Final Project Proposal

Technical Difficulties

Team members: Romarick Anderson, Evrin Lee, Ransom Liu

Core Concept

The concept of the game is to create a zoo themed outdoor escape room, where the player (A monkey) is trapped within a lion's den, scrambling to escape. The player needs to use the puzzles scattered within the den while also avoiding being seen and caught by the lions.

Features

Mechanics

Item interactions:

- We will scatter random pieces of meat that the player can toss to distract the lions. Lions will chase after, then eat the meat, meaning each piece can only be used once
- The player can interact with items such as keys to open doors. The player will have to open a series of doors in the correct order in order to reach the final door.
- The doors will have a code word behind each one, which the player will then have to tell the pigeon to receive the hint about the next door.
- The player is able to hold certain items from their inventory (Equip).
- The player can hide in bushes around the den and hide from the lion while being chased.

NPC interactions:

- Lions will remain docile if the player is at a distance. If the player is too close and within the line of sight of one of the lions, the lion will begin chasing the player until they either get out of range or the player gets caught, in which case the lion will turn docile, and the player loses a life and respawns at the spawn point.
- The pigeon will have dialogue that guides the player. The pigeon stays on a tree just outside the arena and comes down whenever the player calls it to provide the player with clues (quests) on what to do next.

- The pigeon dialogue will start by introducing the plot and more background information about the zoo, and tell the player that they need to unlock all the doors in order, then open the final door. Then they'll give a hint about the first

Branching dialogue:

- The pigeon will provide information like ("I remember there is meat behind that bush that you can use to distract the lion!") to help the game progress and introduce new mechanics to the player.
- The pigeon will also provide hints about the order the doors must be opened.
    - Behind each door will be a code, which the player must correctly repeat to the pigeon. If the player had opened the correct door, and correctly repeats the code word to the pigeon, the pigeon will congratulate them and give them a hint about which door is next.
    - If the player opened the wrong door, hence receiving the wrong code word, the pigeon will laugh at them, attracting nearby lions to the players position, and the player will have to evade the lions. The incorrect door will remain as the player left it, and the player will be able to use the same code when the pigeon hints at that door being next in order.

Quests:

- Find the keys scattered around the area that open a series of doors that are color coded to match the keys.
- Open the door corresponding to the hint given by the pigeon.
- Tell the pigeon the correct code word found behind each door to receive the next hint
- Receive the main key from the pigeon after giving it all of the codes in the correct order
- Find the main door and unlock it using the key given by the pigeon to escape

UI

- A HUD is used to display inventory at the bottom center of the screen. The player should be able to equip the items with the corresponding input.
- There would be an inventory to show which items the player has on them, for example, the keys and the meat pieces.
- The player's life count will also be displayed as part of the HUD in the top left corner

Content

Items

- Meat
    - The player can throw meat to distract lions
    - There are multiple pieces hidden in random places in the area
    - They are obtained by picking them up
    - There is a limited number of pieces in the area, and they are all single-use
- Keys:
    - Can be used to open their corresponding doors - you need to make sure you put the right one for each door
    - They are color-coded to match their corresponding door
    - They will be hidden in random places in the area
        - If the player uses the wrong key for a door, it will make a sound, and the nearest lion will be drawn to the player
- Doors:
    - Can be opened with its corresponding key. When the door opens, there will be a UI pop-up with the code word
    - Doors reveal a code word when opened with the correct key. The player can then tell it to the pigeon to receive a hint about the next door.

NPCs

Bird

- Give hints to the player about which door needs to be opened next through dialogue.
- Hints vary from obvious to more subtle.
- Gives background information to the player about the zoo.
- Starts inside the arena, and calls the player over to go over the introductory stuff and tutorial. Then flies to a nearby tree outside the arena. For the rest of the game, the player can “call” the pigeon when necessary (either to receive hints or tell it the code) and the pigeon will come to the players location, and they can interact with it.
- *calling does not attract lions.

Lion

- Will stay idle if the player is at a distance
- If the player gets too close, a chase state will initiate
- Will chase the player (within a certain distance) until they get close to the player, or until the player is out of the chase range
    - Players can hide in bushes to get lions out of the chase state

- If the player is caught by the lion, they lose a life. The player has 3 lives.

Audio

- There'll be a background track (forrest ambience or suspenseful music).
- There will be a roar sound effect for if the player triggers a lion.
- There will be a pigeon coo/laugh sound that attracts lions when the player gives the pigeon the wrong code.
- There will be a "clunk" sound that attracts lions when the player tries opening a door with the wrong key.

Quest content

- Decipher the hints given by the pigeon to determine the order the doors have to be opened.
- Find the key with the corresponding color of the door you think the pigeon is referring to.
- Open the door and remember the code, and repeat it back to the pigeon.
- Receive the hint to the next door, and repeat until all doors have been opened.
- Go back to the pigeon after doing this, and the pigeon will hand over the key to the final door.
- Go to the final door, and use the key to escape.
- Don't get caught by the lions as you do all this. You have 3 lives.

Environment

- It will look like a big lion enclosure at a zoo. There will be hidden doors for the player to interact with, and a bigger door as the final escape.
- There will be pieces of meat scattered in the area, which players can throw to distract lions.
- There will be bushes in the area for the player to hide in when being chased by a lion.
- Lions will go back to being docile if a player jumps into a bush.

System Architecture

Model-View-Controller pattern

Since the game will involve collecting items and using those items to unlock parts of the game, we will use an MVC pattern to separate the logic for UI elements, item properties, and tracking interactions between the player and the other NPCs. This will be done in order to make the logic of all 3 elements easier to use, understand, and send information to one another.

- Model
    - The key and meat item data will have special traits and properties to either progress the player through the escape room or hinder the lion's movements.
    - A player inventory that keeps track of which keys they have and how many meat chunks they have.
    - The constant value for the interaction distance between the lion and the player, but this should be able to be tuned in the inspector.
- View
    - The UI of the player. This will include an inventory that will show what items the player has collected, a life count that will decrease when being caught by the lion, and a dialogue box that will appear when the player interacts with the pigeon NPCs.
    - The UI also includes the rendered 3d assets for the player, the pigeon, and the lions.
- Controller
    - The logic that will determine the interactions between the player and the lions. An example of something that would be used a lot in ours would be the navmesh. When the lion chases the player, the controller class updates the lion’s position. If the lion gets close enough to the player, the player will lose a life and respawn at the initial spawnpoint.
    - The player movement and pointer system will allow the player to interact with items and doors. For example, if the player clicks on a door with the correct key, a piece of code will be sent to notify that the player has access and will open the door.
    - Logic that keeps doors locked until the player brings the correct key, which will open the door.

Singleton Pattern

- The game will use a singleton pattern to send information between our game logic. Since the game is an escape room, a lot of information needs to be sent throughout the game to change the states of rooms and what can be accessed, for example, how many doors have been opened in the correct order. This will be done by establishing a locator class that will send information to all the systems of the game that need certain pieces of information to activate.
- This is tied to our MVC pattern, because it helps us organize our classes and scripts so model classes only store and modify data, view classes only control UI, and controller classes handle input (decoupling).

Finite State Machine (FSM)

- A FSM can be used to determine the “state” of the lion. The lion will have three states: dormant, scavenging, and hostile. These states are determined by either the player’s distance to the lion, or an event that gets invoked from the GameController (singleton) class, which is then subscribed by the lion or NPC class that notifies it to change to a dormant state.
- FSM is also useful for us to control the npc or the player’s animations. The player won’t be needing animations because we are making a first-person game, but it will be used to control the lion’s animations.
    - The lion uses idle animation when scavenging
    - The lion uses running/sprinting animation when hostile