)
    - Orbiting ring of sparks around the player for X seconds
    - Projectile speed up for the next X seconds
    - Fireworks display (fires single-ring firework patterns at random points (using the same algorithm as the pickup spawning) for the next X seconds)
    - Mobile attacking (based on hold-to-shoot style; allows for full movement speed while attacking)
    - Slow all other players for X seconds
    - Instantly apply a smoke bomb effect for the player without spending heat
    - Camo bullets (lowers the opacity of your sparks for the next X seconds)

Game Structure/Ideal Play Session

Here, we will imagine a scenario where a group of friends decide to play Fire Flight. To set the scene, a group of four high school friends are hanging out at one of their houses and are going through some party games they have, playing each for a few rounds until they decide to switch to a new game. They had just finished up a split-screen co-op game and have decided to play something a bit more competitive to switch things up.

Launching the game, Fire Flight loads in rather quickly due to its small size. ON the main menu, the host double checks the settings screen to make sure the audio is ok. From there, they press the “Start” button and are brought to a screen where everyone can press A (or X on playstation controllers) to join. Once they do so, each player scrolls through the color options for their phoenix, all named after the chemical elements that produce those colors in fireworks. Below that, they scroll through their firework (weapon) options, skimming through the short descriptions and occasionally checking the preview GIFs for any that catch their attention. Once everyone’s readied

up, the host choses the arena to play on, going for the basic empty one since two of their friends haven't played this before.

As the match starts, the experienced players decide to hold back to give the newbies a chance to get used to the controls. After a few seconds of flying around and accidentally shooting one of the people letting them practice, the new players give the OK for everyone to properly start the match. Despite that, some of the players still play a little safe as they experiment with their attacks and movement. Shortly after, the players start moving more aggressively and start to try and directly attack each other now that they're more comfortable with the tools at their disposal. It's also around this point that the first random power-ups spawn, a mini-objective that they fight over and sometimes die for. As they start getting low on lives or even eliminated, they start seeing that player two still has all three lives left! Two of the other players start to try and focus-fire on player two, while the last player starts hiding out in a corner to preserve their last life. Others go out in a comedically played-up blaze of glory, charging into point blank range only to get immediately burned out of the sky. Eventually, the clouds at the edges of the screen start closing in, knocking out anyone who gets caught in them and forcing the remaining two players to wrap up the game by tightening up the playspace.

When one of the new players emerges as a victor, they're all brought to a results screen where they can look at stats such as damage dealt/taken, amount of time survived, items picked up, etc. Perhaps they make some jokes or comments on how one of them hogged up all of the item drops or survived for twice as long as someone else. Regardless, they proceed and return to the color and firework selection screen, taking the opportunity to try out some new combinations based on the last match. This cycle repeats for around 20-45 minutes until they decide to move on to another game for the night.

Systems

Health/Life/Death

When a player gets hit by an enemy projectile, they lose a life. If they have any lives remaining, they enter an invulnerable state for a brief time window where they can move but not shoot as they “rekindle from ashes” After their first death, players leave a trail of damaging fire that is longer for each time they’ve died

This primarily interacts with the combat and bullet pattern systems

Combat

Movement

Basic action the player can take and likely will almost always be doing. Allows for dodging, the collection of pickups, and strategic repositioning on the screen.

Bullet Patterns

Players are able to use energy to fire bullet patterns (fireworks). These determine areas of danger on the screen and are the primary means of interaction between players. This system can be affected by pickups players may have and the choices each player makes when picking attack types in the pre-game lobby.

Interacts with the health system and energy system.

Pickups Spawner

At random points of the game, pickups can spawn on screen that players can move over to collect. Pickups will change bullet patterns players have access to, change the properties of their phoenix, or have other various effects. This system is made to add an aspect of luck into the game, which creates variety and can help make for interesting gameplay decisions.

Heat Resource

Acts as a gauge of when and how many actions players can take. More specifically, using fireworks and the smoke bomb ability spends this resource. Hitting attacks and narrowly dodging (grazing) enemy projectiles refills the heat meter in addition to a slower passive regeneration.

Interacts with the combat system.

Player Customization

Allows players to decide which weapons and abilities they can take into combat as well as their visual appearance in the form of their phoenix's color. This provides strategic depth by letting players mix and match different bullet patterns to try and find synergies that match their playstyle. The color customization also helps with gameplay clarity, as it lets everyone choose a color that they can associate with, meaning they can know which projectiles are theirs and (more importantly) which are *not* theirs.

Interacts with combat systems.

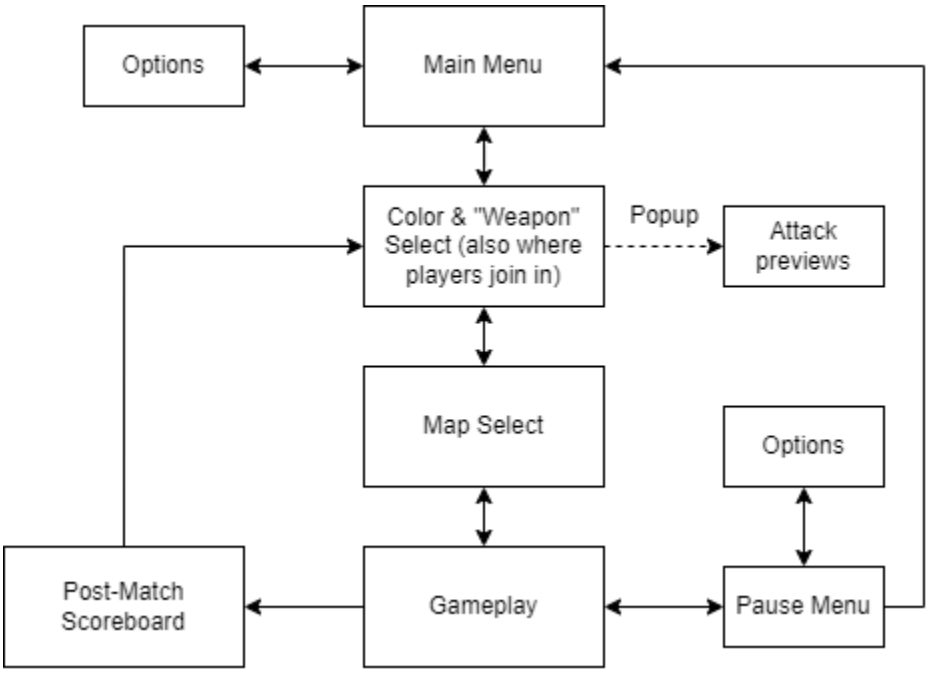
Multiplayer support

Handles tracking the actions and separation of players, assigning each connected controller to a player object in the game. Interacts with essentially all systems.

Menu

Allows players to quit/restart/start games and change the number of players. Additionally provides access to game settings, the game's controls, and player customization options. Interacts with all systems.

User Interface



UI flowchart

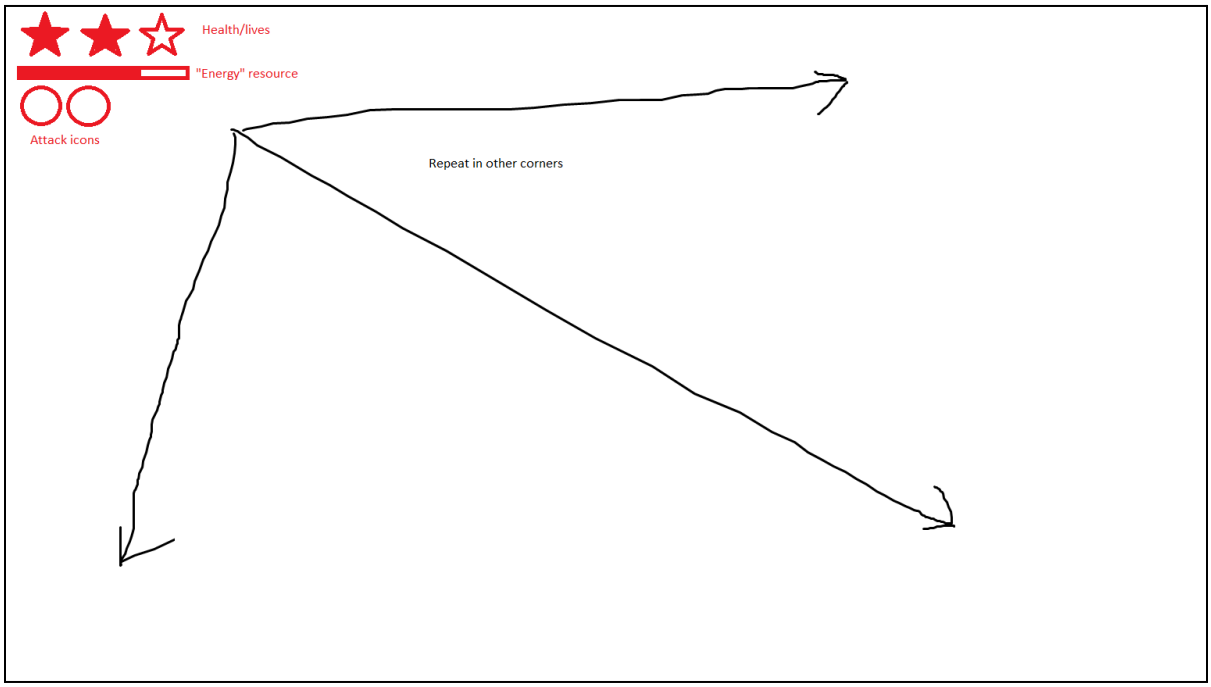
HUD

The goal of the HUD design is to provide all the information needed for moment-to-moment decision making in a match without overloading the player. The HUD elements will need to be visually distinct from other things in the match, namely incoming and outgoing projectiles. Ideally, the most important information a player will need should be near their character at all times, since players are likely to focus on watching their character and their immediate surroundings, as is typical in bullet hells.

There are two core HUD elements: the life counters and heat meters. Of these, players will need to know the state of their own heat meter at all times, as it is effectively their ammunition *and* fuel for their panic button (smoke bomb). This meter is most likely to be placed directly next to each player character. The life counter is important for all players to see for strategic and tactical purposes, as it shows who is close to being eliminated (including yourself) and who is in the lead. However, such information isn't critical for the twitch element of gameplay. As such, this UI element should be placed

near the edge of the screen, either with each player's life counter in the corner of the screen they started in, or all at the bottom of the screen together.

The following images show some rough sketches of different HUD layouts. It is likely we will pull individual elements from multiple designs and put them together.



Themes/Aesthetics [TODO]

We look to draw primarily from East Asian traditions and festivals, such as the celebration of Lunar New Year, lantern festivals, and so on. To match this, the art style should match this traditional feel while still carrying the energy that comes with festivals and fireworks. A type of stylized 2d papercraft aesthetic highlighted with modern lighting effects would fit this game perfectly. The papercraft evokes the style of traditional East Asian paintings, origami, and parade decorations, while the modern lighting and visual

effects emphasize the energy and warmth of the events the game draws inspiration from.

Each aspect of this style also helps with gameplay clarity, as items and characters of importance can be highlighted with the VFX, and game objects can be easily differentiated by the clear lineart or borders from the traditional painting style.

Resources

Development

- Godot
- Github Repository
- Visual Studio/VS Code
- PC-compatible controllers
- Sub-disciplines
    - Player control
    - Projectile behavior
    - Collision
    - UI/UX
    - Sound and animation implementation
    - Scene management

Organization

- Trello
- Google drive
    - Docs
    - Forms

Sound

- Edison

- FL Studio
- Audacity
- Required sound
    - Menu interactions
    - Background music (menu, mid-match, results screen)
    - Attack sfx
    - Damage sfx
    - Rebirth sfx
    - Item pickup/use sfx
    - Smoke bomb sfx
    - Victory fanfare

Art

- [image/animation program(s)]
- 2D animations
    - Hover
    - Moving flight
    - Turning
    - Attacking
    - Death
    - Clouds
    - Firework projectiles
- VFX
    - Flame particles
    - Firework particles and emissions
    - Death effects
    - Smoke
    - Clouds

UI/UX

- GIMP 2.0 (or other image editing software)
- Menu concepting
- UI graphic design
- UX testing
- UI programming
- Onboarding

QA/Playtesting

- Google forms
- Internal playtesting/QA
- Playtest design
    - Deciding on what to test for, what questions to ask, etc
- Data analysis

Progress Timeline

Week 3:

- Github repo and Godot project set up
- Intro to Godot
- Create nodes for players, inputs, etc.
- Controller input
- Placeholder player sprite
- Player movement first draft
- Codify mechanics
- (If artist(s) on board) Moodboard

Week 4:

- Player can aim and shoot
- Bullet and bullet pattern code architecture started
- Multiple controllers and multiple players

- Bullet collision
- UI mockup
- (If artist(s) on board) Phoenix sketches

Week 5 (Milestone 1): 2/16

- Attacks have a small cooldown to avoid being mashable
- Heat meter spent to use attacks. Refills over time and when you hit someone.
- Invulnerability period when hit + some sort of visual indicator (gray out the sprite)
- Lives/health and energy in ui (likely just as text)
- Game stops and/or restarts when a player runs out of lives
- Burst mechanic
- (If artist(s) on board) Projectile sketches
- Milestone writeup and presentation

Week 6:

- Review milestone 1 progress
- Bullet pattern design
- Fixing bugs from milestone 1
- Code overview and documentation
- Foundations of menu and scene management code
- Grazing to generate energy
- Arena shrinks after some amount of time to force the game to end

Week 7 (MIDTERMS):

- Main menu functional
- Start on color/firework selection menu
- Implement one of the new bullet patterns
- Spawn random items on the map during the fight that give whoever collects them energy
- Design 1-3 arenas
- Send out playtests

Week 8:

- Review playtests and adjust accordingly
- Finish color/firework selection menu

- Implement a new bullet pattern
- Implement other item pickup effects
- Implement arenas designed in week 7

Week 9 (SPRING BREAK):

- Send out playtests
- Fix any emergency bugs that show up

Week 10 (Milestone 2): 3/22

- End of match scoreboard
- Attack preview gifs in firework selection screen
- Implement a new bullet pattern
- Implement final item pickup effects
- Soft cutoff for any new features
- Milestone writeup and presentation
- (If artist(s) on board) Should have sketched animations for most or all basic player actions done by this point (hover, moving, turning, dying, attacking)

Week 11:

- Review milestone 2
- Adjustments based on milestone 2 playtesting
- Pause menu during a match
- Hype announcements (ex. “triple kill!” or “X kill streak”)
- Begin work on options menu on the main menu
- Start searching for and implementing assets we won’t get from artists (if any) in time

Week 12:

- Clean up backlog
- Finish options menu and functionality
- Camera adjusts based on distance between players
- Hard cutoff for any new features

Week 13:

- Playtests
- Bug fixes

- Visual improvements
- Balance adjustments
- Control/movement adjustments

Week 14 (FINALS):

- Repeat of week 13
- (If artist(s) on board) Visuals for any segments we have artists for should be complete by this point at the latest

Week 15 (Milestone 3): 4/26

- Final quality of life tweaks
- Bug fixes
- Creating a stable final build
- Post mortem and presentation

Stretch:

- Alternate game mode (King of the hill? Teams? PvE bullet hell survival?)
- Extra attack types
- More arenas
- Match options (stock vs time, items vs no items, other modifiers, etc)
- Custom color picker