Aniss Hamouda, Sophia Pagazani, Claire Whelan, Trevor White
Dr Carolyn Lamb
CISC 226
28/1/2024

Project Proposal

POV: You Are a Bat

Game Concept

In our game, currently named *POV: You Are a Bat*, the player completes a series of puzzles using a unique power to provide temporary vision and affect the world around them in pursuit of deeper bonds with the characters of the game.

To this effect, it mixes elements from the puzzle game and dating simulator genres. The game is set in a world in which animals can speak and act like people. The player is a bat who meets a colorful cast of characters after they move to a new city. To gain their affection and potentially pursue romantic relationships, the player is encouraged to go on expeditions in a dark cave which is rumored to be filled with treasure. They use their echolocation to solve each stage in this dungeon. As they complete these puzzles, an overarching plot surrounding the player's past is revealed.

When setting our player experience goals for the project, a few large ideas became important to us. First, we want the player to navigate the world in a way that requires them to think outside the box. This generated the concept of completing puzzles in a dark world, using a mechanic to increase vision. It would also be used to break crystals and draw enemies nearer, among other uses. Having it affect the environment in this way not only assists in the puzzle elements, but introduces drawbacks that force caution and prevents abusing the feature. This avoids the purely tedious feeling that might otherwise occur, akin to navigating a maze in the dark. Considering all of this, the game's primary focus is this novel game mechanic–echolocation–which is detailed in the coming sections. On another note, we want the player to feel involved, attached and compelled to learn more about the world and its people. Thus, we have decided to incorporate the story very strongly through dating simulator-like conversions. These will induce emotion in the player, incentivise them to keep playing the puzzle-based sections, and produce a fun, satisfying gameplay loop.

Novel Idea

Our game's novel idea is the echolocation mechanic within the dungeon crawler aspect of gameplay. As detailed in the gameplay section below, there will be a series of puzzle-focused levels set in "caves", using point-and-click movement on a grid system. The player must find their way out of each level while collecting coins and trying to overcome challenges such as falling stalactites, hostile enemies and fragile crystals. The unique element of these levels is that the majority of the dungeon will be completely dark, with just a small grid surrounding the player being visible (along with twinkling to mark the location of coins). In order to avoid obstacles and find a way to escape the cave, the player can use

echolocation to create a temporary beam of visibility of a certain range in any of the eight directions along the grid (visualized in Appendix A). If the beam hits a wall on a diagonal within range, the beam of visibility will bounce the remaining number of squares off the wall. Echolocation reveals the location of walls, enemies and other obstacles, allowing the player to gradually map out the level and find resources. However, the visibility produced by echolocation will only be temporary, forcing the player to rely on their memory to navigate the level. The echolocation is also capable of interacting with the map. When the beam of visibility touches a tile with a crystal or a stalactite, it will damage them. When they take enough damage, crystals will shatter and stalactites will fall to the ground. This will add more depth to the echolocation and the puzzles that it is used for.

Abuse of the echolocation mechanic is also disincentivized by enemies who are attracted to the noise and move a set number of squares towards you each time you use it, forcing you to restart the level if they reach you. This adds an important complication to mapping out the level, as the player must not only memorize the layout of the level but attempt to keep track of enemies and track their changing locations each time echolocation is used, as there is always a danger of moving directly into an enemy's path in the dark. These constraints ensure that the player will have to engage mentally with the puzzle in order to solve it, relying on a combination of memory and calculation in order to avoid threats and find rewards. The process of evading enemies and avoiding damage to other obstacles while figuring out our game's novel mechanic should be a fun and engaging challenge for the player, while the premise of a well-known concept like echolocation should make the mechanic somewhat intuitive and minimize frustration in the learning stage. As a part of the overall game, echolocation will be fun as a core element of the levels, while also acting as an ability consistent with the player character's lore that ties the puzzle section in with the wider story.

Gameplay

POV: You Are A Bat is essentially a puzzle-based point-and-click dating sim. To explain better, gameplay can be split into two parts: the puzzle levels and the story. While they will be talked about separately, the two are intertwined, and each motivates interest for the opposite section. Within the puzzle levels, the player is expected to use the echolocation feature, the novel game mechanic as described above, to complete the goal by solving challenges. The levels will contain coins which can be used in the story section to further it along. Since the puzzle levels have the most technical game play, it is better to understand it first before talking about the story. Thus, we will include a small tutorial on game interactions and puzzle elements as the game begins, and it will be covered first in this outline.

At base level, the character moves on a grid. A player will click on a square that they want the character to move to, and the character will take the shortest path to that square. Additionally, as the map is dark and the bounds are not known to the player, if the character is moving towards a square beyond the boundary, it will still take the shortest path, but stop before it hits the wall. One of the major puzzle elements is that the whole level is not known to the player. They will not know the layout, enemies, or obstacles until echolocation is used to reveal a set number of squares. However, the squares are not lit up indefinitely and thus will return to darkness after a period of time. As such, the player will have to rely on their memory to navigate the level (see Appendix A for a rough idea of how this would look). Enemies,

another important puzzle element, pose a threat to the player. If the two collide, for example if the player moves the character to a space that an enemy is on, the character will go back to the beginning of the level, and all progress will be reset. The player must keep track of the enemies position, which is easy when echolocation is used, but there is a drawback. As a consequence to abusing the echolocation mechanic, enemies will move one space towards the character when it is used. The echolocation will stay up long enough so that the player can keep track of where they are, but their change in position could alter how the puzzle needs to be solved or change the strategy that the player is trying to use. The third puzzle element is obstacles. As far as we have figured out, this will be stalactites and crystals (which are essentially stalagmites). The former will hang from the ceiling and only be known to the player by a shadow on the ground when using echolocation. However, when echolocation is used, this will damage the stalactites, as well as the crystals. A stalactite will have three “lives”, meaning it can be hit by echolocation three times before it falls from the ceiling. If the character is in the same square as a stalactite when it falls, the level will reset and they will respawn at the start. Crystals also will have these “lives” and will shatter after that point, removing them from the map.

All these puzzle elements will either work with or against the player who is trying to solve the puzzle. The puzzle may have a goal, like lure all enemies into a pit, or break all the crystals (which is harder than it appears due to the enemy movement mechanic). Later on in development, we may choose to implement sub-goals for the player to gain more rewards, which will be optional. In terms of rewards, a motivation for the player to complete puzzles is coins. The player will receive this for completing a level, but there will also be coins scattered around the map, which can be collected for a greater total reward. These coins are important to the story which will be discussed later in this section. The coins in the dark will sparkle, thus guiding players to certain squares. At most, to keep the scope of the game small, there will be ten levels for the player to complete. As they complete them, the story will be pushed forward. To summarize the puzzle levels, at a 10-second level the player will be strategizing on how to move the character and use echolocation to complete the puzzle goal.

In between the story and puzzles, there will be a map where the player can navigate to different locations including: the shop, the levels, and the different character locations. These will exist in the sewer, some caves, and a city, respectively. This will give the player the freedom to control how they want to play the game. To connect the coins to this section of the game, there will be a shopkeeper, who as mentioned will live in the sewers. They will be some sort of creature or cryptid that sells the character items to use in the story. As brought up earlier, the game is also a dating sim, but unfortunately the shopkeeper is non-romanceable. The items that he sells will be gifts that the player can give to the romanceable characters in order to pursue them. These characters, who at this point are a worm, pigeon, and alligator, can be talked through from the map, where the player will select which one they want to talk to. This is entirely a personal choice, and the route chosen will not affect any gameplay other than the story. In these dialogues, the player will learn things about each character and will have to make choices on what to say. These choices will affect that character’s affection for you, which will be indicated visually through a status bar. Giving gifts that the character likes will also increase that bar. The wrong choices may make the character lose affection. This section will hopefully have adequately compelling stories to keep the player engaged and motivated to play the puzzle levels. Ultimately, at the 10-second level for this section, the player will be choosing which character they wish to pursue and how they are spending their coins to accomplish that.

Target Platform

Our target platform is a single PC with a mouse and keyboard as the controller scheme. We will most often utilize the mouse to control the character, abilities, and more. Our game uses a "point-and-click" system, thus this platform really lends itself well to what we want to achieve. With movement in the dungeons' grid system and choices in the map and dialogue sections using basic left-click selection, we decided to implement right-click for our echolocation mechanic. While there may end up being some optional keyboard shortcuts, we wanted to design the game such that with the mouse alone, the player will have the accuracy they will need to solve puzzles and interact with the world.

Development Tools

Game Development	Unity Hub and Editor, C#, Visual Studio Code
Art Assets	Piskel, Autodesk Sketchbook
Audio Assets	LMMS, Adobe Audition

We will be using Unity for game development. Unity is a game engine which supports many features for 2D game development, including grid layouts which will be useful for our dungeon levels and 2D physics for designing object interaction in these areas. Game design in Unity centers around different scenes in which you place objects and design interactions, which will make it easy for us to design the different areas that make up the structure of our game. Unity allows for the importing of 2D assets as objects and backgrounds, which is useful as we have external tools in mind for making the game's art. The script writing for events and special game object behaviors in Unity is done with C# through Visual Studio Code.

As for art assets, we will be using Piskel and Autodesk Sketchbook to create our own assets. The dungeon sections of the game will have a pixel-art graphic style. This will work well with the tile-based environments of the dungeons. It will also speed up the process of creating the assets for these puzzles because it can proceed individually from the level-design process; using a sprite sheet to paint the layout of each individual puzzle does not require the artist to create unique assets for every level. For this purpose, we are using Piskel to create sprites. Piskel is a free, browser-based sprite editor. It is very handy in that it can be used to design multiple sprites on their own canvases, which are automatically converted into a single sprite sheet. It also allows the artist to visualize how an animation based on the sprite sheet may look without having to import it to Unity first. Moreover, the out-of-dungeon elements, meaning the map, shop, NPCs, backgrounds and UI elements will most likely be created using Autodesk Sketchbook. It is a free drawing app that offers a variety of features that will make designing these more streamlined, such as layers and different brushes. This works for our purposes, as we want these to have a different visual style than the caves, so they will not be pixel art. Also, these assets will likely be larger and quite varied in their composition, so designing them in small pieces like the dungeons would be less effective.

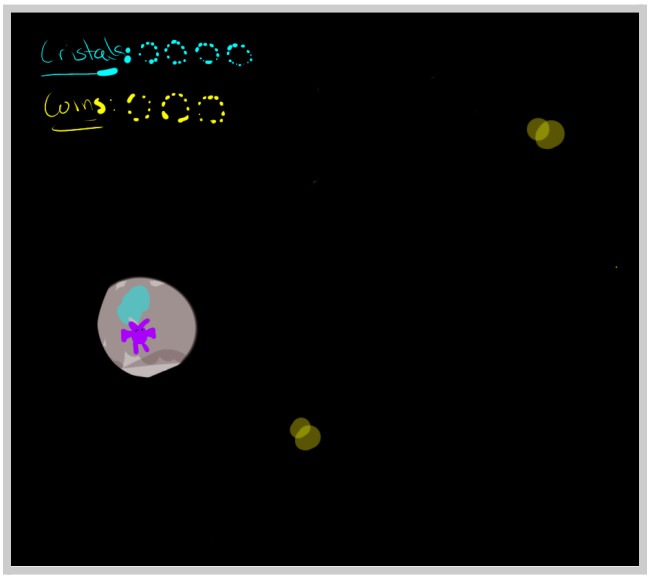
In terms of music and sound effects, we will use LMMS (a free DAW) and Adobe Audition (to mix). For music, we will try to create simple loopable tracks for each section of the game and for each character. Sound effects can also be created with MIDI in this program. As this is an additional piece that would make the game more polished, if there is a lack of time we will grab sound effects and music from freesound.com and incompetech.

Coordination Plan

Our plan to coordinate in completing the project involves very good communication. We have decided that we will have weekly meetings on Thursdays at 12:30 in the apartment of one of the group members. During these 1-2 hour long meetings we will discuss our progress, as well as any ideas or concerns we might have for the project. We will also discuss what we plan on completing before the next meeting. We may also complete some work that needs to be done together, or discuss an appropriate time to meet up to complete said work. When we are not in meetings, we will communicate through Discord. We will all try to work on our own sections of the project, but we also anticipate helping each other out through Discord or in person. Work will be synced using Github, allowing all members to remotely see what changes were made and add changes of their own. Finally, we have also already decided on a tentative timeline for when the different parts of our game must be completed, as well as the different focuses for some team members, such as creating assets or developing the story. We anticipate that these will change during the development process, of course, as priorities shift as a result of playtesting feedback or challenges that may arise during implementation.

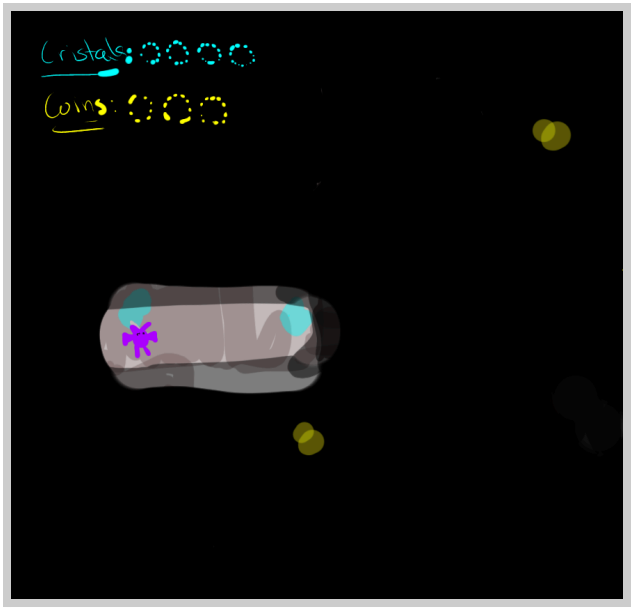
APPENDIX A

Dungeon Concept Art

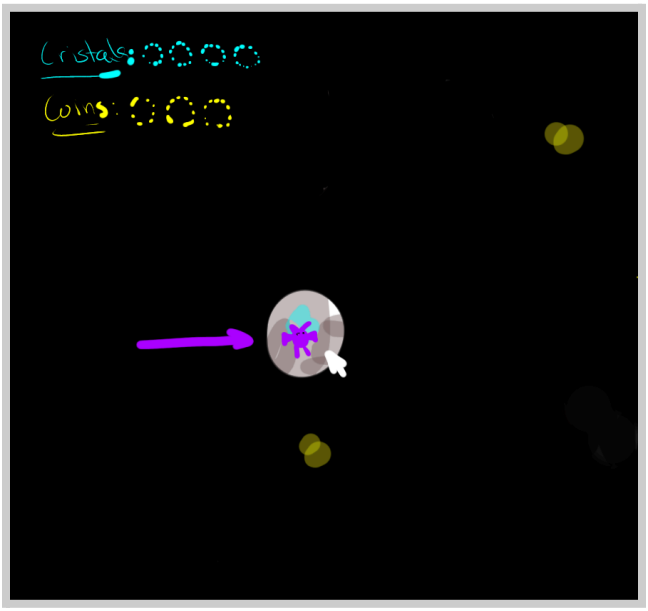


(I) Player idling, with a small radius of vision

(II) A puzzle layout, without the dark overlay



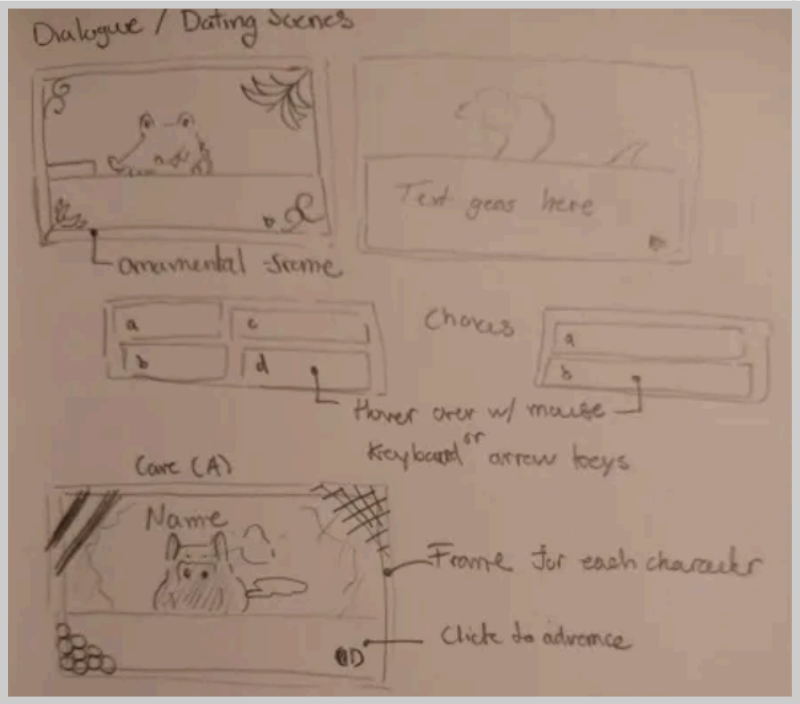
(III) Linear vision increase via echolocation



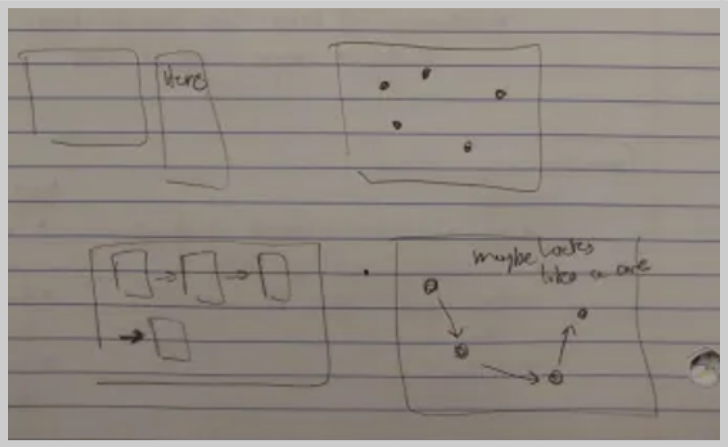
(IV) Point-and-click movement

APPENDIX B

Out-of-Dungeon Concept Art



(I) Concept art for dialogue UI, backgrounds, and characters



(II) Rudimentary visualizations of level selection and world map layout