Progress Report

- Increment 2 -

Group #3

1) Team Members

- Grace Brill (FSU ID: KGB19A | Github ID: kbrill25)
- Elias Diez (FSU ID: EAD19F | Github ID: EliasADU)
- Jason Graham (FSU ID: JG19M | Github ID: JasonAGraham)
- Miguel Malayto (FSU ID: MAM18N | Github ID: mizai125)
- Santiago Ruiz (FSU ID: sr19w | Github ID: santiruz)

2) Project Title and Description

Owl's Omen

Owl's Omen is a first-person 3D parkour game, which is being developed in Unity using the C# programming language. The gameplay revolves around the ability for users to launch off surfaces by clicking on them. Levels are located in one centralized area, with many interconnecting paths and routes. The user's objective is to complete delivery routes within a certain time constraint while being pursued by flying enemies that can track the user. The game environment is interactive, including elements like dialogue and surfaces that affect player mobility and speed.

3) Accomplishments and overall project status during this increment

Overall

Progress during this increment has been much slower than in the previous increment, partly due to most challenging aspects being already completed, and the remaining aspects being more content-based than anything else. We have, however, gotten through a lot of the actionables.

Player Movement System

- Pull to surface added alongside the push away from surface as part of a design change. Inputs reconfigured:
    - Previously: Hold RMB to aim and press LMB while aiming to fire
    - Currently: Hold RMB to aim a pull, let go off RMB to execute the pull. Likewise for pushes and LMB.
- Slow-Down
    - Slow-Down added on Shift, limiter for it not yet added. Slow down just sets game speed to half of normal while shift is held. Tested with all movement features and none are incompatible with the timescale shift as they all account for time scale differences.

    - This change was done as a result of testing showing that most players struggle to move quickly enough, a slowdown mechanic allows the less twitchy-movement savvy players to still complete harder challenges.
- Fine-Tuning
    - Some fine tuning was done to the force and cooldown of pulls/pushes to prevent players from being able to remain airborne for too long easily.

Menu Systems

- Audio
    - Basic audio has been added to the Main Menu and Pause Menu.
    - Main menu plays music and a sound effect when clicking on buttons.
    - Pause menu has received the same sound effects.
- Save system
    - The save system is now working in terms of storing basic values on the clicking of the save button and the loading of these values on the load button.
    - Next step is to implement these values to load position, health, and level the player saved.
- Loading transitions
    - Added basic load transitions to scenes to make it a less jarring transition from menus to ingame scenes.

First Person Animations

- First person animations that were physics-based for the first iteration were stuttering as they were dependent on a linear interpolation function that does not work well with non-constant values (and less so with values changing at a changing rate). This function has now been replaced with one custom-derived for handling inputs whose velocity may be changing at any point and therefore works smoothly.

Visuals/FX

- Aiming reticle added with animations. Aiming reticle only shows up if either mouse button is held down and indicates when the player is pointing at a valid jump surface that is also close enough.
- Pull particle SFX added

Enemy

- Implemented an attack animation to the enemy when the enemy deals damage. Found an animation pack from the unity store and was able to create and store an animation boolean in order to make the animation work. Also had to fine tune the animation loop speed to make sure it looked natural and not overfly fast or lethargic.
- Fixed bug change where if the enemy got too far from the player then the enemy would just sit still, but now the enemy will go back to patrolling.

- Added enemy damage decay. Instead of the enemy having to run into the player or have some type of bullet or object collide with the player, the enemy will be able to do damage based off the distance of the enemy to the player. The closer the enemy gets the more damage that the enemy will deal to the player.

Damage Animation

- Added a red damage screen, where if the player takes any damage from the enemy then a bright red will flash on the player screen, as can be seen in other games like call of duty.
- When a player reaches below a certain amount of health then the red will stay on the screen notifying the player that he is very low health.

Surfaces Affecting Player Movement

- Further refined inputs and feel of all surfaces (unidirectional speed boost, multidirectional speed boost, jump boost for player already jumping, and multidirectional slow down).
- Implemented a multidirectional speed boost, which still does need some further refining but the foundation and functionality is already well-established. Resolved a few bugs that were encountered while developing this feature.

Dialogue System

- Refined the typewriter effect in how the system displays dialogue.
- Added a feature so that the user can skip through dialogue, which is a common feature in games.
- Dialogue speed is affected by the presence of punctuation.
    - Meaning that when punctuation like a dash is encountered, the speed temporarily slows down to mimic human dialogue and reading, which is another common feature in games.
- Began setting the foundation for interaction.
    - In the actual game, the player will interact with the dialogue, which will bring them to another scene for the dialogue.
    - This is something that will be implemented during the 3rd increment.
- Attempted to resolve the bug with the button.

Waypoint Marker

- Added a clamp feature to the marker that ensures that it does not move off-screen when the player looks away from its location.
- Refined the way the marker is displayed when the z-axis is negative (marker location is behind the player camera)
    - Instead of having the marker disabled, the x-axis and y-axis of the marker would be modified to display on the sides of the screen depending on the marker's coordinates while the z-axis was negative.

4) Challenges, changes in the plan and scope of the project and things that went wrong during this increment

Elias:

During this increment I did not struggle with most of the things I had to implement with the exception of the first person animation stuttering. The stuttering, after much trial and error, was found to be due to the player gameobject's movement, which the first person staff object "follows" through linearly interpolating its position, The player's movement can at any point have a one-time force or air friction acting on it, meaning that its velocity is not constant. Constant nonzero velocities already work poorly with linear interpolation, but non-constant non-zero velocities work even worse. I was able to find a derivation online for an interpolation function derived specifically to deal with this edge case, and after implementing it in Unity, it worked. However, there is still some stuttering due to the player moving in what is called the "Fixed Update" loop, which runs at a separate speed than the regular loop as it is used by Unity's physics system. This problem is sadly impossible to fix without removing the animations altogether. That is, there will always be *some* stuttering, which is why we do not see games use these first person animations. I, however, believe that the feedback the player gets from them is worth the tiny stuttering.

The only change in plan/scope is that we are scaling down a bit in terms of the main level, as it may be too ambitious to want a full, open, intricate, massive level for the player to play in.

Santiago:

During this increment I was really struggling on the damage formula for the enemy AI. Since this game is about movement and objectives, we didn't really want to have the player attack enemies but rather run away, we also knew that the player would be constantly on the move so traditional attacking wouldn't be the right move for us. So instead we decided to come up with a formula that would take health off the player based on the distance between the player and the enemy. The problem was trying to come up with / find a formula that would work with us. Elias helped by finding a formula that would help with this overall logic, but the formula isn't working exactly correctly. The farther away you were from the enemy the number would be a very large negative number, but if you were right on top it would be 0. This stumped me for a while because I wanted the reverse effect from the formula. I tried clamping the function and turning it into a negative and taking the absolute value but nothing seemed to work. Ultimately, I just figured that I could offset the formula by a certain number of damage, that ultimately solved the problem. So now instead of that damage approaching 0 when I was on top of the enemy, it would approach 0 + 10, or -10 + 10 and so on as I would get farther away from the enemy.

The only change in plan for this increment was that we were gonna have the enemy shoot "fireballs" / projectiles that would deal damage to the enemy, but we rendered it useless since the player will always be on the run.

Grace

During this increment, I struggled with the dialogue system. Particularly, I have had a difficult time resolving a bug where the button on the dialogue box shows up before the user can put in their response to the prompt. I have been working on it, but I've been slowed down since I got the flu and had extenuating family circumstances. I am hoping to resolve this bug before the last increment once

I am back to 100% and can focus specifically on that bug. I have made some attempts to resolve the bug, but none of them were successful. Instead of strictly focusing on resolving this bug, I began working on fine tuning other aspects of the dialogue system and implemented new features like punctuation affecting the typing speed, refining the typewriter effect, and letting the player skip through dialogue. I wanted to ensure that I was still progressing with the dialogue system and not getting stuck on the bug too much.

Jason

During this increment I struggled on the animations for the transition, the save and loading feature, and the sound effects. I have never worked with animations in Unity before so it was difficult to get them just how I wanted but I am at the point where I am happy with what we have. The saving and loading feature is probably the most important and difficult. I worked on it in the last increment and had to delay it to this one. I have it almost completely working where it does save game data. I am at the point where I need to actually implement this information into the game. I have also added sound effects and music to the menus which was quick and easy and makes the menu a lot more pleasing to interact with.

Miguel

During this increment, I struggled with the waypoint marker and its clamp feature. At first, I tried applying the same concept of how I clamped icons inside the minimap, but in a rectangular shape instead of a circular one. It worked for the most part, but ran into the issue of the clamp not scaling to different screen sizes, and the marker disappearing when its location was behind the camera (which was my previous solution to dealing with the negative z-axis in increment 1). It was a long and challenging process in order to find a solution to these issues. I ended up modifying the DistanceToGoal script so that it could be attached to the Main Camera, and made it so that the x-axis and y-axis of the marker are calculated based on the screen size and the position of the marker relative to the screen (left, right, top, bottom). The marker disappearing was solved by modifying the x-axis and y-axis of the marker depending on its location when behind the camera.

5) Team Member Contribution for this increment

- Elias:
    - **Progress report**: Whole of parts 1,2 (copy/pasted from I1). Part 3: Overall, Player Movement System, First Person Animations, Visuals/VFX.
    - **Requirements and design document:** Whole of parts 1 (copy/pasted from I1). Parts 1,2,3 of part 2 (most copy/pasted from I1). All purple boxes in the Class Diagram and all use cases for the player in the Use Case Diagram.
    - **Implementation and testing document:** Whole of parts 1 (copy/pasted from I1), 2. Part 3: Movement System, FP arm animation, General VFX. Playtest blurb in part 4.
    - **Source code:** Code for all classes in purple in the class diagram are written by me. This increment I wrote FPObjectsWeights and FPStaffSpin, as well as SlowDownMode and some changes to the other scripts.
    - **Video or presentation:**

- Grace
    - **Progress Report:** Accomplishments (Surfaces Affecting Player Motion and Dialogue System), Challenges (denoted by Grace section), Plans for Next Increment (denoted by Grace section), and Video (denoted by Grace section).
    - **Requirements and Design Document:** Overview (Made some minor edits), Functional Requirements (Surfaces Affecting Player Motion and Dialogue System), Non-functional Requirements (Portion about the feel of the surfaces that affect motion), Green Boxes in Class Diagram, and added to Use Case description (Jump section).
    - **Implementation and Testing Document:** Execution-based Functional Testing (Surfaces Affecting Player Motion and Dialogue System) and Execution-based Non-functional Testing (portion about player movement).
    - **Source Code:** DialogueObject.cs, DialogueResponseEvents.cs, DialogueUI.cs, Response.cs, ResponseEvents.cs, ResponseHandler.cs, TypewriterEffect.cs, and SpeedupMulti.cs.
    - **Video or Presentation:** See section 7 denoted by Grace.
- Jason
    - **Progress Report:** Accomplishments in this increment include audio in menus, save/load system, and scene transitions. The video presentation of my work, my challenges and changes while working on this increment, and what I plan to do in the next increment.
    - **Requirements and Design Document:** Added functional requirements and overview updates for the things I added like General Audio, Save/Load, and Scene Transitions and things I plan to work on next in these categories. Main menu and Pause menus in use case diagrams. Audio and Image documentation to dependencies.
    - **Implementation and Testing Document:** Execution-based Functional Testing (portions about save/load system, transitions, and audio)
    - **Source Code:** Crossfade animations, PlayerData.cs, audio clips, Data.cs, LevelLoader.cs, Player.cs, and Save.cs.
    - **Video or Presentation:** See section 7 denoted by Jason.
- Miguel
    - **Progress Report:** Accomplishments (Waypoint Marker), Challenges (denoted by Miguel section), Plans for Next Increment (denoted by Miguel section), and Video (denoted by Miguel section).
    - **Requirements and Design Document:** Functional Requirements (HUD Elements), Blue Boxes in Class Diagram.
    - **Implementation and Testing Document:** Execution-based Functional Testing (Minimap and Waypoint Marker).
    - **Source Code:** Minimap.cs, InsideMinimap.cs, DistanceToGoal.cs.
    - **Video or Presentation:** See section 7 denoted by Miguel.
- Santiago:
    - **Progress report:** Part 3: Enemy, Damage Animation , Part 4: Santiago, Part 5: Santiago, Part 6: Santiago, Part 7: Santiago Video
    - **Requirements and design document:** Parts 4,5 of part 2. Part 4 - Use Case Diagram Enemy and description of enemy. Part 5 - Class Diagram, Enemy and Health System. Red part.

- **Implementation and testing document:** Parts 4-7 of part 3. Part 5.
- **Source code:** Player Health System Script, Blood Image, Enemy Script.
- **Video or presentation:** See section 7 denoted by Santiago.

6) Plans for the next increment

- Elias:
  - Implement the momentum conservation mechanic mentioned in the previous iteration.
  - Iterate through level concepts and get at least one polished working level.
  - While the player is slowed down, raise bloom / brightness or other visual cue.
  - After momentum is implemented, some indication that the player has high momentum like wind going across the screen.
- Grace:
  - Implement the jump boost for players that are not already jumping.
  - Refine the "feel" of the multidirectional speed boost.
  - Resolve the button bug with the dialogue system.
  - Work in the UI of the dialogue system.
  - Work on gamebuilding and interfacing the dialogue system with gameplay.
- Jason:
  - Assist on level design and creation of the world.
  - Split each mission or level into scenes for saving and loading purposes.
  - Integrate the pause menu into every mission.
  - Add in game sound effects and music.
  - Finish save/load to have an in-game effect.
  - Fix any bugs and refine what we already have.
- Miguel:
  - Assist on level design and difficulty concepts.
  - Add in objectives for the player to complete/reach (waypoint markers will be attached to objectives).
  - Refine UI elements.
- Santiago:
  - Work on enemy model (visuals) (particle effects)
  - Add enemy abilities
  - Add visible vision cone to enemy, so player knows what to stay out of
  - Make it so enemies can't collide with each other
  - Assist on level design and make sure enemy can track and follow player through level

7) Link to video

Santiago: https://www.youtube.com/watch?v=84Y-vdlTc94

Elias: https://www.youtube.com/watch?v=4RJmZAgZjBs

Grace: https://youtu.be/gRjY_ssbQ44

Jason: https://youtu.be/GN1HWN4TdrQ

Miguel: https://youtu.be/p9E6zaaoEQI