

Design Document Outline

Group Members: Krystal Kim & Monique Legaspi

Game Title: FLY, ICARUS

1. EXECUTIVE SUMMARY

Fly, Icarus is an exciting Greek-mythology based 2D side scroller game featuring a novel scroll-to-fly game mechanic and a unique, ancient Greek-pottery-inspired art style reminiscent of vintage monochrome games. The player must control Icarus, with wax wings adorned, and fly across the ocean to the next island, dodging both dangerous sea monsters and the heat of the blazing sun, lest he falls into the water and perishes.

Resource management is an important aspect of *Fly, Icarus*. Icarus's difficulty of flight is directly proportional to how many items he is carrying. This means that the more items are in the inventory, the more frequently/forcefully the player will have to scroll to keep Icarus aloft. The player will have to balance the cost/benefit of collecting power-ups, using them at opportune moments, and dropping them when they have too many.

Due to the simple mechanics of the game, *Fly, Icarus* is easy to pick up and allows the player to become more invested in the world of the game, absorbing the stunning visuals and compelling story as they move Icarus along his journey across the ocean. The real challenge comes from monitoring dual status bars, each measuring Icarus's own health and the amount of wax left on his mythic wings, and balancing this vigilance against the strategic collection, use, and discarding of the many power-ups Icarus will encounter as he flies.

This game will be developed only for the PC. We expect the entire development process to take about 5 weeks with a team of 2 game developers.

2. GAME DESIGN - CREATIVE

2.1 High Concept

It's a race against the elements. The player must control Icarus, with wax wings adorned, and fly across the ocean to the next island, dodging both dangerous sea monsters and the heat of the blazing sun, lest he falls into the water and perishes.

2.2 Design Goals

2.2.1 Main Design Features

2.2.1.1 Player goals and objectives.

Main Goal: Fly across the ocean in pursuit of arriving at the next island, reaching as many islands as possible. (In our prototype, there will only be one level, so the player's goal here is simply to reach the next island, thus finishing the game.)

Secondary Goal: While flying to the next island, try to gain, and thus finish the level with, as many points as possible. Contributing factors include:

- Maintaining Icarus's wings as he flies so that they are not melted. This can be achieved by not flying close to the sun for long periods of time. (higher amount of wax left upon reaching the next island = more points)
- Maintaining Icarus's health points (HP) so he reaches the next island in good health. This can be achieved by not flying too close to the ocean and being injured by sea monsters that pop up periodically. (higher amount of HP left = more points)
- Collecting power-ups that help Icarus fly safely, but not so many that he becomes too heavy to fly. (depending on power-up, will help Icarus conserve HP/wax and thus earn more points)

Challenges:

- Maintaining an optimal flight elevation for Icarus, which may change depending on where he is in the level. The player must adjust scroll speed and frequency according to which dangers are most imminent.
- The sun, which is always present, melts the wax in Icarus's wings if he stays close to it for too long.
- Sea monsters, which will pop up out of the ocean periodically along Icarus's journey, dealing him HP damage if their paths intersect. Each monster's reach(height) and damage dealt depends on its species.
- Birds, which appear periodically and fly horizontally in the opposite direction to Icarus. These birds are not malicious, but intercepting them will decrease the wax in Icarus's wings.
- Icarus's constant forward motion. The player must plan out his flight path in order to intersect it with oncoming power-ups, or else they will be missed and left behind forever.

Conflicts:

- Icarus's wings are made of a finite amount of wax. Depletion of this wax means that Icarus's wings have completely melted away, rendering him unable to fly even if he still has HP left. This causes Icarus to fall into the ocean and drown, ending the level with a loss state.
- Icarus's finite amount of HP. Depletion of this HP means that Icarus has lost all strength to fly and is unable to continue, even if his wings still have wax. This causes Icarus to fall into the ocean and drown, ending the level with a loss state.
- Prolonged exposure to the sun will melt Icarus's wings and deplete his wax, so the player must not fly Icarus too high for a long time.
- Being hit by sea monsters will harm Icarus and decrease his HP, so the player must not fly Icarus too low to the ocean (or at least be mindful of monsters when they may pop up).
- Being hit by birds will take chunks out of Icarus's wings and deplete his wax, so the player must take care to avoid them while flying.
- Simply falling into the ocean. Even if Icarus has wax and HP left, if the player stops scrolling, Icarus will stop flying, start falling and drown immediately upon touching the ocean, ending the level with a loss state.
- Strategic interaction with power-ups. Icarus's difficulty of flight is directly proportional to how many items he is carrying. This means that the more items are in the inventory, the more frequently/forcefully the player will have to scroll to keep Icarus aloft. The player will have to balance the cost/benefit of collecting power-ups, using them at opportune moments, and dropping them when they have too many. (There are also a limited number of slots in the inventory, and the player will not be able to collect more power-ups if the inventory is full, which requires more strategy in what to use/eject.)

Win condition: The player wins the game by completing all levels—i.e., flying across each ocean without drowning and reaching every island until the last one. In the full version of this game, there will be multiple islands, but our prototype will have just one.

2.2.1.2 *Main rules and procedures*

Operational rules:

- The player may control Icarus's flight in the vertical direction.
- The player may collect power-ups, use them one at a time, and drop them at any point during the level.
- Flying past a certain altitude will denote that Icarus is too close to the sun and start depleting his wax.
- Intercepting a sea monster will deplete Icarus's HP.
- Intercepting a bird will deplete Icarus's wax.

Main game mechanics:

- Each level begins with Icarus standing at the edge of the previous level's goal island. Gameplay does not start until the player first scrolls upward, launching Icarus into the air.
- The player controls Icarus's vertical motion by scrolling upward with their mouse/trackpad to propel him higher. Ceasing to scroll will cause Icarus to glide

- for a moment, then begin falling towards the ocean. Scrolling downward will increase Icarus's falling speed.
- The player may collect power-ups by flying Icarus into them, at which point the power-up will appear in Icarus's inventory.
 - Each slot in the inventory is mapped to a number key on the player's keyboard. To use a power-up, the player may press the number key associated with that power-up's slot. Each power-up is single-use and lasts for five seconds (unless it's a status-restoring power-up). The player may not use more than one power-up at a time.
 - To eject a power-up from the inventory, the player may press the tilde (~) key. This will eject the least-recently acquired power-up.
 - *On a second by second basis*: The player must always be mindful of Icarus's position in relation to both the sun and the ocean. Deft control of scrolling (and not scrolling) is essential.
 - *On a minute by minute basis*: Continuous awareness/monitoring of how inventory stock is affecting Icarus's ease of flight helps the player maximize the effectiveness of their scrolls. The player must also think long-term and strategize when is the best time to collect, use, and eject power-ups.
 - *What makes this fun or interesting?*: The combination of strategy and constant movement keeps the player fully engaged, both mentally and physically, encouraging more meaningful play and immersion into the world of the game.

2.2.1.3 Player Resources

- *Wax*: Quantifies the amount of wax left on Icarus's wings, allowing him to continue flying. Displayed as a status bar at the top of the screen.
- *Health Points (HP)*: Quantifies the amount of strength Icarus has left, allowing him to continue flying. Displayed as a status bar at the top of the screen.
- *Power-ups*: Single-use items that allow Icarus to perform actions that would otherwise not be possible within the limits of the game. Collectible by intersecting Icarus with them, and used by selecting them from the inventory. All power-up effects are temporary (except for those that restore status points), lasting only a few seconds.
 - *Palm Leaf*: Shields Icarus from the sun, allowing him to fly higher without losing wax.
 - *Monster Repellent*: Prevents monsters from coming out of the ocean, allowing Icarus to fly lower without being attacked.
 - *Gust of Wind*: Propels Icarus forward, increasing his horizontal speed.
 - *Feather*: Allows Icarus to glide horizontally for longer between scrolls.
 - *Ambrosia*: Restores HP to Icarus.
 - *Wax Wad*: Restores Wax to Icarus's wings.

2.2.1.4 Boundaries and Constraints

Boundaries: The player is bound by the borders of the screen, which constitute the world in which Icarus may fly. Icarus is also always flying forward (horizontally to the right), so the player is bound by this motion. Since this is an isolated, offline, single-player computer game, the player may not introduce any outside elements to the game (and is likely unable to without hacking).

Constraints: The player's in-game interaction is constrained by their keyboard and mouse. They may only scroll up or down (or stop scrolling) to control Icarus's vertical flight. They only use the number keys (and ~) to interact with the items in the inventory.

2.2.2 Appeal

What is the game genre?: Greek mythology side-scroller

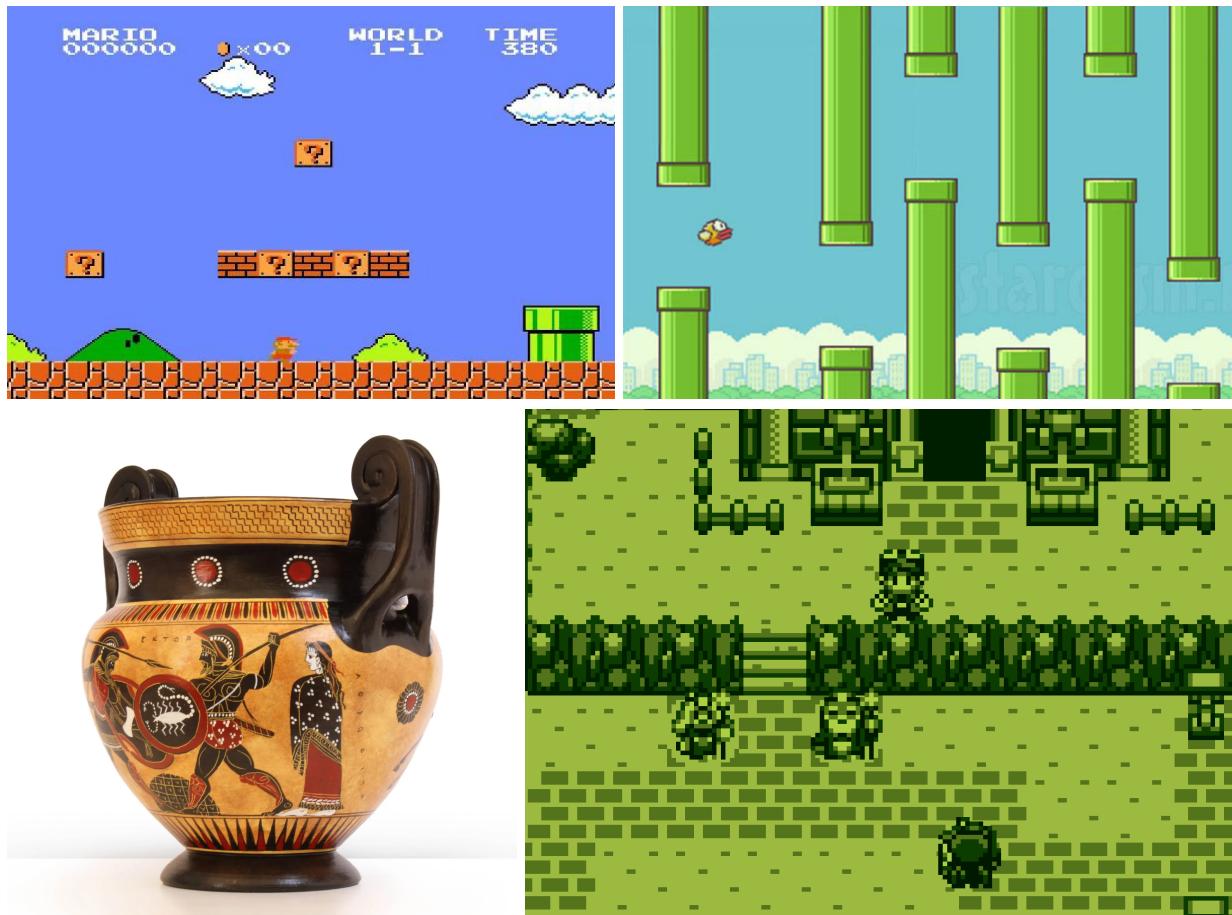
Who is the target audience?: People who enjoy Greek mythology and classic 2D side-scrollers

What is their age?: Tweens, teens, and new adults

Why is the game fun to play?: Simple game mechanics, compelling art, and story

2.2.3 Look and Feel

The game will have the general feel and functionality of classic side-scrolling games such as Super Mario and Flappy Bird. Its aesthetics (color scheme and art style) will derive inspiration from ancient Greek pottery, paying homage to its source material, as well as vintage monochrome video games. See below.

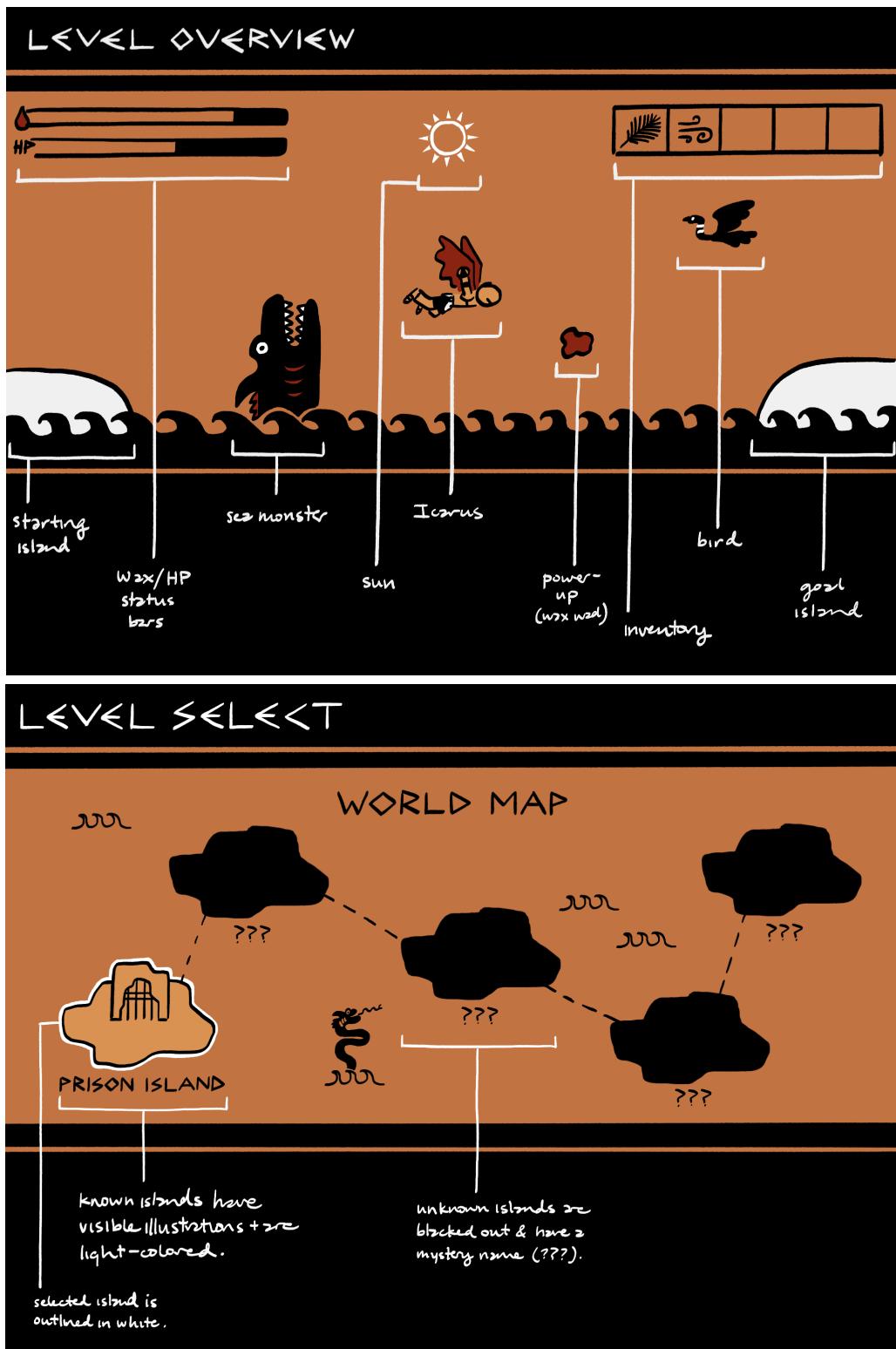


2.3 Worlds, Characters and Story

2.3.1 Back Story

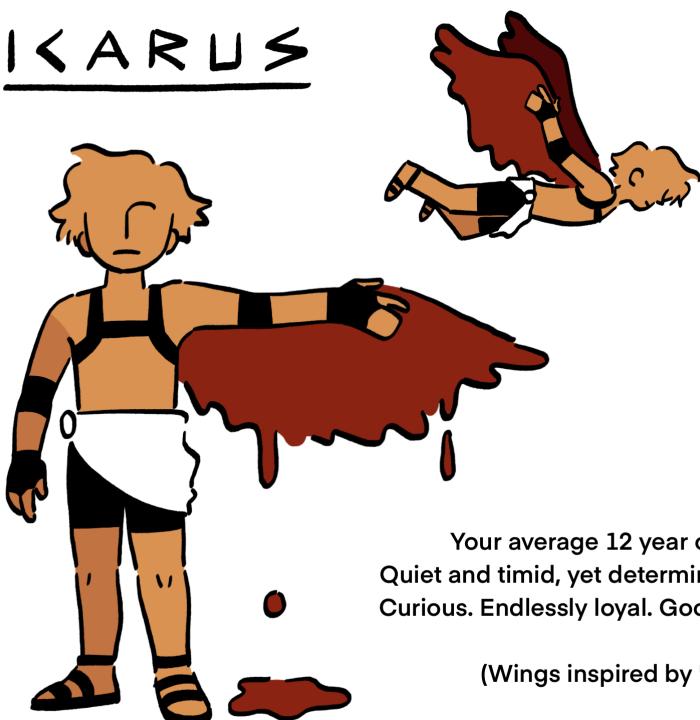
Prior to the beginning of the game, Icarus and his father, Daedalus, are trapped on Prison Island by one of Daedalus's enemies. Daedalus has fashioned a single pair of wax wings, which he gives to Icarus, urging him to use these wings to escape from Prison Island and find someone who will set Daedalus free. Icarus's in-game journey details his own escape and search for the key to his father's freedom.

2.3.2 Spaces/Worlds



2.3.3 Characters

IKARUS



Your average 12 year old boy with wax wings.
Quiet and timid, yet determined to get the job done.
Curious. Endlessly loyal. Good at listening to orders.

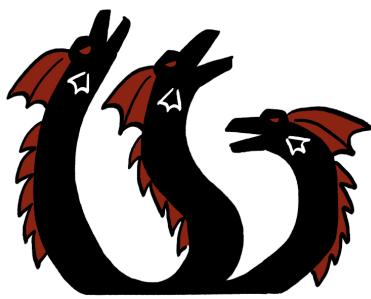
(Wings inspired by "Autobiography of Red"
by Anne Carson.)

SEA MONSTERS



CETUS

KRAKEN



HYDRA



CHARYBDIS



2.3.4 Levels of Difficulty

As the player progresses through the levels, difficulty will increase in two ways:

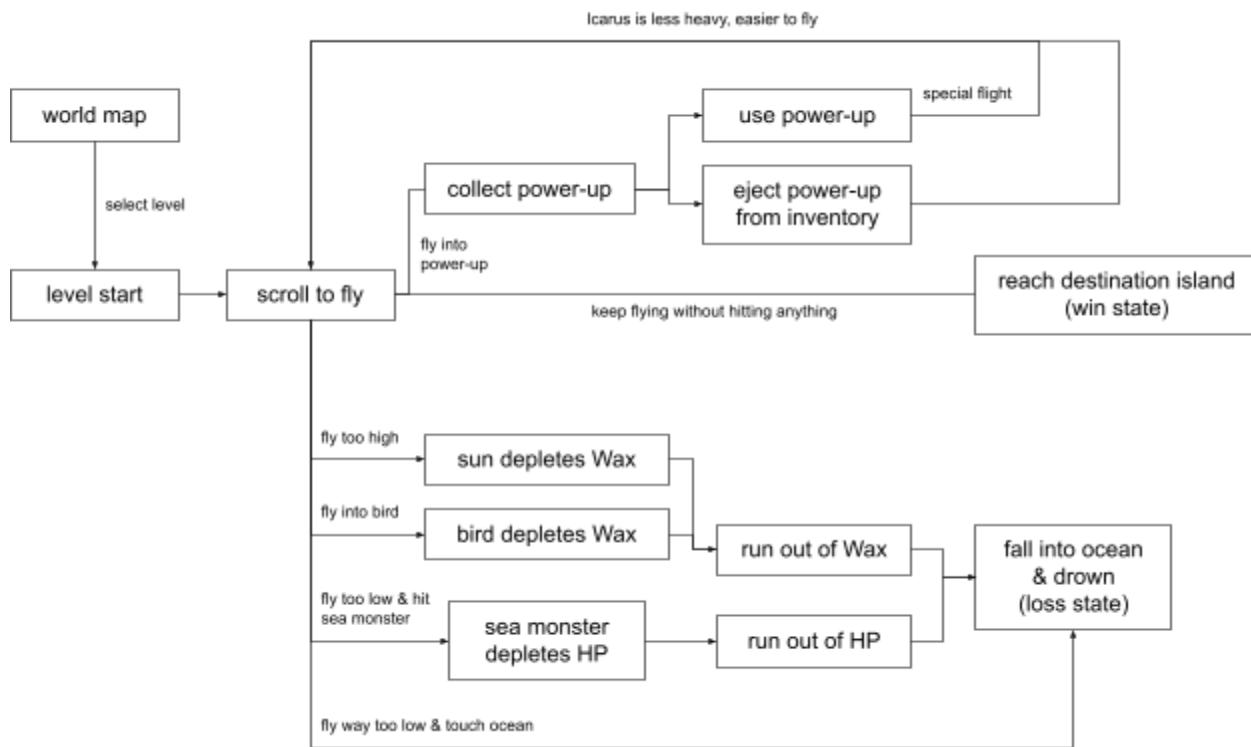
- The distance between islands will increase, meaning the amount of time the player spends scrolling to keep Icarus aloft will also increase.
- More species of sea monsters will be incrementally introduced, each with their own reach and damage dealt to be learned/memorized, and monsters as a whole will spawn more often.

2.4 Interaction Models

2.4.1 User Interface - Navigation and Movement

The player will control the game by using their mouse's scroll wheel (or scrolling on the trackpad) to adjust Icarus's vertical flight, the number keys to access inventory items, and the ~ key to eject items from the inventory. As shown in the Level Overview illustration under Section 2.3.2, there are two status bars at the top of the screen indicating Wax and HP level, respectively, as well as five inventory slots, each mapped to the 1-5 number keys.

2.4.2 Game Play Sequence and Levels



2.4.3 User/Environment – Obstacles and Props

The player interacts with the environment by flying through it and by colliding with objects, to Icarus's benefit/detriment. Collision detection will be handled by adding a collider component to Icarus and to each collide-able object (power-ups, sea monsters, birds, ocean). Collisions with objects do not affect the physics of Icarus's flight, but the type of object indicates how Icarus is affected otherwise. If hit by a sea monster, his HP decreases. If hit by a bird, his Wax decreases. If colliding with the ocean, Icarus drowns.

immediately. If colliding with a power-up, it disappears and Icarus adds it to his inventory.

Physics is used in calculating how scrolling affects Icarus's vertical flight. Icarus's upward speed is a function of how quickly the player scrolls; we will be utilizing tutorials such as this one (<https://youtu.be/-dgUoKeCAGg>) to do so. Otherwise, physics is largely ignored.

2.4.4 User/Character

The player experiences the game from a third-person perspective of their character, Icarus. The player interacts with Icarus by moving him and giving him power-ups to enhance his flight experience. The player moves Icarus by scrolling to control his vertical flight. Icarus flies horizontally at a constant rate throughout a level, so no player input is needed in this facet of his movement. By moving Icarus around, the player keeps him from flying too close to the sun, as well as from running into sea monsters and birds. The player also moves Icarus in the path of power-ups and equips them to Icarus from his inventory by pressing their slot's associated number key. If Icarus is holding too many power-ups, the player ejects the least-recently collected power-up by pressing the ~ key.

2.4.5 Character/Character

There are no other characters in the game besides Icarus, unless the sea monsters and birds count. Their collision conditions are discussed in 2.4.3. No physics is used when determining their movement.

Birds spawn once every few seconds at the right edge of the level scene, at some variable altitude, then fly horizontally left across the screen at a constant speed before disappearing behind the left edge of the scene. If a bird collides with Icarus, it disappears immediately, and Icarus sustains Wax damage.

Sea monsters spawn once every few seconds behind the ocean, rising at a constant speed, remaining still for a set amount of time, and then descending back into the ocean. If a sea monster collides with Icarus, it disappears immediately, and Icarus sustains HP damage.

2.4.6 Puzzle Design

Not featured.

2.4.7 Motion Tracking

Not featured.

2.4.8 Multi-Player

Not featured.

2.4.9 Mobile

Not featured.

2.4.10 Networked Play

Not featured.

2.5 Performance and Scoring

2.5.1 State Variables

What are all the character, environment and gameplay variables necessary to save/restore or pause/resume the game or virtual world experience?

- wax (int): How much wax is left in Icarus's wings.
- hp (int): How much HP Icarus has left.
- inventory (list/array/map): List of items currently in Icarus's inventory.
- powerupEquipped (bool): True if a non-status power-up is currently equipped. Prevents more than one power-up from being equipped at once.
- powerupTimer (int): Starts once powerupEquipped is true, then counts down from 5 seconds, indicating how long the power-up is in effect. Once it hits zero, powerupEquipped goes back to false.
- sunBlocked (bool): True if the Palm Leaf is currently equipped, allowing Icarus to fly past the sun threshold without losing wax.
- monstersRepelled (bool): True if the Monster Repellent is currently equipped, preventing monsters from being spawned.
- windBlowing (bool): True if Gust of Wind is currently equipped, increasing Icarus's horizontal flying speed.
- feathered (bool): True if Feather is currently equipped, slowing Icarus's speed of descent.
- drowned (bool): True if Icarus touches the ocean, ending the level immediately.
- currScore (int): Player's running score for the current level.
- hiScore (int): Player's high score for the current level.
- altitude (float): Icarus's current flying height.
- horizSpeed (float): Icarus's current horizontal flying speed. Remains constant unless Gust of Wind is equipped.
- liftSpeed (float): Icarus's current upward flying speed. Varies depending on the player's scroll speed.
- fallSpeed (float): Icarus's current speed of descent. Remains constant unless Feather is equipped.
- distance (float): Icarus's horizontal distance from the previous island.

2.5.2 Feedback

The game employs both positive and negative feedback to enrich the player's experience.

- **Positive Feedback:** As the player clears more levels, Icarus's max Wax and HP values increase, allowing him to last longer in the sun or take more damage from monsters without dying.
- **Negative Feedback:** As Icarus collects more power-ups, it becomes harder to scroll and keep him flying; this discourages item-hoarding. Additionally, birds will fly towards Icarus at random altitudes every so often, preventing Icarus from simply hovering in the optimal area between the sun and the ocean if the player becomes skilled enough to do so. Sequential levels also become more difficult, as is described in Section 2.3.4.

2.5.3 Performance and Progress Metrics

The game will monitor the player's progress by showing the player how much wax/HP Icarus has left via two status bars at the top of the screen. The player will know they have reached the end of the level once they see the destination island and Icarus breaches land. The player wins a level once Icarus has landed on this destination island. The player loses a level when Icarus falls into the ocean and drowns, and they must replay this level from the beginning to clear it.

3. GAME DESIGN - IMPLEMENTATION DETAILS

3.1 Design Assumptions

3.1.1 Hardware

	Windows	MacOS
Operating System	Windows 10 64-bit version 1909 revision .1350 or higher, or versions 2004 and 20H2 revision .789 or higher.	Latest MacOS Monterey
Processor	Quad-core Intel or AMD, 2.5 GHz or faster	Quad-core Intel, 2.5 GHz or faster
Memory	8 GB RAM	8 GB RAM
Graphics Card	DirectX 11 or 12 compatible graphics card	Metal 1.2 Compatible Graphics Card
RHI Version	DirectX 11: Latest drivers DirectX 12: Latest drivers Vulkan: AMD (21.11.3+) and NVIDIA (496.76+)	–

3.1.2 Software

	Windows	MacOS
Operating System	Windows 10	MacOS latest Monterey
DirectX Runtime	DirectX End-User Runtimes (June 2010)	–

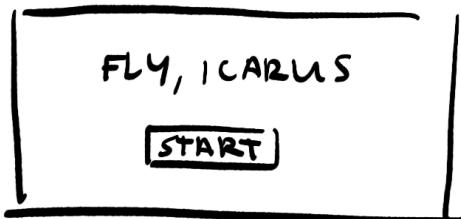
Additionally, we will use Unreal Engine 5 to develop the game.

3.1.3 Algorithms and Techniques

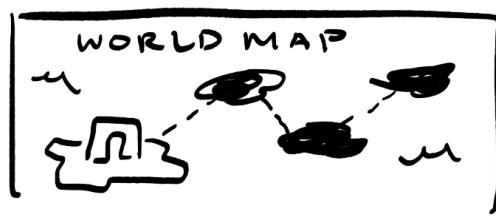
- **Icarus's flying velocity:** We will use something akin to this tutorial (<https://youtu.be/-dgUoKeCAGg>) to adjust Icarus's flying velocity as a function of how quickly the player scrolls up/down. Icarus's flying animation will simply be a loop of 4-ish frames of him flapping his wings, and his asset as a whole will be affected by both gravity and this flying velocity.

- **Wax bar update:** The Wax status bar will start at a max of 100 and decrease for every tick that Icarus is above a certain altitude, or will decrease a set amount every time Icarus hits a bird. If the value hits or goes below 0, the player will no longer be able to scroll, and Icarus will fall into the ocean and drown.
- **HP bar update:** The HP status bar will start at a max of 100 and decrease a certain amount every time Icarus hits a sea monster (amount varies depending on monster species). If the HP value hits or goes below 0, the player will no longer be able to scroll, and Icarus will fall into the ocean and drown.
- **Inventory affecting Icarus's flight:** Icarus starts off with a mass of 1, and each item in his inventory will add a constant value (maybe around 0.2) to his mass, thus increasing the effect of gravity on him and making it harder to fly. Ejecting or using inventory items will remove this mass from him and thus make it easier to fly again.
- **Bird spawning:** The birds will spawn every 5 seconds, give or take some random amount (within 1-2 seconds), at a random altitude on the right side of the screen, then travel at a constant speed towards the left side of the screen and disappear.
- **Monster spawning:** The monsters will spawn every 5 seconds, give or take some random amount (within 1-2 seconds) and multiplied by some constant determined by the monster's species, emerging out of the ocean and rising to an altitude also determined by species, lingering for 1-2 seconds, then dipping back into the ocean.
- **Power-up spawning:** The power-ups will spawn every 10 seconds or so at a random point on the right side of the screen, remaining stationary as Icarus flies towards their x-coordinate.
- **Wax depletion due to Sun proximity:** If Icarus flies past a certain altitude, he will begin to lose Wax every tick as a linear function of how many feet above this altitude he is flying.

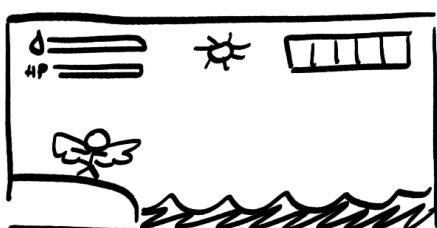
3.2 Storyboards



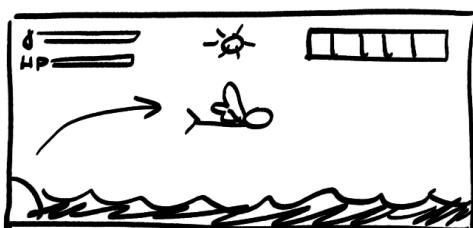
start page



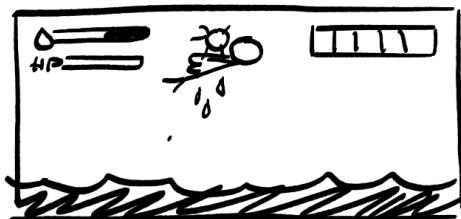
level select



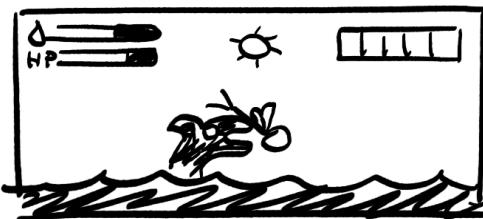
level start



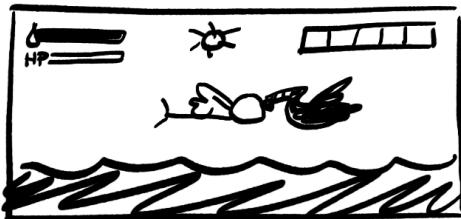
launch - scroll to fly



too close to sun - lose wax



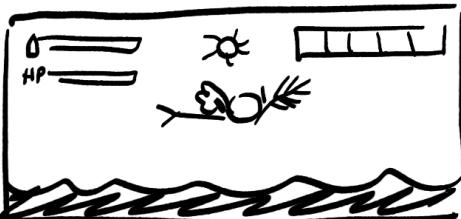
touch sea monster - lose HP



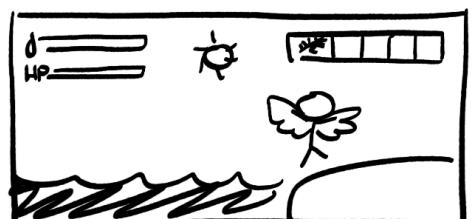
touch bird - lose wax



lost too much wax/hp - draw - LOSE:(



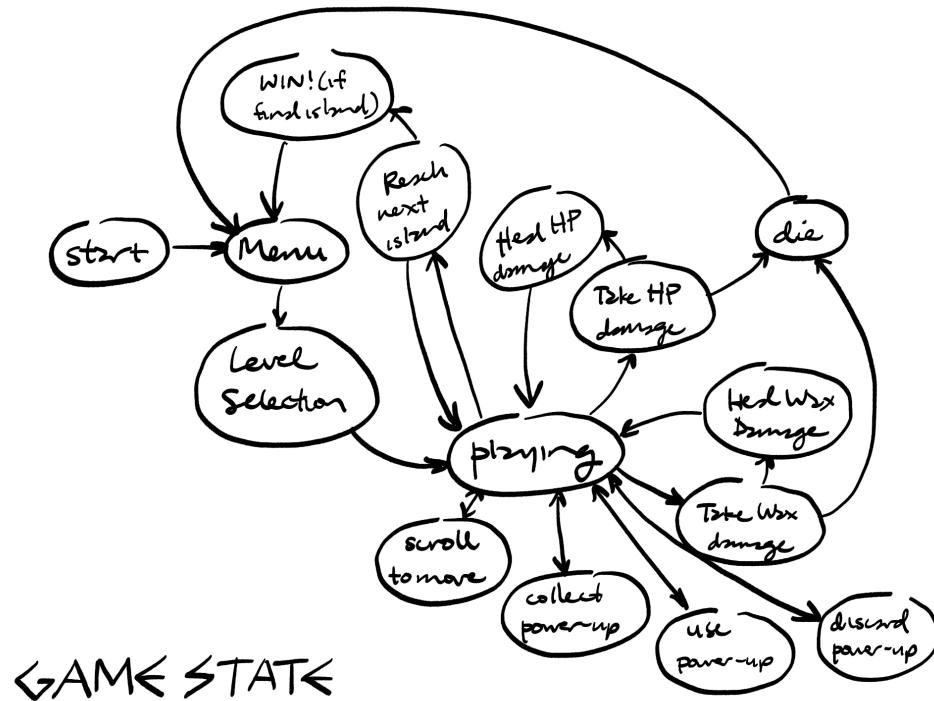
touch power-up - add to inventory



land on next 1st/2nd - win level! :)

3.3 Design Logic

3.3.1 Finite State Machines – Events/States/Effects



3.3.2 User Solution/Actions

The player will win the game by completing all the levels. There will be no puzzles in the game; the only strategy involved is mastering scroll-flight and figuring out when to collect/use/discard power-ups.

3.4 Software Versions

3.4.1 Alpha Version Features (vertical slice through total experience)

The alpha version of the game will feature:

- Single island level
- Inventory
- Wax meter
- HP meter
- Single species of monster that rises out of ocean periodically, decreasing HP if intercepted by Icarus
- Sun, which decreases Wax if Icarus flies past a certain altitude
- One placeholder power-up (non-functioning)
- Player can control Icarus's flight by scrolling
- Player can collect power-ups by flying into them
- Simple assets
- Score tracking

3.4.2 Beta Version Features

The beta version of the game will include all of the above features, plus:

- Additional species of monsters, with different reaches and damage amounts
- All status- and movement-affecting power-ups, functioning
- Birds that fly towards Icarus, decreasing Wax if intercepted by Icarus
- Player can equip power-ups by pressing associated number key
- Player can eject power-up by pressing ~ key
- Detailed assets
- Sound effects
- Persistent high score

3.4.3 Descriptions of any tutorial levels and/or self-running Demos

There will be no official tutorial level; however, there will be a Help option on the Main Menu that has screenshots of the game explaining the basic functions within a level, such as scrolling to fly or using power-ups with the number keys.

4. WORK PLAN

4.1 Tasks

Task 1 – Set up base code (All members)

Duration: 7 Days

- **Subtask 1.1.** Complete Flappy Bird UE tutorial
 - Find a Flappy Bird tutorial for Unreal Engine and complete it. The basics of this game should serve as the foundation for FLY, ICARUS.
- **Subtask 1.2.** Set up physics for scroll flight
 - The rest of these subtasks are self-explanatory.
- **Subtask 1.3.** Set death/loss state when Icarus hits bottom of screen
- **Subtask 1.4.** Set win state when Icarus reaches next island
- **Subtask 1.5.** Calculate and display current/high score

Task 2 – Basic HP/Sea Monster functionality (Krystal)

Duration: 5 Days

- **Subtask 2.1.** Functioning HP status bar
 - HP status bar is displayed, can be increased/decreased, reaches 0, never goes above max or below min.
- **Subtask 2.2.** Static sea monster
 - Sea monster that doesn't move and is always present (for testing).
- **Subtask 2.3.** Icarus hitting sea monster decreases HP
- **Subtask 2.4.** Losing all HP sets off death/loss state
- **Subtask 2.5.** Spawn moving sea monster
 - Make it so sea monsters spawn every so often and move up and down.

Task 3 – Basic Wax/Sun/Bird functionality (Monique)

Duration: 5 Days

- **Subtask 3.1.** Functioning Wax status bar
 - Wax status bar is displayed, can be increased/decreased, reaches 0, never goes above max or below min.
- **Subtask 3.2.** Static bird
 - Bird that doesn't move and is always present (for testing).
- **Subtask 3.3.** Icarus hitting bird decreases Wax
- **Subtask 3.4.** Icarus rising above Sun threshold decreases Wax
 - As a function of altitude, described in Section 3.1.3.
- **Subtask 3.5.** Losing all Wax sets off death/loss state
- **Subtask 3.6.** Spawn moving birds
 - Make it so birds spawn every so often and fly horizontally left at a constant rate.

Task 4 – Basic inventory functionality (Krystal)**Duration: 3 Days**

- **Subtask 4.1.** Functioning inventory
 - Display inventory that can be added to and deleted from.
- **Subtask 4.2.** Static power-up
 - Power-up that doesn't move and is always present (for testing).
- **Subtask 4.3.** Icarus hitting power-up adds it to inventory
 - In two or three sentences describe what you plan to do in this subtask.
- **Subtask 4.4.** Pressing associated number key uses power-up
 - In two or three sentences describe what you plan to do in this subtask.
- **Subtask 4.5.** Pressing ~ key ejects power-up
 - In two or three sentences describe what you plan to do in this subtask.
- **Subtask 4.6.** Inventory stock affects flight speed
 - In two or three sentences describe what you plan to do in this subtask.

Task 5 – Variant monsters and power-ups (Monique)**Duration: 3 Days**

- **Subtask 5.1.** Create ruleset for monster species
 - Determine how often certain species of monsters spawn, how long, and how much damage they deal.
- **Subtask 5.2.** Implement variant monsters
- **Subtask 5.3.** Include variant power-ups
- **Subtask 5.4.** Functioning variant power-ups

Task 6 – UI (All members)**Duration: 2 Days**

- **Subtask 6.1.** Main menu
 - Play button and credits
- **Subtask 6.2.** Pause menu
 - Like the main menu, but with paused game screen in the background.
- **Subtask 6.3.** Win/loss state screens (with reset/menu buttons)
- **Subtask 6.4.** Help menu (as described in Section 3.4.3)

Task 7 – Playtesting (All members)**Duration: 3 Rounds**

Task 8 – Final touches (All members)**Duration:** —

- *Subtask n.1. Polished assets*
- *Subtask n.2. Sound effects*
- *Subtask n.3. Background art for menus*
- *Subtask n.4. Background music*
- *Subtask n.5. Trailer/gameplay vids*

4.2 Milestones

4.2.1 Minor

- Basic gameplay
- Player movement
- Score tracking
- Win/loss states (death or reaching island)
- Working status bars
- Sea monster interaction
- Bird interaction
- Sun interaction
- Working inventory
- Item interaction
- Creature/power-up variance
- Better models
- Embellishments

4.2.2 Major

- For the **Alpha Version**, we would like basic functionality to be present so the game is playable, but not necessarily that polished. There would be a single variant of monster that can pop up and do damage, as well as a single default power-up that can be collected, used (maybe not with any effect, but at least to show they can be utilized), or ejected. There would be functioning UI and menus.
- For the **Beta Version**, we would like to expand the catalog of sea monsters and power-ups and implement them throughout the game (with proper functionality), as well as have more polished gameplay, better assets, sound effects, and background music. We will take into account user feedback and adjust Icarus's flight and interaction with game elements as necessary.

4.3 Development Schedule

