Game Creation - Echoes Through the Screen

A Major Qualifying Project Report

Submitted to the faculty of

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor of Science

By

Zesheng Chen

Nelson Diaz

Bright Lin

Submitted to

Professor Charles Roberts

Worcester Polytechnic Institute

Abstract

Echoes Through the Screen is a murder mystery visual novel that applies both traditional computer controls along with the affordances of a cellphone. Players interact with characters and venture into the mystery of a murder at their college by discovering clues scattered throughout the game. A core aspect of the game is the player's real-world phone acting as a stand-in for the phones they interact with in-game. The goal of this MQP was to create a game with a novel control system and choices that enhance the gameplay experience by increasing replayability and exciting the player. After developing the game, we conducted playtests to evaluate the reaction of players and determine if we succeeded at our design goals.

Echoes Through the Screen was developed in the Unity engine for a Windows PC and Vue.js for Android phones in the Chrome browser. This report describes the process of creating our game from the story writing and game design to the technical design details. Throughout the project, our team was able to create a novel product, overcoming hurdles and pitfalls along the way.

Acknowledgments

Our team would like to thank the following for their contribution to the development and completion of this project:

Professor Charlie Roberts for advising our project, giving us feedback, and supporting us through the MQP.

Botao Han for writing the game’s story, helping with the minigame designs, and for the development of our game’s phone UI and minigames.

Shiming De for helping in the development of our game’s phone UI, minigames, and helping us find artists for our character models.

MingZhang Guo for helping us with our 3D models.

All of our playtesters provided us with their time and feedback.

Table of Contents

List of Figures

1. Introduction

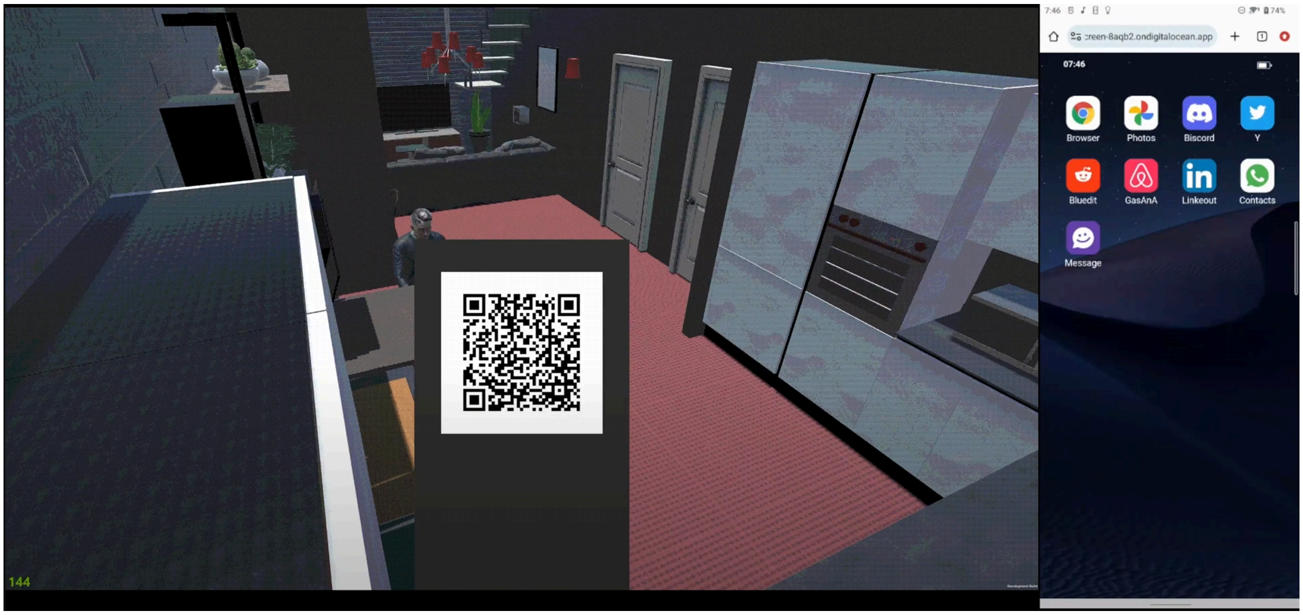


Figure 1: Side-by-Side Gameplay - UI for our Phone and Computer

Echoes Through the Screen is a 2.5D story-driven, single-player, detective-RPG game set in modern-day Worcester. The game is targeted towards an audience that enjoys a game that is plot-driven with many aspects of mystery solving and gameplay that involves small puzzles and mini-games. The player assumes the role of a college student named Tom, who is studying at WPI (Worcester Polytechnic Institute).

The story revolves around Tom, the protagonist, who accidentally discovers a peculiar photo of a female college student on his good friend Jacob's phone. This discovery inadvertently involves him in a murder case. Just when Tom starts to suspect Jacob as the murderer and secretly searches for clues to the case, he realizes that the situation is far more complex than he initially thought. As Tom continues his investigation, his relationship with Jacob undergoes unforeseen changes. The actions players take as Tom will influence the entire narrative and the fates of those around him, including his own.

One of the core aspects of our game is the use of the player's phone in the real phone as a stand-in for the phone that the player interacts with in the game. There are times periodically throughout the game where the player will shift their attention from the computer screen and instead interact with the interface we built for the phone (Figure 1). Our game is designed to take advantage of features available through a phone's browser that would not be available on a computer, such as touch control and vibration. This gives us the opportunity to make the interaction with the phone immerse the play in the game world in ways that we would otherwise not be able to.

This paper is separated into several different sections that venture further into each part of the team's development process on *Echoes Through the Screen*. Each section contains important background information to contextualize it. Our story section talks about our writer's inspirations and their process of writing throughout the pre-production along with how it is tied into the design of the game. Next, the design section breaks down what inspired our designer to create many of the puzzles and mini-games throughout the game. Our technical section begins by discussing our research on different computer and mobile connection methods which will talk about how we decided to implement a specific method for our game's main mechanic. Afterward, we will discuss our programmer's development process and also many of our technical hurdles throughout production. We will then only briefly talk about the art for our game due to our team not having a dedicated artist. Our methodology section describes how we approached our playtesting phase and what changes we made based on player feedback. Finally, we conclude with a reflection on how *Echoes Through the Screen* realized the goals we set from the beginning not only as a game but also as a graduating project. We also discussed gameplay aspects we could have improved on for the development process.

2. Story Development

For the game's story, we wanted intrigue with a small amount of suspense and thriller elements. We found several games with interesting storytelling details when searching for similar games. "Who's Lila?" is a horror and detective game that inspired us when we thought of the plot (Garage Heathen, 2022). Although we did not design a script containing Cthulhu elements like those in "Who's Lila?", some of its storytelling presentation techniques are worthwhile. There is another game called "NORCO", a point-and-click narrative adventure game with a heavy dose of American Southern Gothic (Geography of Robots, 2022). This game's narrative aspect is very close to what we expected our game to be. In designing the plot and writing the script, we learned from this game and made our game story-driven, so it would be based on the narration between characters and the interaction between players and in-game objects; this is how we push the plot forward.

While writing the plot and script, we were also inspired by novels and manga on suspense, abuse, and Stockholm Syndrome. For example, the famous thriller "Confessions" by Japanese novelist Kanae Minato (Minato et al., 2014), the Japanese manga "Erased" (Kei, 2012), and the Chinese novel "Ban Cheng" (Zhewanzhou, 2017). These books and manga gave us many ideas for writing an intense and thrilling plot and script.

Another important point of the story is the creation of characters in the game. In this suspenseful thriller game we have designed, each character plays a key role, forming an intricate storytelling web. First, the player will be an ordinary college student who becomes the story's protagonist. This character will be the player's proxy in the game, and through the player's perspective, they will explore the truth of the murder case.

The remaining four characters are each unique and have a strong relationship with the game. First is the victim, a popular college student, where the murder takes place, beginning the entire story. Next is the murderer, hiding in the corner and trying to hide the truth from the player. The other two characters are the key clues in the game. Each one may be involved in deep secrets, but whether or not they can be trusted becomes a question that players need to think deeply about.

The Relationships and interactions between these characters will trigger a tense atmosphere in the game. Players must interact with these characters to collect clues and deduce the truth. The personalities, behaviors, and dialogues of these characters will add depth and complexity to the game, plunging the player into a world full of suspense and tension.

Meanwhile, the game is set in a university and incorporates elements of everyday life. These characters are not only crucial players in the murder case but also feature universal college life characteristics, such as exam pressure, friendship entanglements, and love relationships. This sense of everyday life makes the game more relevant to the player's life and makes the sense of suspense stand out in everyday life. Through the diversity of these characters and the setting of the game's background, we hope to present players with a game experience that is both realistic and heart-wrenching, allowing them to immerse themselves in the dual thrill of suspense and thriller.

When developing the story for our game, our initial step was to draft an outline to help us design the script more easily. This outline was not isolated; it was created concurrently with the level design to ensure that the narrative and gameplay elements complemented each other, and we discussed it in more detail in the Design Development section. As we moved from the outline to writing the entire script, one of our biggest challenges was getting a good story ending. We wanted to ensure the ending felt reasonable, resolved the central conflicts, and maintained a suspenseful mood and tone matching our game. Besides, our narrative approach centers around the dialogue. Our dialogue is a crucial key for story progression. We carefully designed it to embed essential pieces of information, requiring players to pay close attention to conversations. This design choice ensures that players are deeply engaged with the narrative because missing critical details in the dialogue could miss vital clues or story elements.

3. Design Development

The design for this game has many aspects. First, we considered the story and level concurrently. We have also designed flowcharts for our plot (see Fig. 2), which benefitted us in designing levels and scenes.

With flowcharts, the design team can identify possible logic errors in the game, and we can identify the choices and decision points that the player needs to make in the game. This visualization presents the game's character relationships and storyline in a graphical form. Each team member can see the game's structure, improving teamwork efficiency.

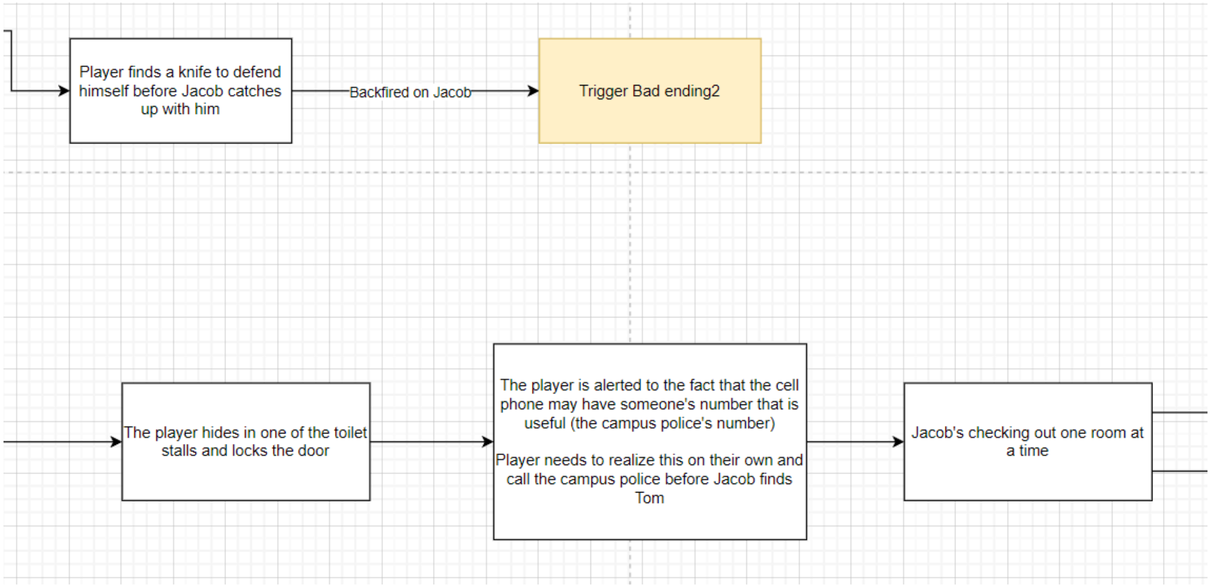

Player finds a knife to defend himself before Jacob catches up with him
Backfired on Jacob
Trigger Bad ending2
The player hides in one of the toilet stalls and locks the door
The player is alerted to the fact that the cell phone may have someone's number that is useful (the campus police's number)
Player needs to realize this on their own and call the campus police before Jacob finds Tom
Jacob's checking out one room at a time

Figure 2: Part of the Flowchart Plot

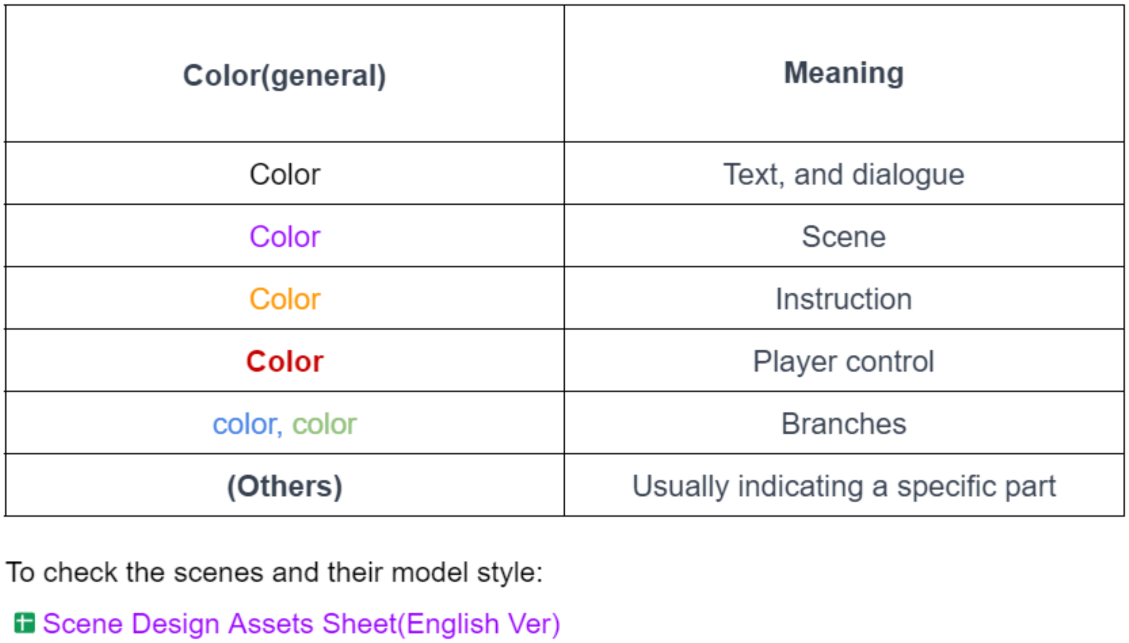
From the games that inspired us, many surprised us with their exciting level design and scene design, and some of the minigames puzzle-solving gave us some insights. For example, "Who's Lila?", has excellent camera and pixel art design. Some of its scenes have fixed cameras that players can see from different angles, creating mystery and innovation. Another game is "Do Not Feed the Monkeys" and its puzzle design is really incredible (Fictiorama Studios, 2018). This game inspires us to include minigames in which players navigate the PC in-game and search for clues.

In our game, we will have in-game choices for players to make; some of them may impact the gameplay and even alter the storyline. Multiple endings will be affected by players' decisions during the gameplay. Meanwhile, players can interact with particular objects, such as a phone that can check photos and make calls or a key that can unlock a lab door. These designs will bring a variety of fruitful gameplay for players so they can play a second or even a third time to reach the true ending.

Based on the above ideas, the game mechanics will include character control, object interactions, decision-making, and puzzle-solving. For the scene design, since the game is based on 3D pixel art, we took inspiration from ‘Who's Lila?”, “NORCO”, and “World of Horror” (panstasz, 2023). To introduce more realism into the game, we visited some real places, such as the WPI campus and Back Bay in Boston.

The game is structured over four days, from Day 1 to Day 4, unfolding in a linear narrative. New and existing scenes are featured daily, along with minigames to further immerse players. In addition to simultaneous computer and phone interactions, there are also standalone, computer-based, puzzle minigames. The early part of the story is highly guided, with players following the linear narrative. After the third day, players are presented with choices in addition to scripted character dialogues, as mentioned above.

Initially, we planned to have a second playthrough, a route that may lead to the game's true ending. However, in the middle of the implementation, after a meaningful discussion with every team member, we realized it would be out of scope since we already had a large amount of work to deal with, considering the challenge of implementing phone connection and minigames. Our first playthrough (Day 1 - Day 4) contained some endings that we mentioned earlier in this section. It was unfortunate that we had to give up the second playthrough. However, our first playthrough was undoubtedly filled with necessary and intriguing storytelling and gameplay that were carefully designed, and provide a fun and novel experience to the players.



Color(general)	**Meaning**
Color	Text, and dialogue
Color	Scene
Color	Instruction
Color	Player control
color, color	Branches
(Others)	Usually indicating a specific part

To check the scenes and their model style:

Scene Design Assets Sheet(English Ver)

Figure 3: Color-coded System of Design Document

For the ideation of our game and level designs, we organized our game design document using a color-coded system to differentiate between elements like dialogue, system mechanics, and player interactions. This approach helped us stay organized and keep our ideas flowing smoothly. The brainstorming phase for our minigames and level designs was engaging. We explored dynamic interactions between phone and PC platforms, and developed exclusive minigames for PC. Our puzzle designs, particularly those involving elements of cryptography, were crafted to be thought-provoking yet manageable. Finding the right balance for the difficulty of our minigames and levels was a challenge. Our goal was to design simple and manageable puzzles, aiming for a level of challenge that keeps players engaged and motivated.

4. Technical Development

In this section, we dive deeper into why we selected certain software frameworks and how we used them to make *Echoes Through the Screen*. At the end of the section, we discuss the challenges we faced and how we overcame them.

4.1 Computer and Mobile Connections

Even in the early stages of our game's development process, we knew the main mechanic we wanted for our game was using both the player's computer and mobile devices. This is not a never-before-seen concept but it is rare. It is a very intriguing way of approaching games and we decided that our game would mainly focus on this mechanic. The games that we looked at that did have this connection between the two devices often used the mobile devices as a sort of controller for a desktop game. We wanted to make our game different, so we decided to create a game where the mobile device in the physical world would represent the phone object inside the digital game. To make this happen, we needed to do extensive research on different ways to make the connection possible. Looking at many previous games that employ this specific mechanic, we discovered that wired connections were not common, so we decided to follow previous examples and avoid them.

4.1.1 Mobile Development Platform

We ultimately decided to implement wireless connections for our game, and researched how to make this happen. In the next two subsections, we will talk about the pros and cons of using a Unity app on the desktop paired with a web page running on mobile devices.

4.1.2 Unity App

We researched games that were using mobile apps that could connect to desktop games. In this setup, players would have to get the main game on their desktop computer and also download the mobile app associated with the game either in the App Store or Google Play Store. A good example of this is a game called Eon Altar. Eon Altar is a role-playing co-op game that uses players' mobile devices as controllers and for other game mechanics.

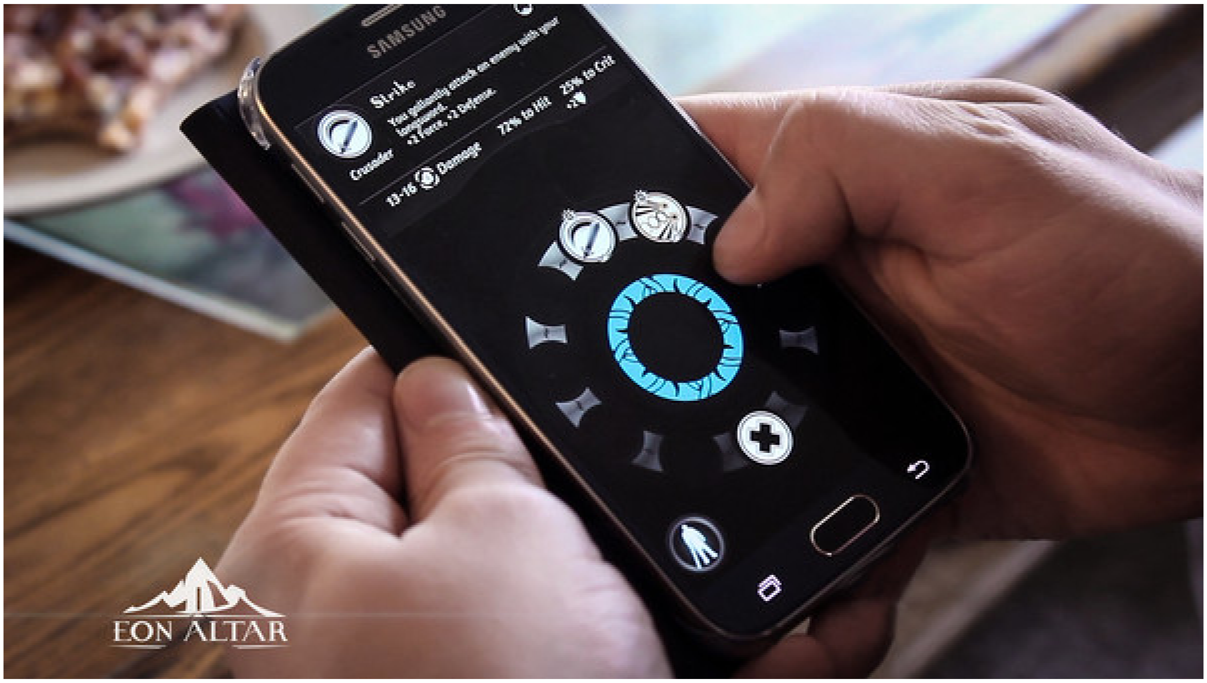


Figure 4: Eon Altar Mobile App Example (Flying Helmet Games, 2016)

One of the pros of making the mobile side an app is that it allows us to perform development for both the computer side and mobile side using Unity. Doing this would allow us to take advantage of one of Unity's many networking libraries in order to share information between mobile devices and a desktop computer.

However, choosing to create an app also means that we would need to keep the app on whichever app store we wanted users to download it from. This not only means that the app would need to be continuously updated, but we would also need to make whatever payments the app stores required. Additionally, we were concerned it would end up being difficult to have the mobile app work across multiple operating systems.

4.1.3 Web Browser

While we were researching the connection using the Unity app, we also started to look into using the web browser as an alternative distribution channel. With this method, players can open a web browser on their mobile devices to connect to the main game on their computers. As we continued our research, we found several games that were using web browsers on their mobile devices to connect to desktop computers. A prime example would be the Jackbox games. Jackbox games are a series of party games that have players connect their mobile devices to a website to participate. We decided this was the best path to follow, however, due to there being variable support for access to the various sensors in mobile devices in different browser / operating system combinations, we primarily targeted Chrome on Android phones.

4.2 Software

With this decision made, we had to choose a couple of different software engines to create our game. We needed a game engine to create the main game on the computer and another one too create different interactions or mini-games for mobile devices. We also decided on source control software to better manage the work between different programmers.

4.2.1 Unity

Our choice of game engine for the desktop computer aspect of our game was Unity. Multiple members of our group had previous experience with Unity and were confident with operating inside this game engine. Since we wanted to achieve 2.5D graphics, we decided to use 3D models in Unity in order for our scenes to have some depth and for our player to move towards or away from the camera. In order to gain our pixel art style, we found a way to create a shader in Unity that would downscale the resolution of the game, making it look pixelated. This shader will be further explored later in section 5.

Even though there was little debate on our choice of game engine, a major change came from the higher-ups of the company that owns Unity which made us second-guess it. Around September of 2023, Unity announced that as of the 1st of January 2024, developers would be expected to pay an additional fee per game download after reaching a certain threshold in revenue (Francis, 2024). This was a change that discouraged many developers from using Unity. We were also swayed to move away from Unity, but after a short amount of research and looking back at the work we have already accomplished, we decided that changing game engines would have done more harm than good.

4.2.2 Git/GitHub

Source control is a vital part of any software development project and was just as important in ours. It was simple to choose a source control for our team because we all had a clear idea that using Git for version control and GitHub to host our repository was the right choice. Many members of our team had extensive experience with both Git and GitHub either from previous classes or personal projects, making these systems easier to understand and quicker to work with. We also decided to use GitHub Desktop as a main tool to effectively manage our work, since it can provide members of our team who weren't focused on development with a simpler interface. This enabled team members that did not have as much experience with Git's traditional command-line interface to quickly learn core concepts. Another major system we implemented was blocking off the main branch so that no one could push to it without permission. This way each person will have to push to their own branch and the main branch can be secure from any changes that could mess up previously working versions.

4.2.3 Connection Protocols: WebRTC & Socket.IO

WebRTC is a protocol used to establish peer-to-peer connections over the Internet (“WebRTC”, n.d). It is often used for real-time communication on the web for things such as video calls and voice chats. We used WebRTC to communicate information between the player’s phone and computer. This both cuts down on latency when sending messages and increases scalability since messages do not need to be routed through a central server.

In order to negotiate a WebRTC connection, it is necessary for the peers to exchange information about each other. To do this, we use Socket.IO. Socket.IO is a web protocol built to have bidirectional communication between a Socket.IO client and a known Socket.IO server (“Introduction | Socket.IO”, n.d). Both the phone and the computer run Socket.IO clients that connect to the web server hosted on DigitalOcean which is also used to serve the web app for the phone. The web server then acts as a temporary middle-man that helps the phone and computer establish a WebRTC connection between each other. After the WebRTC connection is established, the two peers no longer need to use the web server in order to communicate. Figure 5 is a graph that represents how the different aspects of the connection interact with each other. To use Socket.IO and WebRTC with the Unity engine, we used two open-source libraries: SocketIOUnity and com.unity.webrtc (itisnajim, 2021) (Unity-Technologies, 2019).

When exchanging messages with each other, the phone and computer both conform to the same format. The first part of the message is the minigame number that is currently being played. The second part of the message then holds the message of what to do or what has happened in that minigame. To see the complete list of messages, see Appendix A.

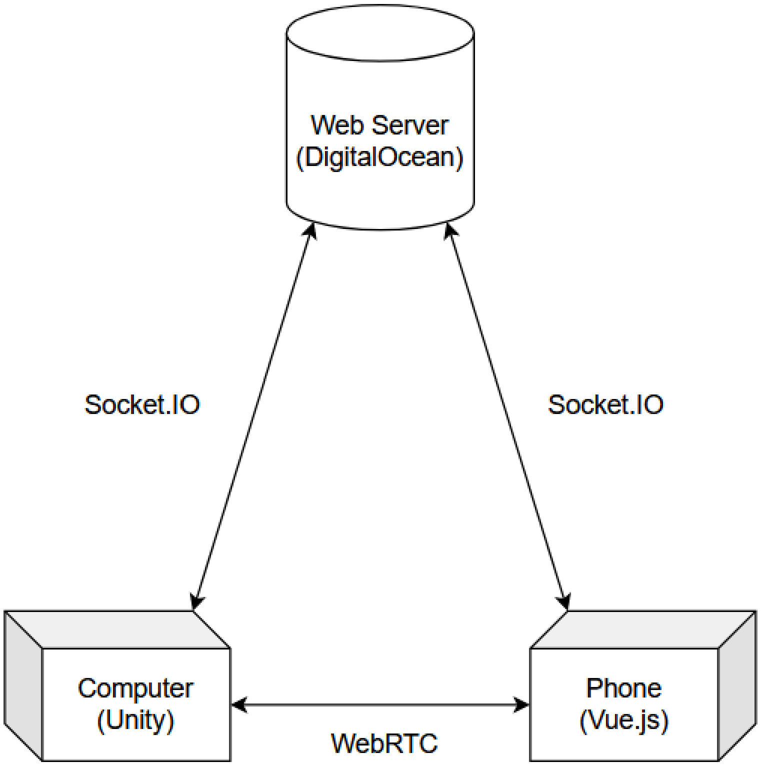


Figure 5: Graph Representation of the Connection Communication

4.2.4 Web App: Vue.js

For the phone, we have the player access a web app by scanning a QR code displayed on the computer. This web app was developed using the Vue.js framework for JavaScript. This allowed us to easily create a dynamic user interface that was easy to update during gameplay. Depending on the player's current progress through the game, different objects will be added or removed from the phone's interface. Examples of this are photos being added/removed from the 'Album' app or notifications being displayed.

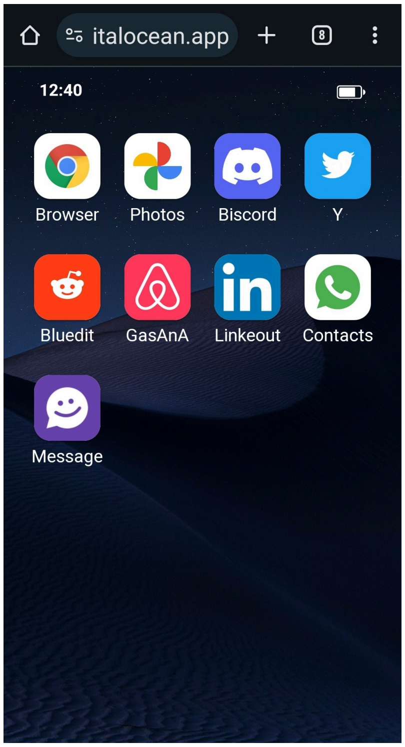


Figure 6: Picture of the Phone UI

When the webpage is first loaded, the phone initiates a connection between it and the computer. This connection is maintained as long as the webpage stays loaded in the player's phone browser. If the player ever reloads the page, the old connection will be discarded and a new one will be created.

To learn more about the development of the phone's web app, refer to the document written by our teammate Botao Han[1].

4.3 Unity Development

In this section, we will go further into our development process for the game on the computer. In the following subsections we discuss how we created many of our in-game systems that were essential for the game to run. These systems included the game state manager, the interaction system, the player movement, and many more.

[1] Phone Dev: https://github.com/nadiaz2/DROPTABLE_DigitalOcean/blob/main/Technical_Code.pdf

4.3.1 Game State Management

When we started the project we questioned how we wanted to manage the progression of the game. We decided to try and implement a game state system that would keep track of the state of the game at all times. This system would be able to be accessed by all scripts and could be used universally to keep track of what events needed to happen throughout the game. We also decided that each scene needed to have its own scene manager, which can control what is happening throughout the scene and has most of the information for the game's progression stored in one script. This way not only will the code be a lot more manageable and clean, but many different actions or events can be triggered using this one script. Throughout the game, there will always be two game states active. One of them is the overall game state of the game which is used to determine the progression between scenes and also keeps track of choices made throughout the game. Figure 7 shows the different paths the player can choose and how that affects the game state. The other is the scene-specific states, these scene states are only used in the scene they are currently on and will not be used in any other scenes.

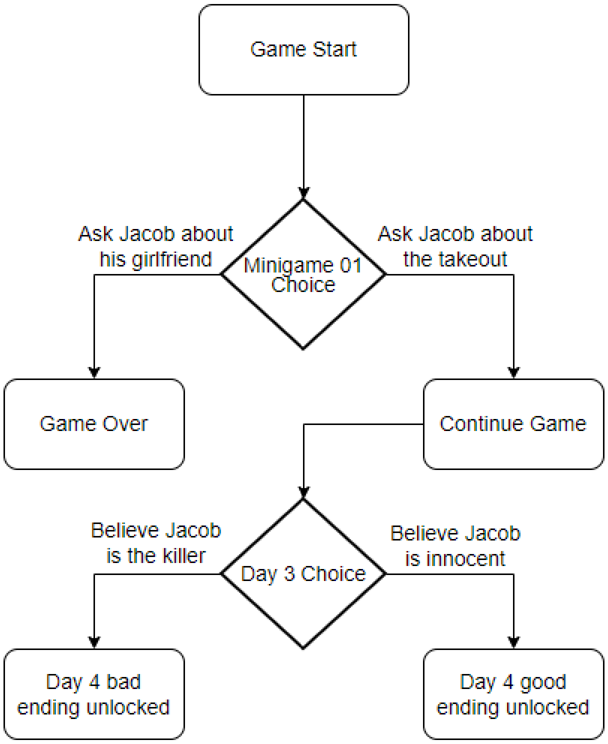
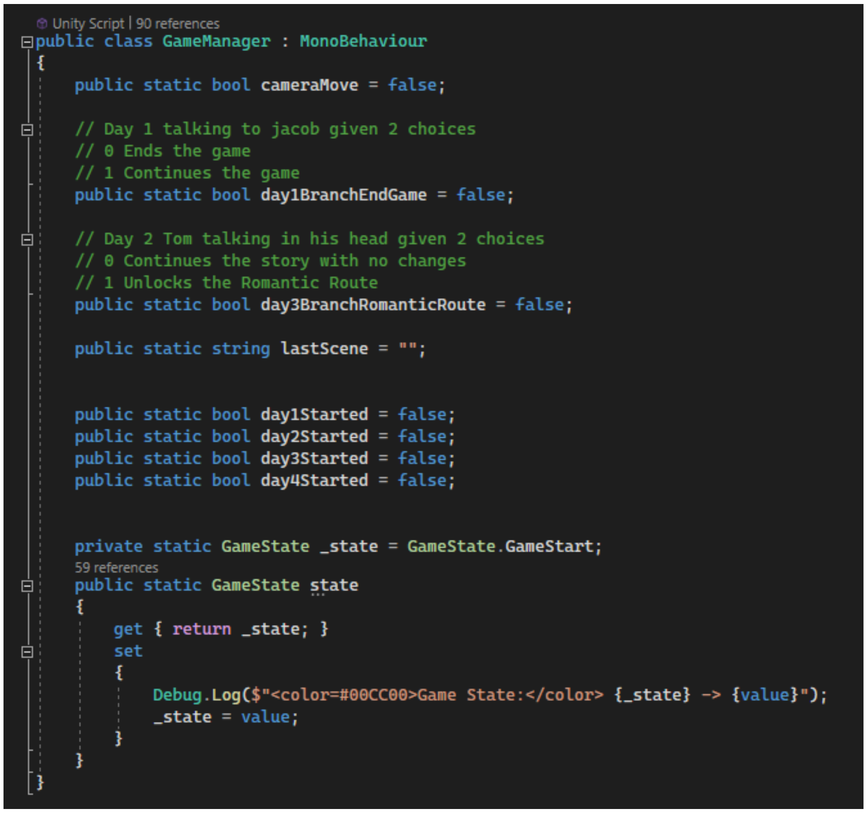

Game Start
Ask Jacob about his girlfriend
Minigame 01 Choice
Ask Jacob about the takeout
Game Over
Continue Game
Believe Jacob is the killer
Day 3 Choice
Believe Jacob is innocent
Day 4 bad ending unlocked
Day 4 good ending unlocked

Figure 7: Flowchart of the Global Game State

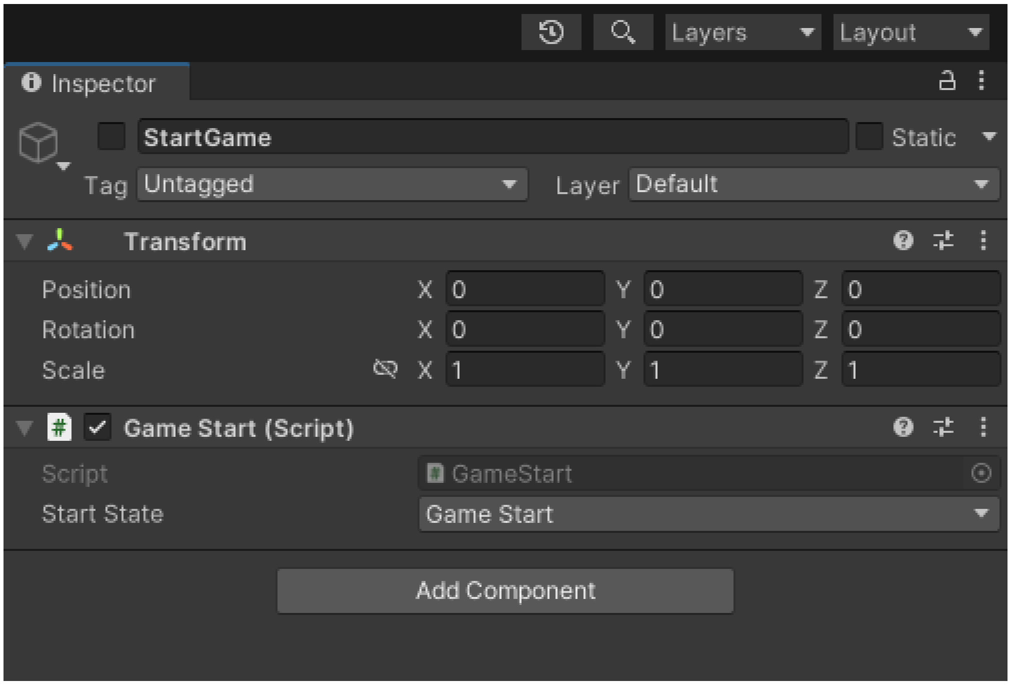
Since our game progresses throughout four days, the game state is also split into four days. Each day has its own game states that determine the progression of the game on that day. As we advance the game, different events will change the game state so that the new state can be used to set off new scripted events. This makes sure that the older events cannot be repeated since the game state condition is not being met anymore. A prominent way we used this switch of game state is when we are changing between scenes. Since scene states are not supposed to be called between scenes there is no way to know what part of the story you are in if you are revisiting a previously seen scene, which would have an older scene state. The solution we found was to have the overall game state be changed at the end of each scene so that when you switch scenes, the new scene will know that you are currently in the most present game state. This way the scene manager can use this information to set the current scene state to something relevant to the progression of the game.



```
Unity Script | 90 references
public class GameManager : MonoBehaviour
{
    public static bool cameraMove = false;

    // Day 1 talking to jacob given 2 choices
    // 0 Ends the game
    // 1 Continues the game
    public static bool day1BranchEndGame = false;

    // Day 2 Tom talking in his head given 2 choices
    // 0 Continues the story with no changes
    // 1 Unlocks the Romantic Route
    public static bool day3BranchRomanticRoute = false;

    public static string lastScene = "";


    public static bool day1Started = false;
    public static bool day2Started = false;
    public static bool day3Started = false;
    public static bool day4Started = false;


    private static GameState _state = GameState.GameStart;
    59 references
    public static GameState state
    {
        get { return _state; }
        set
        {
            Debug.Log($"<color=#00CC00>Game State:</color> {_state} -> {value}");
            _state = value;
        }
    }
}
```

Figure 8: GameManager Script

To make use of our game state system more effectively we created a GameStart script. The script created a section in the Unity Editor where we could select a specific game state and running the game would bring us to that specific game state and the corresponding scene. This script is mainly used for testing by giving us the ability to start the current session of the game in a specific chosen game state and scene. For example, if we were working on the second minigame, we could just set the game state in the GameStart script to the game state that corresponds to the state that was happening right before the minigame began. This way we will be able to start the minigame by going through the smallest amount of gameplay possible. This was crucial to help us save time and prevented us from having to go through the entire game every single time to test specific parts.



StartGame
Static
Tag Untagged
Layer Default

Transform

Position	X	0	Y	0	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

☑ Game Start (Script)

Script GameStart
Start State Game Start

Figure 9: Unity Inspector for the Object Holding the GameStart Script

During each scene, the scene manager will have their own scene state which can be used to trigger certain events that are crucial to advance the game and story. For example, certain events, such as the first phone minigame, can only be triggered if you are in the living room scene and the living room state is Day1JacobsBack. After the minigame is finished then the scene state will be set to something different so the minigame cannot be triggered again and the new scene state can play the player's dialogue. This is the basic structure that we use throughout the architecture of the game.

4.3.2 Connection Management

A single script is used to manage the connection on the computer side from start to finish. When the game loads into the title screen for the first time, Unity creates a game object that does not get removed when transitioning between scenes. On creation, this object creates a Socket.IO connection with the web server along with a unique identifier that represents this game session. This identifier is then shared with the web server for future use. For the phone to connect to the computer, a QR code is generated that contains a URL to the web server with the game session's unique identifier included as form data. The website that the phone loads uses that identifier to initiate a connection with the computer.

The computer keeps the phone synchronized with the current state of the game by sending it a message whenever the player begins a minigame. In order to prevent messages from being lost in transit, the computer expects a reply in acknowledgment for every message it sends. The computer keeps an active list of all messages that have not been acknowledged yet and resends messages every second.

In terms of the computer receiving messages from the phone, different scripts can subscribe to messages corresponding to a specific minigame. When subscribing, the other scripts pass in a callback function that takes the message information as an input parameter. When a message is received by the computer, the connection manager first filters out any acknowledgment messages. It then breaks the message into two parts – the minigame number and the message command. The connection manager checks if another script has subscribed to messages for that minigame. If so, it calls the respective callback function, passing the message command as the input.

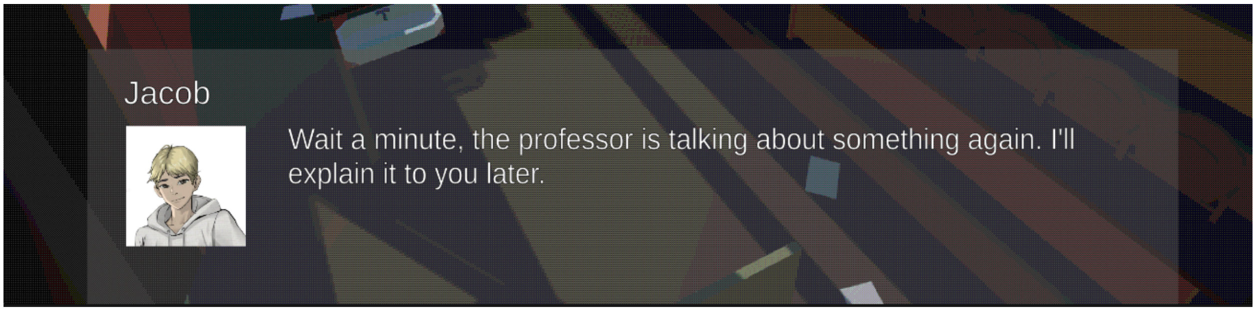
Managing the connection in one script and letting other scripts subscribe to events allowed us to implement new minigames easily and keep the networking code separate from the rest of the game code. This meant that whenever the networking code needed to be updated, no code in other scripts needed to be changed.

4.3.3 Interaction System

Our interaction system centered around the tags for game objects provided by Unity and implementations of a custom Interactable interface. Every object that was able to be interacted with in the game had to be tagged as Interactable as well as have its own script that extended the Interactable interface. Using this setup, an InterationController script was able to track the position of the player and determine which interactable object was closest to them within a certain range. A highlight would be added around that item and when the player interacts with it, the InteractionController calls the interaction function from that object's script. It can do this due to the object's script inheriting from the Interactable interface. This interface also allows objects to disable themselves for interaction, as well as change the text that appears for the interaction prompt.

4.3.4 Dialogue/Subtitle System

Since our game is a story-driven we needed to create a way to display dialogue when characters talked and to show the main character's internal dialogue. To achieve this we decided to build our dialogue system drawing from a YouTube channel called Brackeys (Brackeys, 2017). This channel had a specific tutorial that taught us the basics of how to create a simple dialogue system. We used this to create a template of the dialogue system which could display the name of the character speaking and the actual dialogue. We later decided to add a character portrait that displayed a drawing of the character speaking. This portrait will help the player recall who is speaking and remind them of previously seen characters.



Jacob

Wait a minute, the professor is talking about something again. I'll explain it to you later.

Figure 10: Dialogue UI

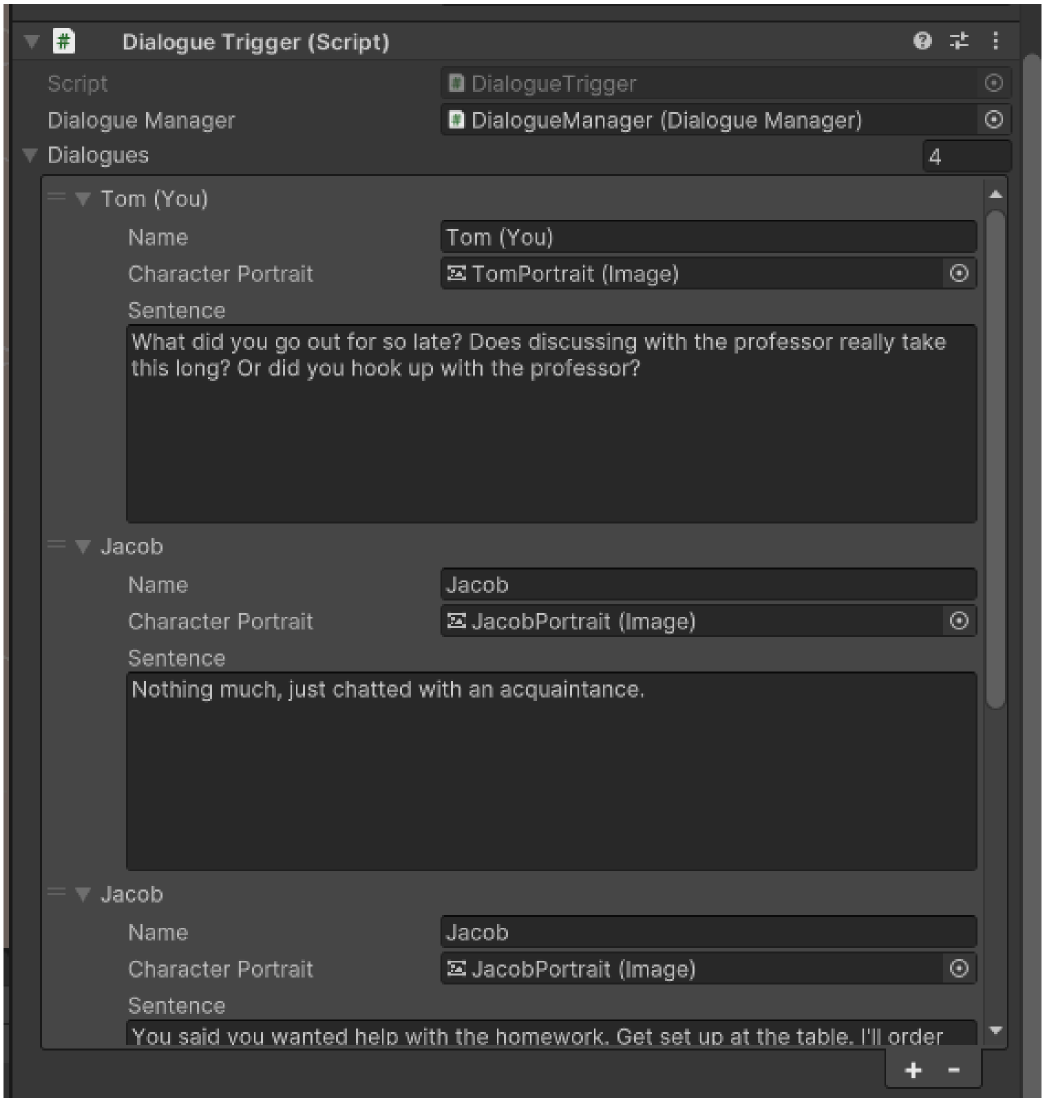
We have three separate scripts that work together for the dialogue system to work. We have the Dialogue Manager script, the Dialogue script, and the Dialogue Trigger script. The Dialogue script is just a script that will hold the name, dialogue, and character portraits. This script will be used in the Dialogue Manager when we need to access these variables.

Next is the most important script of the three mentioned, the Dialogue Manager. This is the script where all the different variables and objects are gathered together to be processed and displayed. In this script, there are functions that will start dialogue, continue dialogue, and end dialogue. We wanted to create a cool effect where it would seem like each word is being typed out when displaying the dialogue, so we used the Unity StartCoroutine function to make the effect happen.

The Dialogue Trigger is the last script that is used, this script is not only used to hold the contents of the Dialogue script but also is where we input all the different dialogues we need throughout the game. As shown in Figure 11, the Dialogue Trigger script is a component in the Unity Inspector and this is where we would manually input the three values that the Dialogue requires. Multiple of these dialogues can be created for each Dialogue Trigger to fit the needs of a single session of the character interaction.

The dialogue system is built on the UI and has animations that bring the dialogue up from below the screen when the characters are speaking and will bring it back down when the dialogue ends. Whenever the dialogue UI is up, we programmed it so that the player could not move their character so they would have full attention to the story.

Although we had a dialogue system that would be able to display interactions between the characters, we also needed a way to display internal dialogue and dialogue that seemed more like announcements or reminders. That is why we created a subtitle system that is also built off of the old dialogue system. This new subtitle system will not bring up the dialogue UI instead it will just display the dialogue at the bottom of the screen (Figure 12). This system is used when the main character Tom has internal dialogue or if the game needs to remind the character to do an action to progress the game. It is also used if there is a dialogue between characters that do not require the dialogue UI. This situation only happens if the interactions are short or if the game wants the player to still have the ability to move and interact with other objects in the scene.



Dialogue Trigger (Script)

Script: DialogueTrigger

Dialogue Manager: DialogueManager (Dialogue Manager)

Dialogues: 4

Tom (You)

Name: Tom (You)

Character Portrait: TomPortrait (Image)

Sentence: What did you go out for so late? Does discussing with the professor really take this long? Or did you hook up with the professor?

Jacob

Name: Jacob

Character Portrait: JacobPortrait (Image)

Sentence: Nothing much, just chatted with an acquaintance.

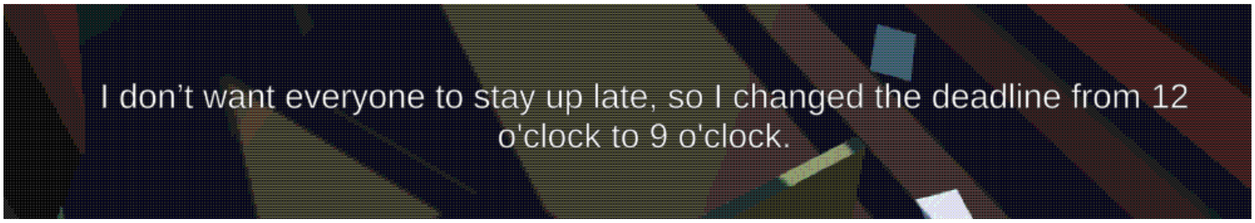
Jacob

Name: Jacob

Character Portrait: JacobPortrait (Image)

Sentence: You said you wanted help with the homework. Get set up at the table. I'll order

Figure 11: Dialogue Trigger Component in Unity Inspector

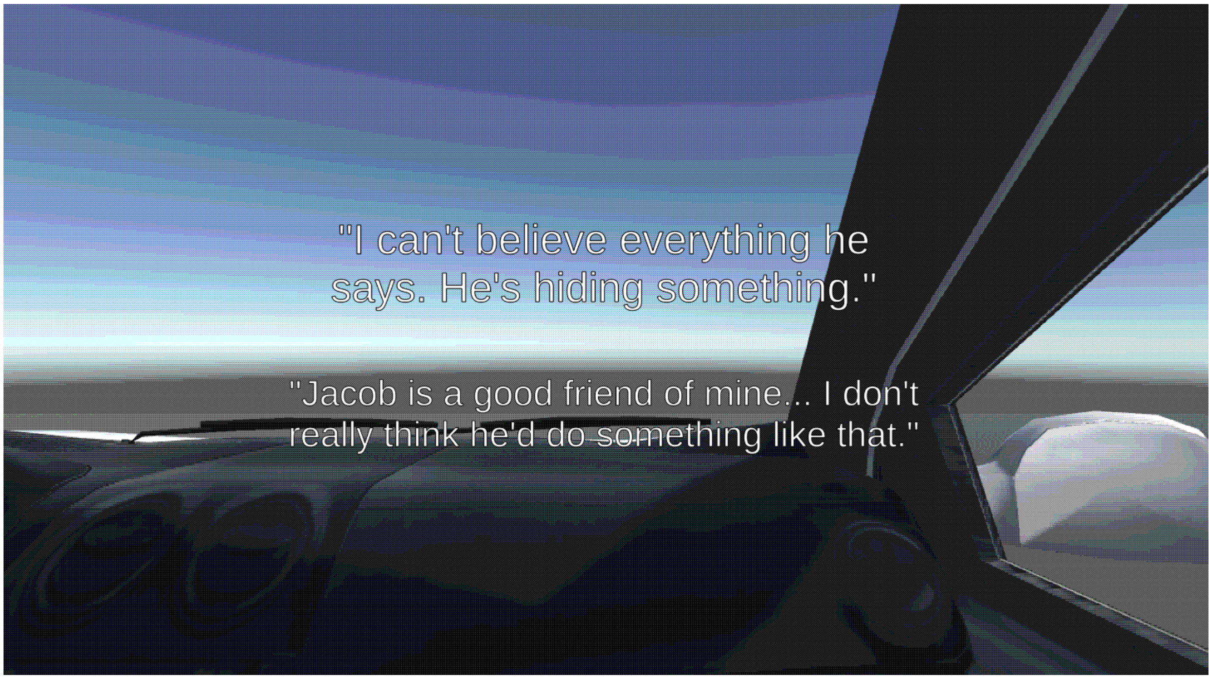


I don’t want everyone to stay up late, so I changed the deadline from 12 o'clock to 9 o'clock.

Figure 12: Subtitle UI

4.3.5 Choice System

As the player progressed through the game and played the mini-games, we also wanted to include a way for the player to change their fates and the ending of the game. Our writer and designer decided that they wanted to achieve this by adding different decisions that the player could make during certain moments in the game. These decisions will be very important and will change the path and end of the game. For the implementation, the simplest way we found was to build a similar script structure as the dialogue and subtitle system, as shown in Figure 13.



"I can't believe everything he says. He's hiding something."

"Jacob is a good friend of mine... I don't really think he'd do something like that."

Figure 13: Choice Menu

4.3.6 Player Movement

There are three types of scenes that we have in our game: 3D third-person, 3D horizontal panels, and first-person. The most common perspective in our game is 3D third-person, this perspective's movement directions will be aligned with the camera axis. This means that where we place the camera will determine how the player will move in each scene. We will dive further into this concept in the next section: Camera Control and Placement. The next perspective we had was the 3D horizontal panels, in these instances, we had the same basic controls as the 3D third-person perspectives. Since the scenes that require this perspective only needs the player to move horizontally, the only difference is that we removed forward and backward movement, so the player can only move left or right. In the first-person perspective, the player will be looking through the eyes of Tom and will only have the ability to look around and interact with objects in their vicinity. Movement has been removed because all of the first-person instances involve Tom sitting and does not require the player to move around.

4.3.7 Camera Control and Placement

As part of the design of the game, we wanted the camera control and placement to depend on which scene the player is in. In most scenes, the camera would be placed in third-person and the player would not be able to directly control the camera. However, in the scenes where the camera is in first-person, the player would be able to rotate the camera and look around. To achieve this, we created two complementary camera systems.

In the third-person scenes, the camera position is fixed, and the player can move through the scene freely. In large scenes, the game can track the player's position and change the camera perspective if the player moves far enough away from the camera. This is done through the use of hidden triggers that the player character will walk through in the scene. When the camera moves, this also changes the movement direction in order to keep it in line with the camera's orientation. By moving the camera placement, we are able to keep the player character visible throughout the scene and make it easier for the player to see different areas in the scene.

In first-person instances, the camera position never moves, since the player is not able to move their character's position. Instead, the player is sometimes able to control the camera orientation by using their mouse to look around. This is done by attaching the camera to a game object and rotating that game object when the player moves the mouse.

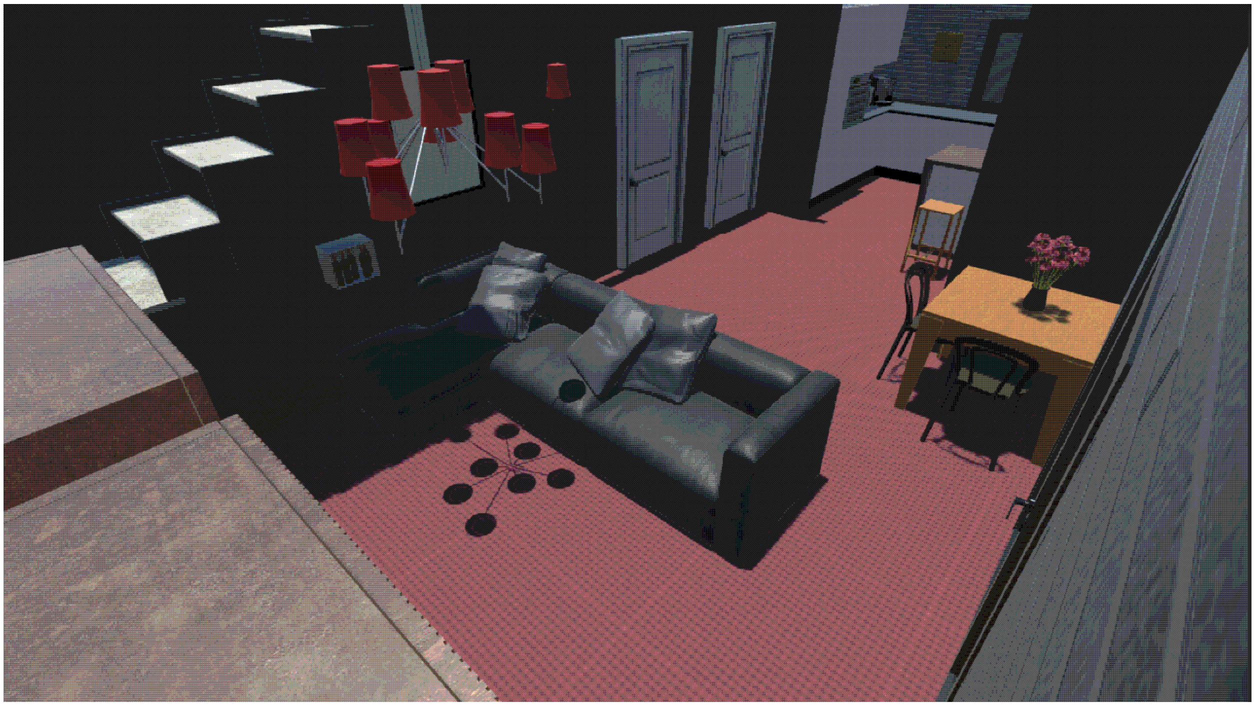


Figure 14: House Entrance Camera Position

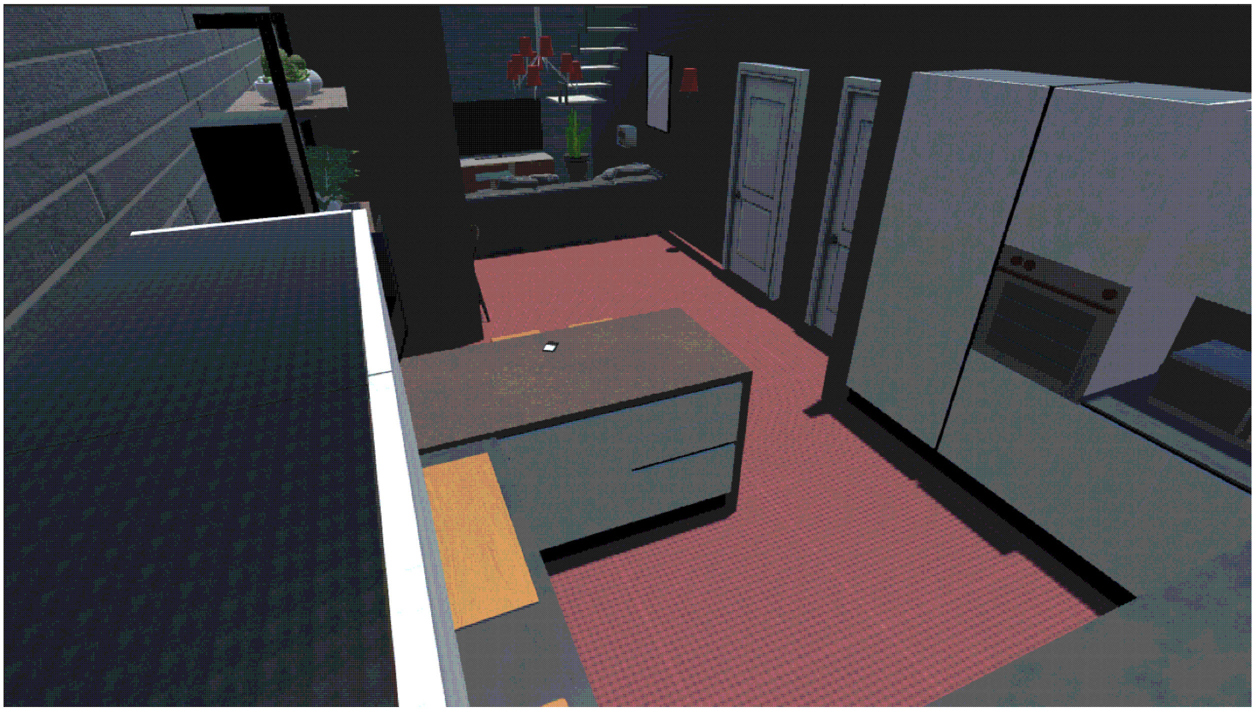


Figure 15: House Kitchen Camera Position

4.4 Challenges

By and large, the biggest challenge that we encountered during development was trying to create a way to connect the phone and computer in a way that was both low latency and allowed us to use the sensors in the browser. Addressing these challenges required us to redesign how our networking architecture was structured multiple times.

Initially, we tried to set up the Unity game to host a web server locally, with 2-way communication being enabled via Socket.IO. There were multiple problems that we encountered in this attempt. The first of these problems was that there is no official Socket.IO server library for C#, which is the programming language that Unity uses. We tried to solve this by instead launching a Node.js server from Unity and then having both the phone and computer connect to that. This is when we discovered a second problem. In order for JavaScript to be allowed access to certain sensors in the browser, such as device orientation, the user needs to be connected via HTTPS (“Window: deviceorientation event”, 2023). Due to this, we would not be able to host the web server locally on the computer that was running the game without introducing significant additional complexity, such as signed SSL certificates for individual players.

To solve this, we transitioned our web server onto DigitalOcean. By using DigitalOcean, we were able to have our players connect to the website via HTTPS, thereby allowing us to use the sensors that we wanted. However, making the phone and computer communicate through a server that is hosted on the open internet means that messages would take much longer to travel between the two. This was undesirable for us since it meant that players could feel an unnecessary latency when playing the game. It also meant that the web server would have to maintain all active connections of people who were playing the game, causing there to be a scaling issue if many players wanted to play at the same time.

Both of these problems were solved through the use of WebRTC. WebRTC allowed us to establish a peer-to-peer connection between the phone and the computer, cutting down on latency and allowing the webserver to scale much better.

5. Art

We wanted our game to have a pixel art style in order to create a mysterious atmosphere similar to “Who’s Lila?”. However, we did not have a pixel artist on our team, so we opted to use a shader that downscaled the resolution, achieving the same effect. This means that we were able to make use of standard 3D models and animations. This reduced the amount of custom artwork needed for this project. As the original team contains only technological students, we chose to use outsourced character design and character modeling, while finding other on-campus art students for object modeling. Figures 16 and 17 below show concept art that helped guide our art style and character design.

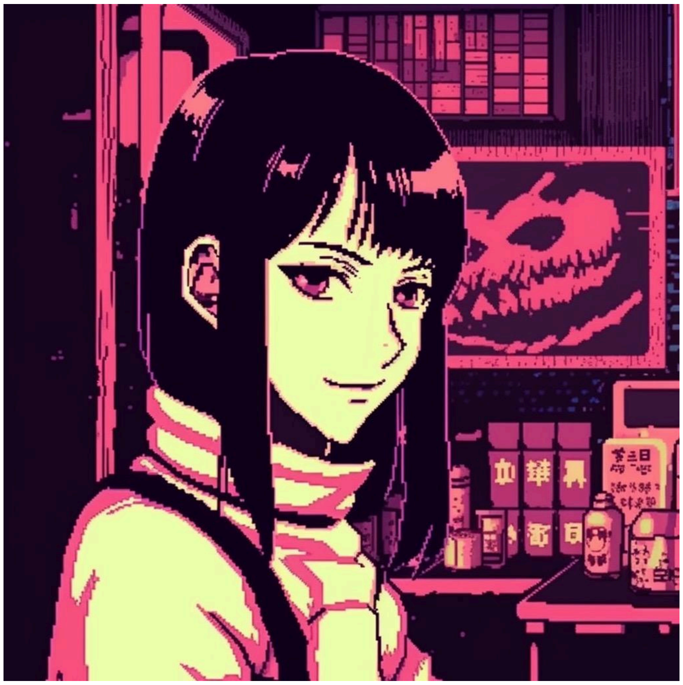


Figure 16: Concept Art of Overall Art Style (generated by MidJourney Bot)

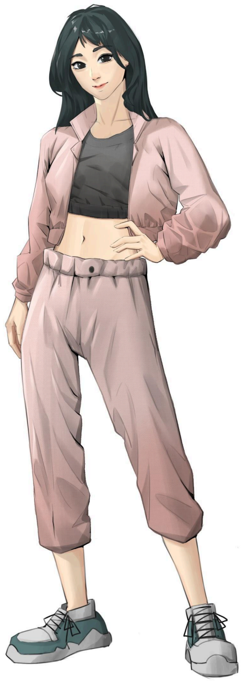


Figure 17: Character Design Made by an Outsourced Artist

6. Evaluation

In order to evaluate our project and determine if we succeeded in our design goals, we performed playtesting to gather player feedback. After creating a demo of the game, we sourced player testers from current WPI students. Before starting, each playtester was given a brief summary of the game's concept and was asked to sign a consent form (Appendix B). IMGD playtesting credit was also available for the playtester if they needed it. The playtesters were then asked to play through the demo fully, with each session taking about 30–45 minutes to complete. We used a think-aloud protocol to hear the active thoughts of the playtesters and took notes throughout the playtest in order to document each player's thoughts and actions throughout the game. After the playtesters finished playing the demo, we asked them to fill out an anonymous survey via Google Forms (Appendix C). This mix of qualitative and quantitative data gathering allows us to thoroughly understand how the players felt about our game and judge our own work. The specific procedures and steps for our playtesting are recorded in Appendix D.

6.1 Results

In total, we were able to perform playtesting sessions for six playtesters, each of which gave us their own perspectives and experiences while playing through our game. Our biggest takeaways had to do with the playtesters' overall experience, engagement with the story, and interaction between the phone and PC. For the full survey results see Appendix E.

6.1.1 Survey Responses

Overall according to our survey, most of the playtesters had a good experience with *Echoes Through the Screen* (Figure 18). This is a good result because it means that our game in general has a solid foundation. In one of our survey questions asking the playtester what stood out to them the most, one of our playtesters responded that they were very engaged with the environment and the small details throughout the game.

> "I really liked connecting the dots between sharp objects in the trash bags and then realizing that Jacob definitely hid the murder weapon there. I also was baffled when Tom died during the night, that helped me understand that Jacob is for sure doing shady stuff."

This indicates that we were able to achieve interesting game design and pique the playtester's curiosity.

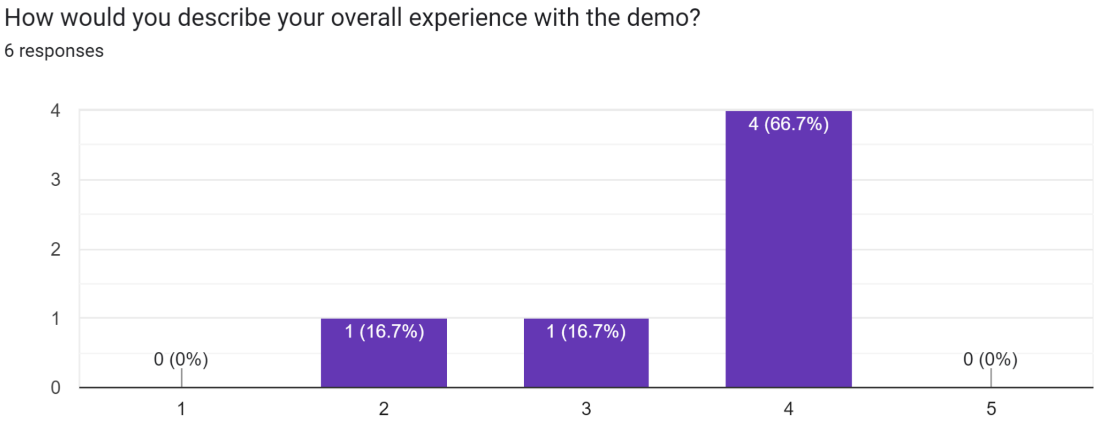


Figure 18: Survey Results - Overall Experience

Amongst all the playtesters, around two-thirds of them reported that they were engaged with the storyline (Figure 19). Many playtesters were interested in the mystery and expressed interest in how the story would continue after the demo. Despite this, not all the feedback was strictly positive. Instead, some found that the writing was too stiff and others found the story too linear and did not have much depth. This was partially due to poor translation of the story in Mandarin, the original language it was written in.

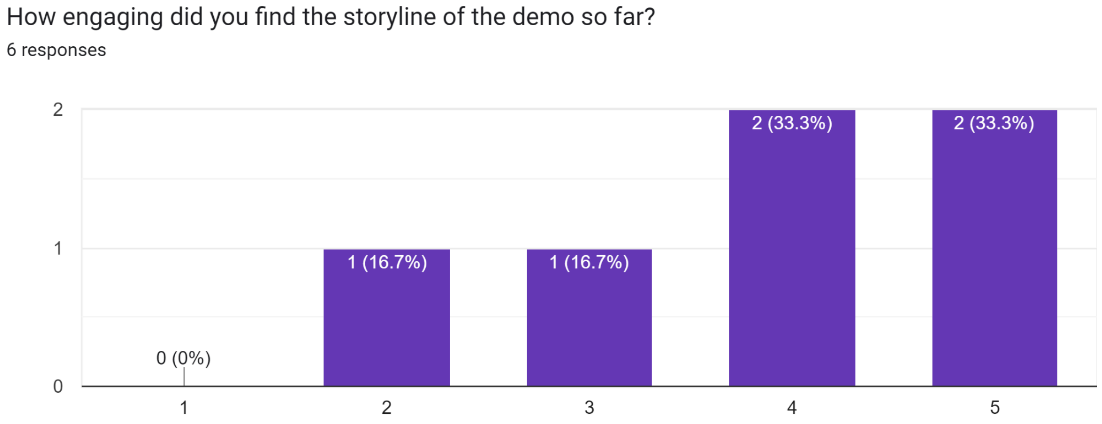


Figure 19: Survey Results - Storyline Engagement

When asked about their experience with the interaction between the phone and PC, we received very divided responses. While some playtesters enjoyed the interaction with their physical phone, others felt that switching between the computer and the phone detracted from the gameplay experience (Figure 20). As this was a key aspect of our game, this was a surprising result to see. From our question that asked about the uncertainties in setting up or using the phone-PC connection feature, we got feedback that suggested that switching between the phone and PC was not immersive enough and felt like it was forced into the game. It is hard to tell if we simply did not do enough to engage the player with the phone or if the use of a real-world phone breaks immersion in the game by forcing them to switch focus.

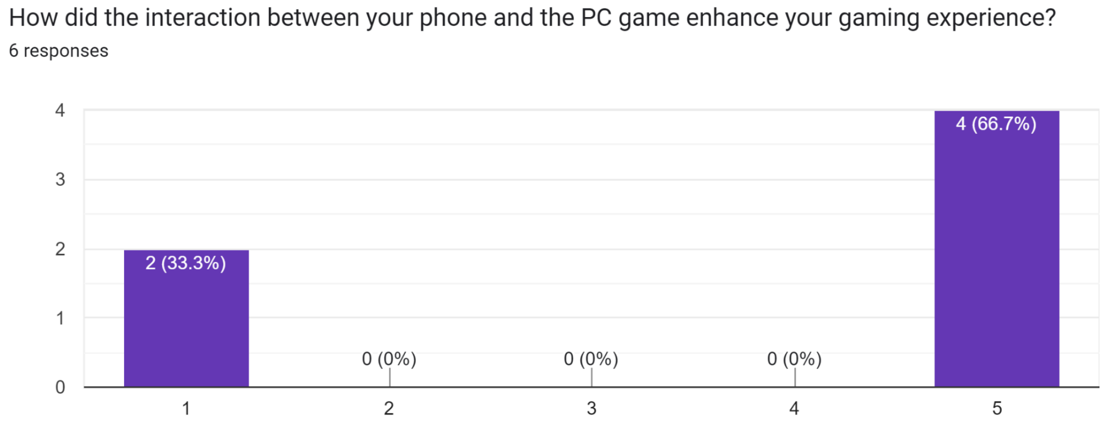


Figure 20: Survey Results - Phone and PC Interaction

As we performed these playtests we observed some crucial aspects of our connection that we believed could have been a major immersion breaker. The most important was having our playtesters rescan the QR code every single time they needed to use their phone and making them close the tab to the website every single time they finished interacting with the phone. This was later improved upon and we were able to make it so the player only had to scan the QR code once at the first minigame. They also did not have to close the tab of the website after every phone interaction. While we believe this made the phone interaction better, it is difficult to conclude if playtesters felt the same due to a lack of data.

6.1.2 Think-aloud Observations

From observing the playtesters play the game, we noticed three major patterns emerge. First, we noticed that players sometimes had difficulty navigating the scenes due to there not being enough contrast between objects and it not being obvious if it was possible to interact with an object at first glance.

The second pattern we noticed was that many of the playtesters did not understand how they were meant to interact with the phone. Sometimes, playtesters were unsure if they needed to interact with the phone or the computer to proceed. Other times, playtesters were not able to figure out what action the game expected them to perform on the phone next. A clear example of this was during the minigame on day 1. This is the first time the player interacts with the phone and the game expects the player to explore the interaction. The game proceeds when the player clicks on a specific photo in the album. Many of the playtesters did not figure out that they were meant to click on the photos and, therefore, needed instruction from one of our group members on how to proceed.

Finally, the last pattern we observed during playtests was that some playtesters felt the subtitles automatically continued too quickly. This led to them feeling rushed to read through them and sometimes missing information if they were unable to finish reading in time. For a full look at the notes we took during playtesting, see Appendix F.

7. Conclusion

As mentioned earlier, the most difficult part about developing *Echoes Through the Screen* was figuring out a system to connect the phone and computer together that allowed us to use the web API for sensors and kept the latency of messages low. This required us to redesign our networking code multiple times and took a significant amount of effort. However, now that we have largely solved this problem, our team sees this method of using a cell phone in a computer game as promising. This format holds a lot of potential due to the flexibility of the browser as a development platform and multiple playtesters expressed excitement in the idea.

From a developmental perspective, we learned a lot about the difficulties of developing two different applications simultaneously that needed to communicate with each other across programming languages. One of the biggest challenges that we encountered was with testing our code. Testing any of the networking or minigame code required both sides to have the feature complete. This meant that one developer's work would sometimes be blocked by another developer's, causing frustration. Any development that did not rely on the other platform, though, went fairly smoothly. Our mobile developers and Unity developers were both able to create systems for themselves that helped them add on to what they already had and made their development process smoother.

We would like to once again thank everyone who was involved in this project. This includes the modelers who helped us make all the objects and characters for our game, as well as our ISP members who were a part of our team from start to finish, helping us to create the game all the way from story writing to mobile development. Without your help, we would have not been able to accomplish what we did.

References:

Acerola, director. *Color Quantization and Dithering*. 2022. *https://www.youtube.com/watch?v=8wOUe32Pt-E*.

Acerola. “GarrettGunnell/Post-Processing: My Unity post processing pipeline and shaders.” *GitHub*, https://github.com/GarrettGunnell/Post-Processing. Accessed 24 April 2024.

Arvtesh. “Arvtesh/UnityFx.Outline: Screen-space outlines for Unity3d.” *GitHub*, https://github.com/Arvtesh/UnityFx.Outline. Accessed 24 April 2024.

Brackeys. “How to make a dialogue system in Unity.” *Youtube*, 23 July 2017, https://youtu.be/_nRzoTzeyxU?si=6ID5vpM3udtExOq5. Accessed 23 April 2024.

Brazie, Alexander. “Video Game Mechanics: A Beginner's Guide (with Examples).” *Game Design Skills*, https://gamedesignskills.com/game-design/video-game-mechanics/. Accessed 24 April 2024.

Colestia. *A Hand With Many Fingers*. Colestia, 9 June 2020, https://colestia.itch.io/a-hand-with-many-fingers.

Fictiorama Studios, and BadLand Games Publishing S.L. *Do Not Feed the Monkeys*. Alawar, 23 October 2018, https://store.steampowered.com/app/658850/Do_Not_Feed_the_Monkeys/.

Flying Helmet Games. *Eon Altar*. Flying Helmet Games, 22 June 2016, https://store.steampowered.com/app/382050/Eon_Altar/.

Francis, Bryant. “Unity introducing new fee attached to game installs.” *Game Developer*, 12 September 2023, https://www.gamedeveloper.com/business/unity-to-start-charging-fee-pegged-to-game-installs. Accessed 23 April 2024.

Garage Heathen. *Who's Lila?* IndieArk, 23 Feburary 2022, https://store.steampowered.com/app/1697700/Whos_Lila/#:~:text=Who's%20Lila%20is%20a%20point,one%20to%20see%20her%20last.

Geography of Robots. *NORCO*. Raw Fury, 24 March 2022, https://store.steampowered.com/app/1221250/NORCO/.

"Introduction | Socket.IO." *Socket.IO*, 5 April 2024, https://socket.io/docs/v4/. Accessed 24 April 2024.

itisnajim. "itisnajim/SocketIOUnity: A Wrapper for socket.io-client-csharp to work with Unity." *GitHub*, 2021, https://github.com/itisnajim/SocketIOUnity. Accessed 24 April 2024.

"Jackbox Games." *Jackbox Games - Home*, https://www.jackboxgames.com/. Accessed 24 April 2024.

Jahn, Michael. "micjahn/ZXing.Net: .Net port of the original java-based barcode reader and generator library zxing." *GitHub*, https://github.com/micjahn/ZXing.Net. Accessed 24 April 2024.

Kei, Sanbe. "Erased (Manga)." *Wikipedia*, Wikimedia Foundation, 11 Apr. 2024, en.wikipedia.org/wiki/Erased_(manga).

Lucas Pope. *Return of the Obra Dinn*. 3909, 18 October 2018, https://store.steampowered.com/app/653530/Return_of_the_Obra_Dinn/.

Minato, Kanae, and Stephen Snyder. *Confessions*. Mulholland, 2014.

Mountains. *Florence*. Annapurna Interactive, 13 February 2020, https://store.steampowered.com/app/1102130/Florence/.

Octavi Navarro. *Midnight Scenes: The Nanny*. Octavi Navarro, 27 October 2021, https://store.steampowered.com/app/1771070/Midnight_Scenes_The_Nanny/#:~:text=Released,a%20new%20dimension%20of%20horror.

panstasz. *WORLD OF HORROR*. Ysbryd Games, 19 October 2023, https://store.steampowered.com/app/913740/WORLD_OF_HORROR/.

Pixel"Amaya, Daisuke ``. "Why Pixel Art Games Have Become Widely Popular." *RocketBrush Studio*, 8 December 2021, https://rocketbrush.com/blog/pixel-art-games-popular. Accessed 24 April 2024.

Rusty Lake. *The White Door*. Second Maze, 9 January 2020, https://store.steampowered.com/app/1145960/The_White_Door/.

Shynkarenka, Ivan. "chronoxor/NetCoreServer: Ultra fast and low latency asynchronous socket server & client C# .NET Core library with support TCP, SSL, UDP, HTTP, HTTPS, WebSocket protocols and 10K connections problem solution." *GitHub*, https://github.com/chronoxor/NetCoreServer. Accessed 24 April 2024.

STA. "sta/websocket-sharp: A C# implementation of the WebSocket protocol client and server." *GitHub*, https://github.com/sta/websocket-sharp. Accessed 24 April 2024.

2-REC. "2-REC/unity-nodejs: Start a JavaScript script with Node.js from a Unity project." *GitHub*, https://github.com/2-REC/unity-nodejs. Accessed 24 April 2024.

Unity-Technologies. "Unity-Technologies/com.unity.webrtc: WebRTC package for Unity." *GitHub*, 2019, https://github.com/Unity-Technologies/com.unity.webrtc. Accessed 24 April 2024.

"WebRTC." *WebRTC.org*, https://webrtc.org/. Accessed 24 April 2024.

"Window: deviceorientation event - Web APIs | MDN." *MDN Web Docs*, 15 December 2023, https://developer.mozilla.org/en-US/docs/Web/API/Window/deviceorientation_event. Accessed 24 April 2024.

Zhewanzhou. *Ban Cheng*. Qingdao Chu Ban She, 2017.

Appendices:

Appendix A: Computer-Phone Connection API

Message Format

- ➢ "MINIGAME NUMBER+DASH+ACTION NAME"
- ➢ Example: "01-FOUNDPHOTO" means mini-game number one, the player just found the required photo

Connecting from the Phone

- ➢ Establish the Connection
  - ○ Add the following code to the header of the page’s HTML file:
    - ■ <script src="/socket.io/socket.io.js"></script>
    - ■ <script type="module" src="/js/connection.js"></script>
  - ○ These scripts will automatically set up and maintain the connection on their own. It should not be necessary to edit the connection.js file. This allows for any necessary edit to the file to be an easy drag and drop replacement. The interface for sending and receiving messages via this connection will stay consistent.
- ➢ Sending Messages
  - ○ In any script that you wish to send a message from, add the following line of code to the top of the file:
    - ■ import connection from "/js/connection.js"
  - ○ After doing this, any time you want to send a message to Unity, simply call:
  - ○ connection.send(string)
  - ○ Note that in order to use import, the script must be marked as type=”module” when importing it into the HTML.
- ➢ Receiving Messages
  - ○ If you wish to receive messages that are sent from Unity, make sure the following line of code in included at the top of the script file:
    - ■ import connection from "/js/connection.js"

- Then, supply a function that will be called when a message is received to the onUnityMessage property of the connection object. This function should take in one parameter which will hold the message that was received. Ex:
  - connection.onUnityMessage = function(message) {
  - console.log(message)
  - }
- This allows the connection.js file to ensure the connection is valid and reinitiate it if necessary before the message is sent. Note that in order to use import, the script must be marked as type=”module” when importing it into the HTML.

Expected Messages

- Minigame 01
  - START - Sent from computer. This minigame is starting.
  - FOUNDPHOTO - Sent from phone. The player has found the required photo on the phone.
  - FINISH - Sent from phone. This minigame is finished.
- Minigame 02
  - START - Sent from computer. This minigame is starting.
  - DIALOG - Sent from computer. Open the announcement on phone
  - ALBUM - Sent from phone. The player opened album
  - BROWSER - Sent from phone. The player opened browser(every time)
  - CHATAPP - Sent from phone. The player opened the chat app(for the first time)
  - FOUNDCHAT - Sent from phone. The player opened key dialogue.
  - DIALOGCLOSED - Sent from phone. The player closed the announcement.
  - FINISH - Sent from phone. This minigame is finished.
- Minigame 03 (Crack Password)
  - START - Sent from computer. This minigame is starting.
  - CHATAPP - Sent from phone. The player opened the chat app(for the first time)
  - KEYCHAT - Sent from phone. The player opened key dialogue.
  - FOUNDPHOTO - Sent from phone. The player has found the required photo on the phone.

    - FINISH - Sent from phone. This minigame is finished.

- Minigame 04 (Phone Call)
    - START - Sent from computer. This minigame is starting.
    - CALL - Sent from phone. The player pressed the call button.
    - POCKET - Sent from phone. The player flipped the phone.
    - FINISH - Sent from phone. This minigame is finished.

Appendix B: Playtesting Consent Form

Informed Consent Agreement for Participation in a Research Study

Investigator: Charles Robert, Zesheng Chen, Shiming De, Nelson Diaz, Botao Han, Bright Lin

Contact Information: cdroberts@wpi.edu, zchen13@wpi.edu, sde@wpi.edu, nadiaz2@wpi.edu, bhan@wpi.edu, blin2@wpi.edu

Title of Research Study: DROPTABLE Game Creation

Introduction: You are being invited to participate in a research study. Before you agree, however, you must be fully informed about the purpose of the study, the procedures to be followed, and any benefits, risks or discomfort that you may experience as a result of your participation. This form presents information about the study so that you may make a fully informed decision regarding your participation.

Purpose of the study: The purpose of this study is to collect advice on the art, music, and gameplay of the game created by the game, as well as improve the game experience through user's feedback.

Procedures to be followed: 1. The researcher will introduce the procedure to the test subject. 2. The researcher responses for corresponding subjects will read and introduce the consent form to test subjects and ask them to sign it if they agree with them. 3. The researcher will collect the consent form and preserve them with care. 4. The researcher will need to introduce the DROPTABLE game to the test subject. 5. The researcher will ask the test subject to play through the whole game. 6. The test subject can explore if they wish to. 7. Then, the researcher will ask the test subject to fill out a google form containing questions about advice on the game.

Risks to study participants: Participants will not face any possible physical harm during the course. The participant can also stop the study at any time during the whole process if one feels uncomfortable toward anything in the environment.

Benefits to research participants and others: This Study will not give participants any general benefit.

Record keeping and confidentiality: Researchers will store all information in a private google drive which only game developers have access to. All personal information data will not be recorded.

In addition, records of your participation in this study will be held confidential so far as permitted by law. However, the study investigators and, under certain circumstances, the Worcester Polytechnic Institute Institutional Review Board (WPI IRB) will be able to inspect and have access to confidential data that identify you by name. Any publication or presentation of the data will not identify you.

Compensation or treatment in the event of injury: There is no foreseeable risk of injury associated with this research study. Nevertheless, you do not give up any of your legal rights by signing this statement.

For more information about this research or about the rights of research participants, or in case of research-related injury, contact the Investigator listed at the top of this form. You may also contact the IRB Manager (Ruth McKeogh, phone 508 831-6699, email irb@wpi.edu) and/or the Human Protection Administrator (Gabriel Johnson, phone 508-831-4989, email gjohnson@wpi.edu).

Your participation in this research is voluntary. Your refusal to participate will not result in any penalty to you or any loss of benefits to which you may otherwise be entitled. You may decide to stop participating in the research at any time without penalty or loss of other benefits. The project investigators retain the right to cancel or postpone the experimental procedures at any time they see fit.

By signing below, you acknowledge that you have been informed about and consent to be a participant in the study described above. Make sure that your questions are answered to your satisfaction before signing. You are entitled to retain a copy of this consent agreement.

___________________________________ Date: ___________________

Study Participant Signature

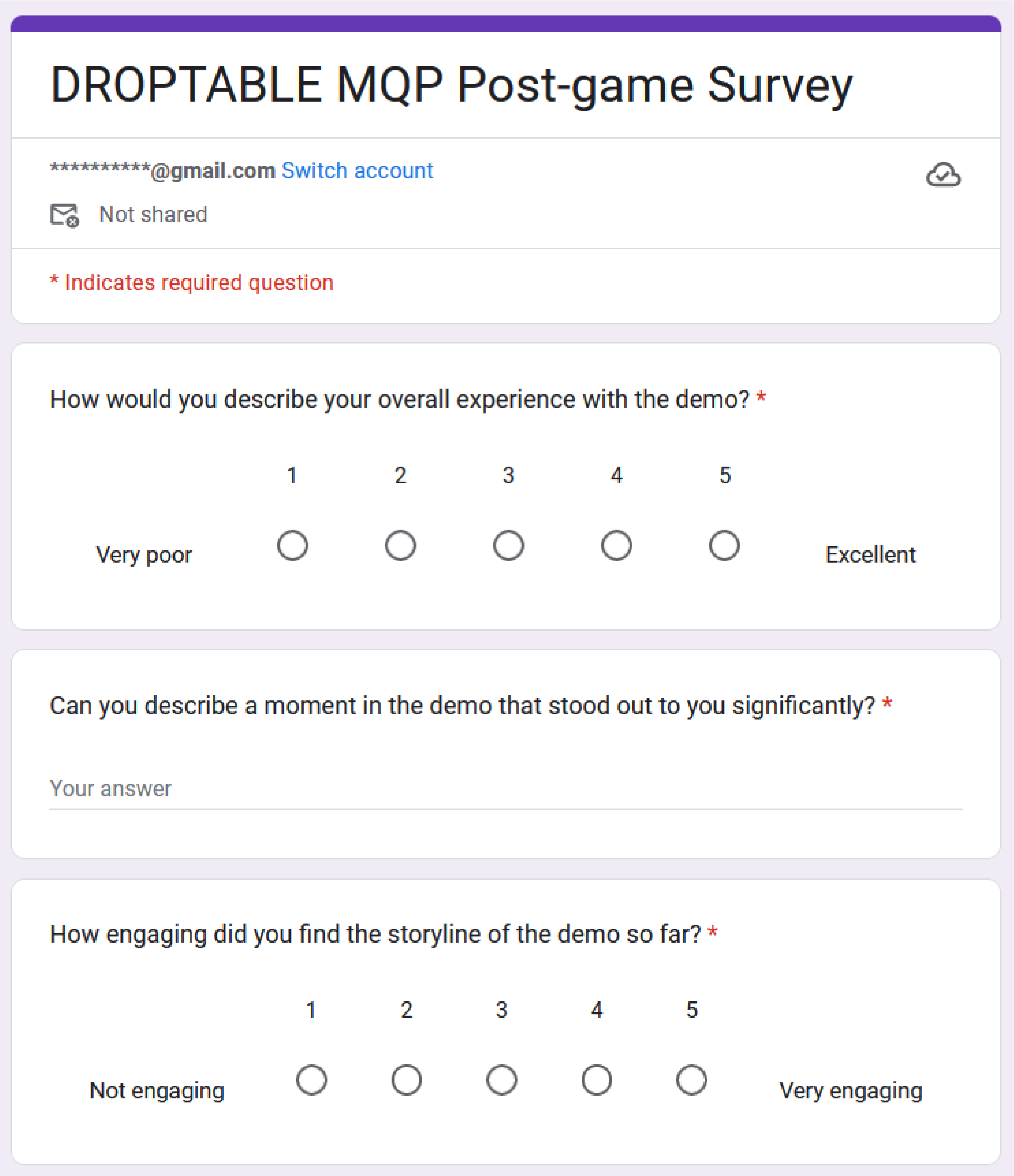
____

Study Participant Name (Please print)

___________________________________ Date: ___________________

Signature of Person who explained this study

Appendix C: Playtesting Google Form



DROPTABLE MQP Post-game Survey

**********@gmail.com Switch account

Not shared

* Indicates required question

How would you describe your overall experience with the demo? *

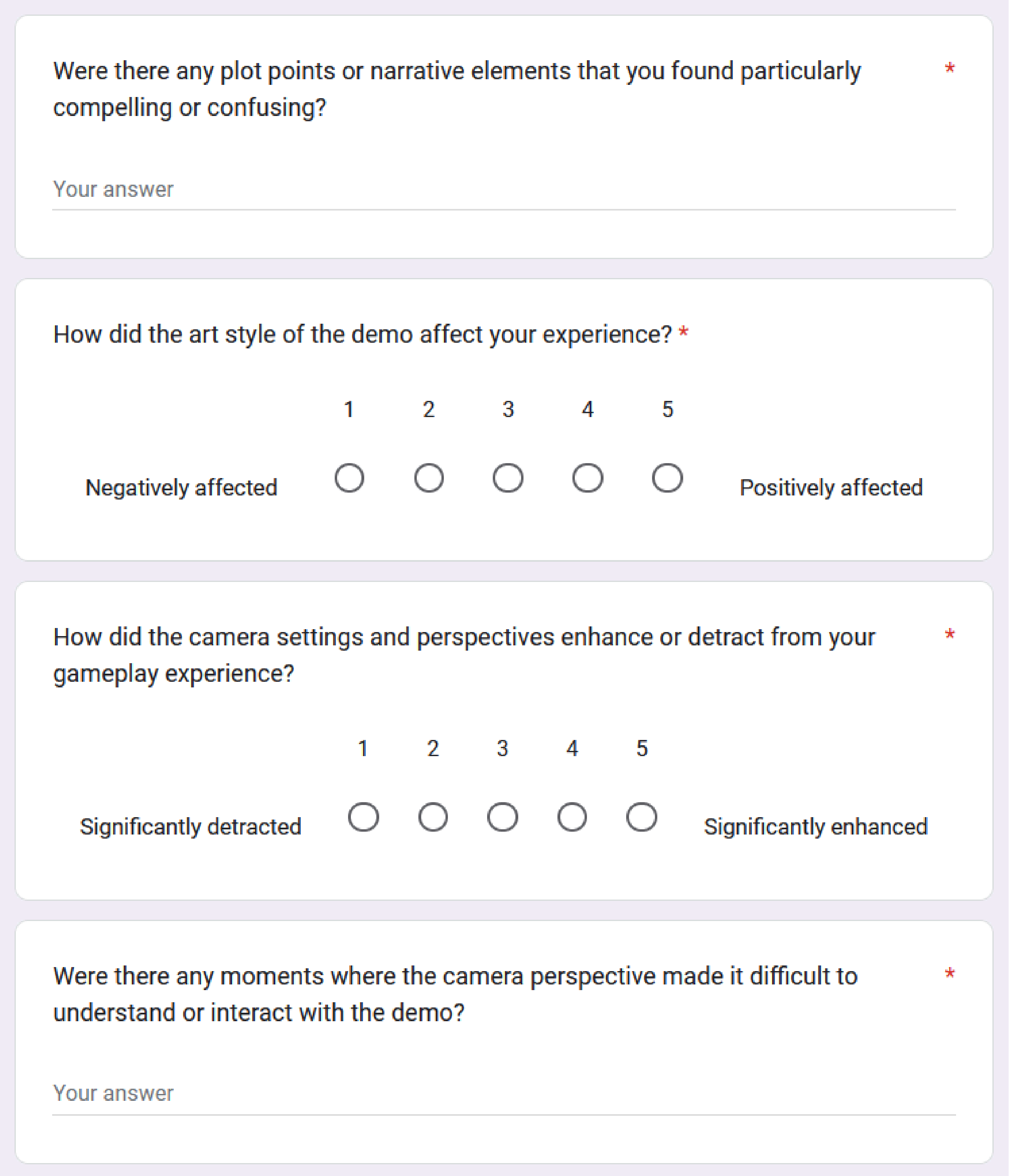
	1	2	3	4	5	
Very poor	○	○	○	○	○	Excellent

Can you describe a moment in the demo that stood out to you significantly? *

Your answer

How engaging did you find the storyline of the demo so far? *

	1	2	3	4	5	
Not engaging	○	○	○	○	○	Very engaging



Were there any plot points or narrative elements that you found particularly compelling or confusing? *

Your answer

How did the art style of the demo affect your experience? *

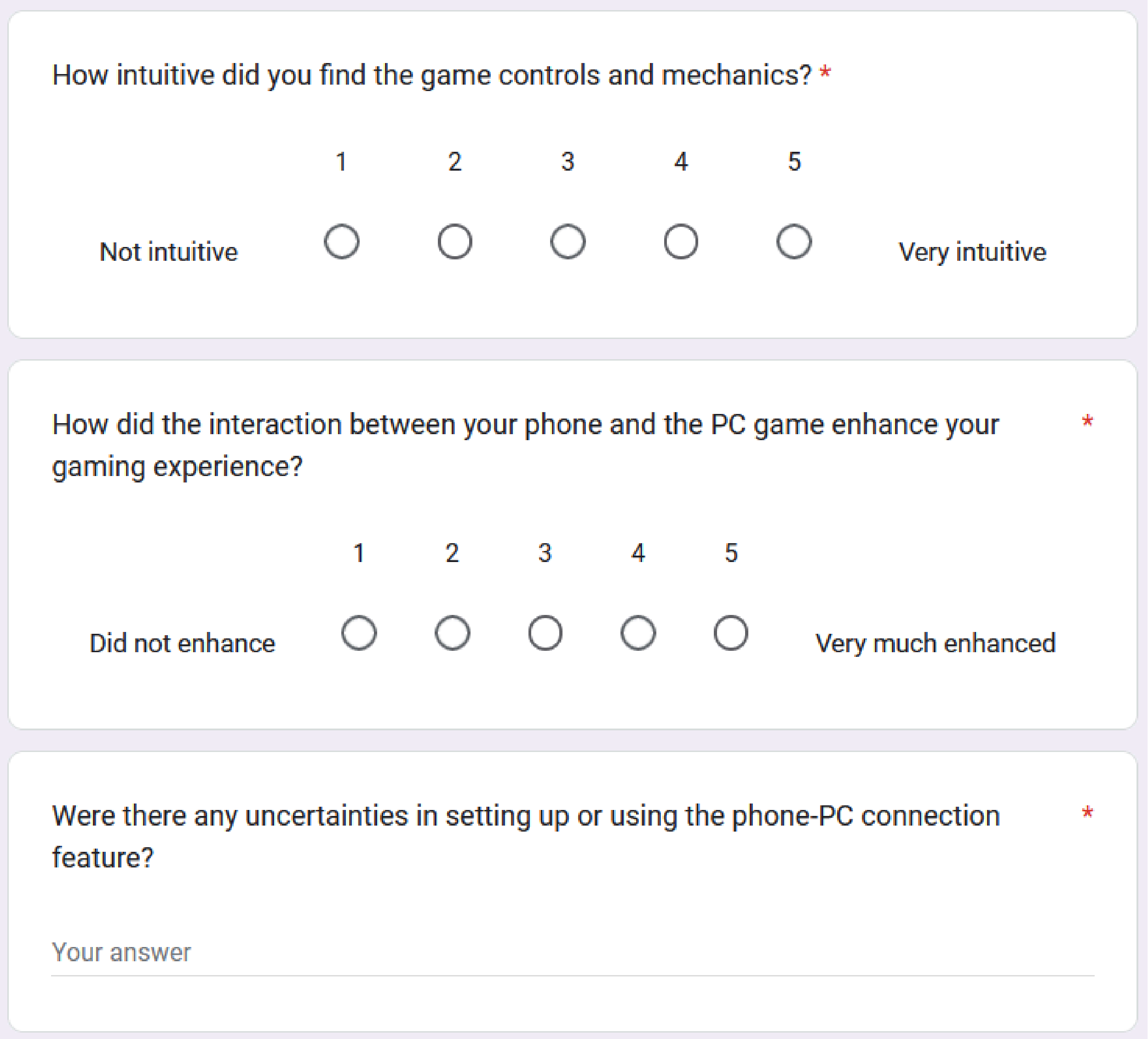
	1	2	3	4	5	
Negatively affected	○	○	○	○	○	Positively affected

How did the camera settings and perspectives enhance or detract from your gameplay experience? *

	1	2	3	4	5	
Significantly detracted	○	○	○	○	○	Significantly enhanced

Were there any moments where the camera perspective made it difficult to understand or interact with the demo? *

Your answer



How intuitive did you find the game controls and mechanics? *

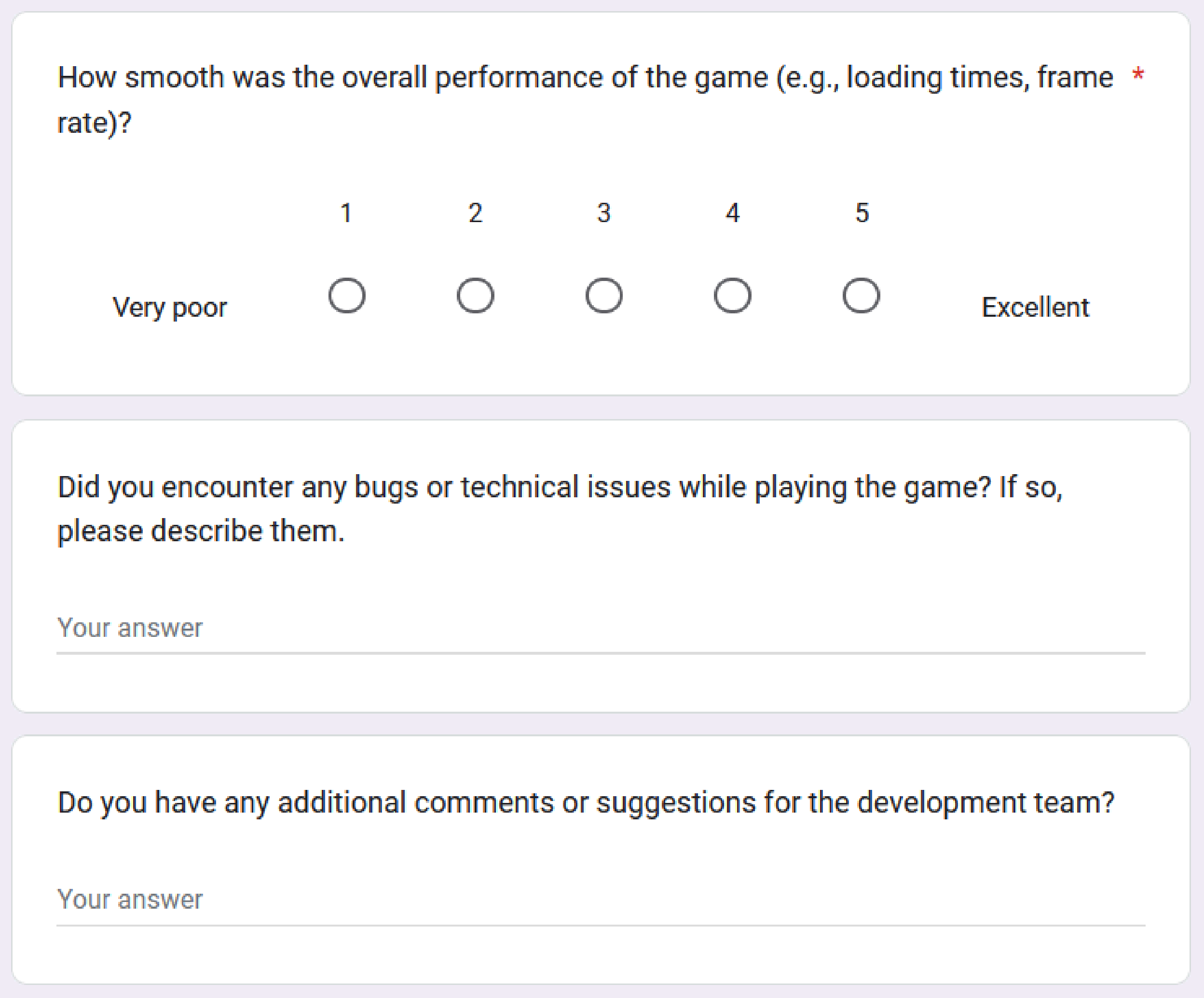
	1	2	3	4	5	
Not intuitive	○	○	○	○	○	Very intuitive

How did the interaction between your phone and the PC game enhance your gaming experience? *

	1	2	3	4	5	
Did not enhance	○	○	○	○	○	Very much enhanced

Were there any uncertainties in setting up or using the phone-PC connection feature? *

Your answer



How smooth was the overall performance of the game (e.g., loading times, frame rate)? *

	1	2	3	4	5	
Very poor	○	○	○	○	○	Excellent

Did you encounter any bugs or technical issues while playing the game? If so, please describe them.

Your answer

Do you have any additional comments or suggestions for the development team?

Your answer

Appendix D: Procedure of Study

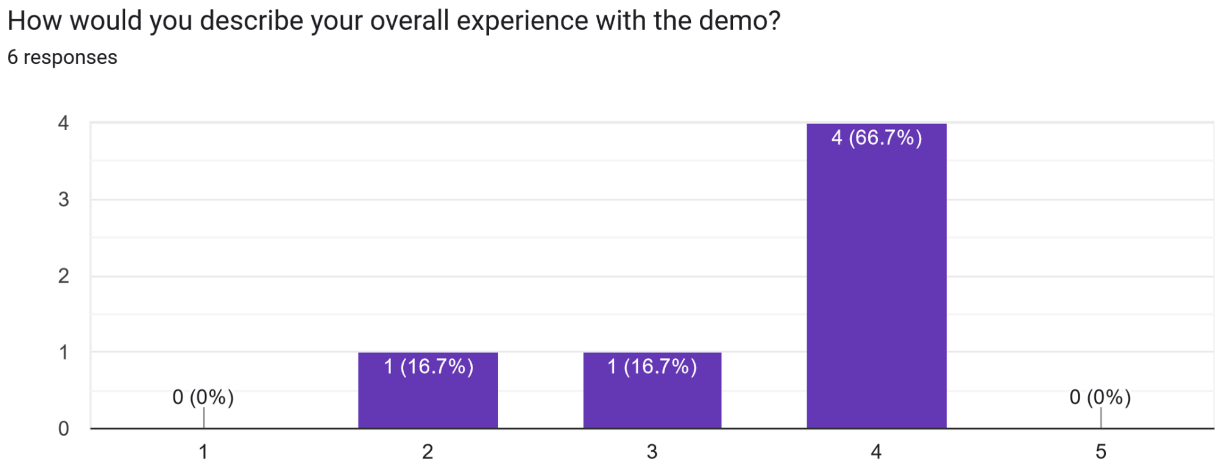
What is a DROPTABLE?

DROPTABLE is an inventive project designed to create new PC gaming mechanics through the Unity Engine. It introduces a unique gameplay concept where players use their phones as part of the game's controller, as crafted by our team. The game experience contains a visual novel within a detective narrative, accompanied with minigames that use the cell phone mechanic for a deeper engagement.

The Procedure of the Study:

- The researcher will introduce and ask the participant following the COVID-19 protective regulation.
- The researcher responses for corresponding subjects will read and introduce the consent form to test subjects and ask them to sign it if they agree with them.
- The researcher will collect the consent form and preserve them with care.
- The researcher will need to introduce the DROPTABLE game to the test subject.
- During process 4, the researcher will briefly summarize the game without spoilers.
- The researcher will ask the test subject to play through the whole game.
- The test subject can explore if they wish to.
- During the process 5 and 6, the researcher will ask the subject to use “think aloud protocol”, in which the subject needs to speak out aloud what they are doing and the researcher will not explain anything unless necessary. These steps will take 15-30 minutes, up to 45 minutes if necessary. The researcher will observe the subject's action and record them without any characteristic information.
- Then, the researcher will ask the test subject to fill out a Google form containing questions about advice on the game.
- If they need an IMGD playtest score, the researcher will sign on a paper with the statement that proves they have engaged in the event.

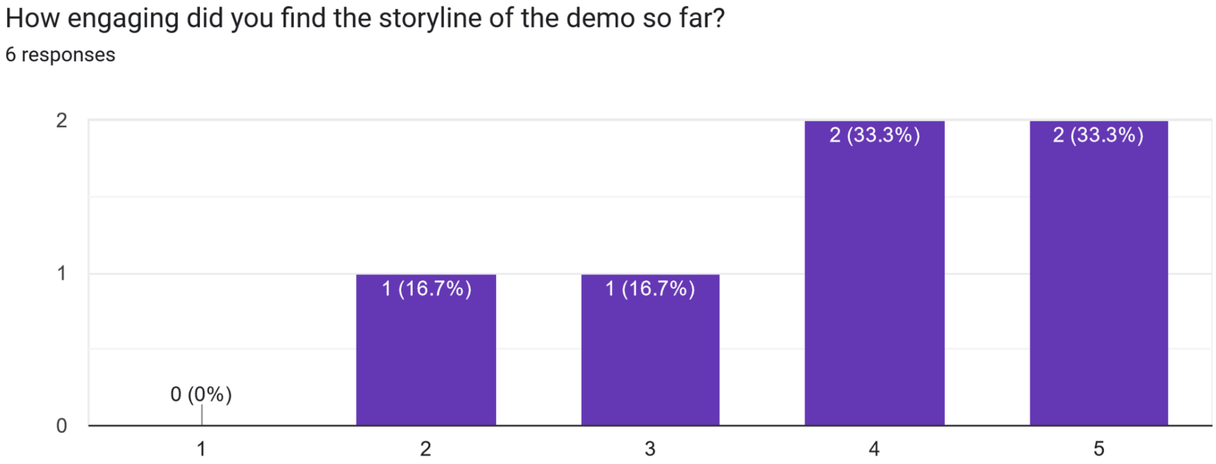
Appendix E: Survey Results


How would you describe your overall experience with the demo?

6 responses

Can you describe a moment in the demo that stood out to you significantly?

- ➢ I really liked connecting the dots between sharp objects in the trash bags and then realizing that Jacob definitely hid the murder weapon there. I also was baffled when Tom died during the night, that helped me understand that Jacob is for sure doing shady stuff.
- ➢ The second time I got Jacob's phone I was very alert and made sure I was ready to close the tab quickly
- ➢ The phone and computer compatibility is cool and would be interesting to explore further, though it kind of feels forced into the game rather than a game around that.
- ➢ No.
- ➢ I think using your actual phone to represent looking through your friend's phone is a really cool idea, and would be very cool to expand upon.
- ➢ the phone mechanic is interactive

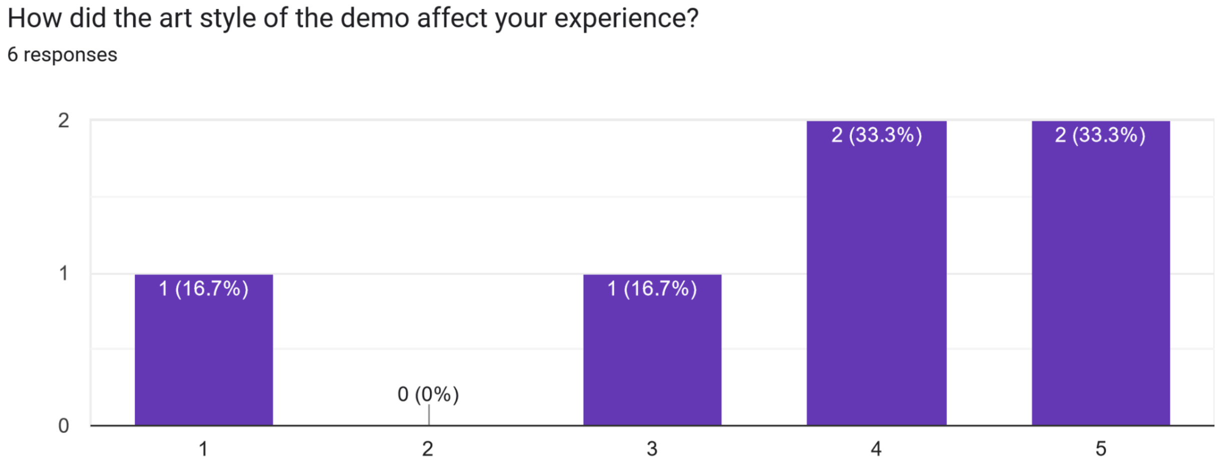


How engaging did you find the storyline of the demo so far?

6 responses

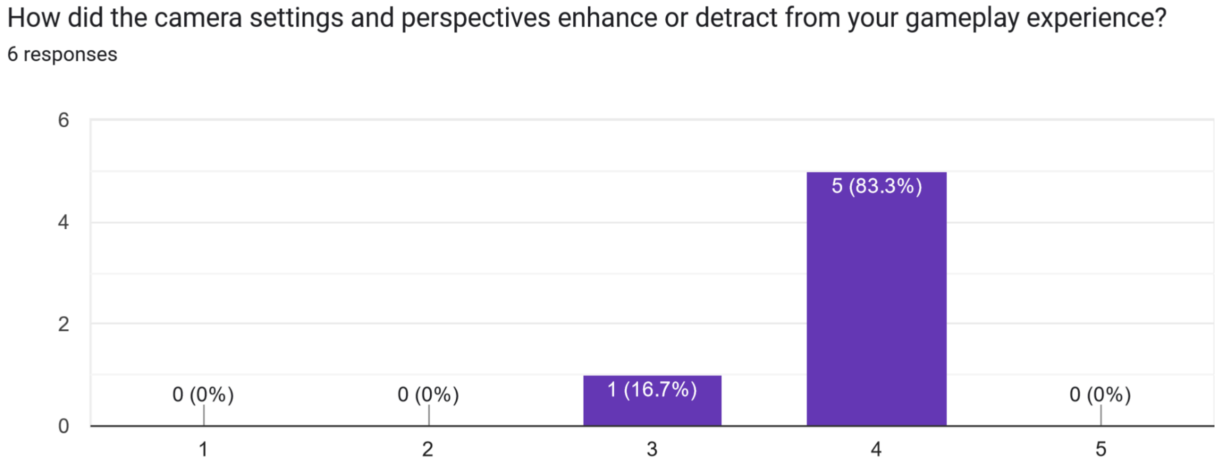
Were there any plot points or narrative elements that you found particularly compelling or confusing?

- Once again the trash bags, but also the design of Rachel made it very easy to become attached to her and recognize her in the other scene. I was a bit jarred by the dialog though, the characters sounded very robotic, and seeing as they are college kids they would absolutely be using more slang or shortening their sentences.
- I found basically every interaction with Jacob interesting since a lot of it was done with this suspicion. I liked how the player got the option of suspecting him at the end of the third day
- The story while linear was way too rigid and doesn't seem to have much depth. Everywhere you go seems like it is only there for a story reason but in some of the story bits you don't really seem to have a reason to actually do such things.
- It was rather straightforward
- Jacob said they were on a long drive to Boston, but they met someone who worked at a store there that also goes to their school. Being a part-time worker at a faraway store is kinda crazy to me.
- the skirt clue is interesting



How did the art style of the demo affect your experience?

6 responses



How did the camera settings and perspectives enhance or detract from your gameplay experience?

6 responses

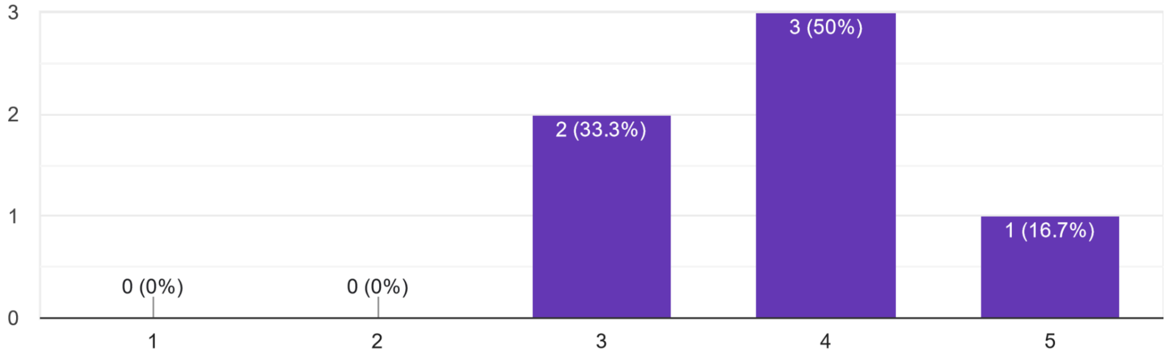
Were there any moments where the camera perspective made it difficult to understand or interact with the demo?

- ➢ It was easy to pick up, but the cafeteria was hard to navigate.
- ➢ The zoomed-out shot in the cafeteria made it difficult to get to or see the vending machine
- ➢ I like the camera perspective, but at the same time, it made moving around some of the places sort of difficult at times.
- ➢ Yes, in the cafeteria scene, I was very confused as to what to do next. Some of the angles made certain parts of the map look like they were walkable when they were not.

- Sometimes the camera would change unexpectedly and it would take me a second to adjust to the new perspective.
- The vending machine was too far away and blended in with the shaders, making it hard to see.

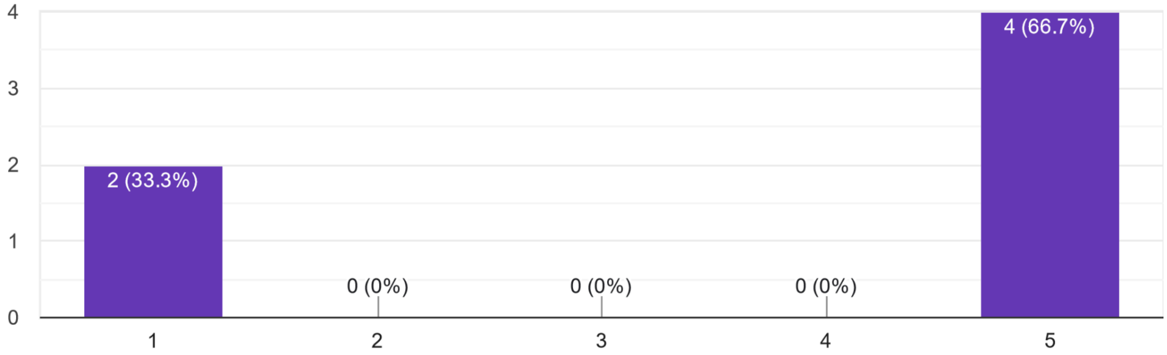
How intuitive did you find the game controls and mechanics?

6 responses



How did the interaction between your phone and the PC game enhance your gaming experience?

6 responses



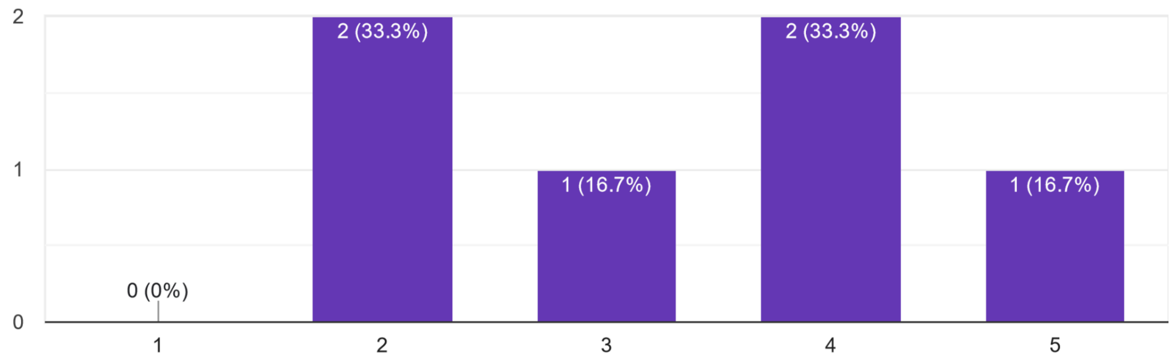
Were there any uncertainties in setting up or using the phone-PC connection feature?

- We had to reset it a few times, and I noticed that the dress picture disappeared the moment I first clicked on it. Other than that I enjoyed the connection between them, it was super fun!
- I put the phone upside down instinctually when I was supposed to close the tab

- The phone-PC connection is definitely an innovative feature but it sort of just feels like it was forced into the game rather than having a game built around it. Such a feature is a selling point to a game as a new thing, you want to capitalize on that.
- There was no indication that I needed to use my phone to progress the storyline.
- I don't think so, just it doesn't appear to work on iOS.
- Make cookie policy known to users

How smooth was the overall performance of the game (e.g., loading times, frame rate)?

6 responses



Did you encounter any bugs or technical issues while playing the game? If so, please describe them.

- The characters do a little dance whenever you walk them into something, Morgan doesn't stop walking, the connection between the phone and computer is pretty iffy at times, and Jacob's phone is on the table 24/7 when they are in the car its just an empty void out there, some dialog goes way too fast to read entirely, the second time I snooped Jacob's phone he repeated the dialog 3 times before I could move on, and the clothing store is difficult to navigate because you just walk around until you find something to click on (maybe have Jacob walk to the items you want to look at?).
- The player character wiggles a bunch when coming into contact with anything else which looks a bit funny
- Issues with the phone at the start, Movement is a bit janky and rigid, easy to glitch out the character, day 2 displayed day 1 instead

- There was one instance in the bedroom during a transition where Tom started spinning uncontrollably
- The QR code kept appearing throughout the game, but I didn't necessarily have to scan it, I just refreshed the page on my phone.
- going out of bounds with the other classroom door, some text repeating, and pathfinding sometimes looks funky

Do you have any additional comments or suggestions for the development team?

- Although I found a lot of bugs, this game is very well done. I do enjoy playing and I am very hooked on the story and play style of it. Please keep me in the loop for when it is updated because I would love to playtest more or play the full game when it is released. Thank you so much!!
- The art style is very cool and really adds to the mood
- There were quite a few times where I wasn't sure exactly what to do and I ended up wandering around until I found the answer. This especially happened the first time I used the phone & in the clothing store.
- An indicator to Press E to advance dialogue would be cool.
- A better indicator is that some dialogue is automatic and some require button pressing to advance it.
- Make sure people use Android

Appendix F: Think-aloud Observation Notes

Playtester 1:

- ➢ Day 1
    - ○ Difficulty knowing to leave class
    - ○ "I like the pixilated vibe."
    - ○ Size of characters is inconsistent & jarring
    - ○ The subtitles before the start of minigame 01 are too fast
    - ○ Connection did not work the first time
        - ■ Restarted game to try again at observer's request
    - ○ Tom's room is bland
    - ○ Skipping dialogue is nice
        - ■ Confused why some dialogue isn't skippable (referring to subtitles)
    - ○ Excited about the choices
    - ○ The "ARG" type concept is really cool
    - ○ Dialogue is a little stiff
    - ○ Tom's death after choosing the top option was shocking
- ➢ Day 2
    - ○ In some scenes, it is difficult to see the objects
    - ○ Believes that Jacob is 100% the killer
    - ○ Made a connection between the sharp objects in the garbage bag at the start of the game and the murder
    - ○ Clicking things multiple times in minigame 02 causes dialogue to loop
- ➢ Day 3
    - ○ Understood the controls for the door minigame fairly well
    - ○ Attempted to go back into Jacob's room after talking with him
    - ○ Can't find objects to interact with quickly in the clothing store
        - ■ Found all the objects eventually on their own
        - ■ Finding each object individually was difficult

Playtester 2:

- ➢ Day 1

  - Tried to speak with the Professor before leaving class
  - Explored the scenes
  - Didn't know to click the skirt picture
    - Needed to be given direction from the observer
  - Did not pick the option that kills Tom
- Day 2
  - Difficulty seeing where the vending machine was in the cafeteria
  - Bug: "Day 1" text appeared when going back into the cafeteria
  - Was guided through minigame 02 without any difficulty
- Day 3
  - Succeeded on the door minigame
    - Did not see Jacob's model spawn after finishing the minigame due to being in the kitchen camera perspective (confused what to do next until the camera perspective happened to change)
  - Attempted to go back to Jacob's room after talking with him
  - Found items in the clothing store fairly easily
    - Two of the items being near each other threw the playtester off at first
  - Controlling the camera with the mouse in the car was intuitive
  - Make small comments about the strange wording of dialogue
- Final Thoughts
  - Liked the mystery & phone gameplay
  - Phone added an interesting layer to the visual novel gameplay

Playtester 3:

- Day 1
  - The mouse showing at the start of the game confused the player
  - Tried to click to advance past Morgan dialogue
    - Inconsistency in controls to advance dialogue (click isn't always available)
  - Guided well through the progression of day 1 due to subtitles
  - Morgan being frozen while walking was weird
  - Unsure what they were meant to do on the phone

    - Didn’t know to click the photos
    - Needed to be guided by observer
  - Connection didn’t work the first time
    - ICE candidates were not exchanged
    - This bug has never been seen before
  - Chose the bottom option during the choice dialogue
- Day 2
  - Found the vending machines easily
  - Found the person to talk to easily
  - “Day 1” text appeared again when entering the classroom
    - Slightly confused the player
  - Waited for subtitles to end before clicking anything during minigame 02
- Day 3
  - Accidentally skipped the door minigame by trying to click “E” through subtitles
  - Did not know to control the camera with the mouse in the car scene
  - Walked past the store before subtitles were finished, making them need to search rather than finding it naturally
  - Found the item in the back of the store second
  - The dress stood out and they found it quickly
  - Chose the romance route option in the final car scene
- Final Thoughts
  - Remove glitching student walking (the NPC snapping to the ground in the classroom)
  - Let the player walk up the stairs in the classroom
  - Add a scene outside the school rather than just going straight into the town
  - Phone parts are railroaded
    - No narrative reason for the play to go on the phone for minigame 01
  - Did not understand the point of introducing Morgan
    - Felt there was nothing that came of it
  - Story feels very rigid
    - Every place is only visited for 1 purpose

    - Different versions of text is weird (dialogue vs subtitle)
    - Movement is slow and janky (spinning when walking into something)
        - Especially in clothing store
        - Maybe add “Shift” to run
    - Nothing to interact with downtown
        - Gives a feeling of “Why does this exist”
    - Highlight all possible intractable objects to help with discoverability
    - Connection with the phone feels forced
        - The phone UI could just be integrated into the game

Playtester 4:

- Day 1
    - Explored classroom
    - There should be some indicator for where the seat next to Jacob is
    - Tried to talk to the professor
    - “Did I just exit the classroom straight onto the street?”
    - Characters/models blend into the scene
    - Felt like the wording was weird
    - Thought the buildings on the street were strange for the location
    - There should be more of an indication as to what you can do
    - Found the change of movement from 3D to 2D was odd
    - “Why do I look so old?”
    - Didn’t like being forced onto Jacob’s phone
    - Felt there should be an indication as to what to do on the phone
    - Clicking on many of the apps did nothing, killing exploration
    - Not sure if this is Jacob’s phone or Tom’s
    - Pages take a long time to load
    - The images showed felt random & unnatural
    - No indication to look back up at the screen
    - No indication to use mouse to select choice
    - Before it was only “E” to speak

    - Chose the non-death option
- Day 2
    - “Why do I have to scan the QR code instead of just looking at it on my laptop”
    - Movement was slow
    - The visual change from the street to the cafeteria was so jarring that they missed the subtitle
    - “There should be a way to pull up the previous dialogue.”
    - Unaware of what to do after picking up lunch from the vending machine
    - “My roommate killed her!”
    - Couldn’t see the door to go to class
        - Wandered around for a long time trying to find where to go
    - Objects need more contrast
    - Kept trying to click through the classroom subtitle
    - “For a game with so many settings, it’s a shame you can’t interact with any of them.”
    - “The only clues you are given are the ones you are forced to see.”
    - Contents in browser were distracting to the subtitles on the computer
    - “Why does my roommate have a whole lab?”
- Day 3
    - Expected to interact with the door minigame with “E”rather than the mouse
    - Empty scene for the car scene was strange
    - Didn’t know to use the mouse to look around in the car
    - Had difficulty reading the sign for the clothing store
    - Jacob’s movement was glitchy
    - Found the 1st item easily
    - “Am I just meant to wander around to find the next item?”
    - “The walking is so slow.”
    - “There should be a light glow or outline to indicate what I should interact with next.”
    - No picture for the receptionist dialogue was strange
    - Talking about the dress in front of Jacob is strange

    - “Am I just supposed to stand here ‘till they’re done?”
    - Thought that Tom was really bad at lying
    - Discovered the ability to control the camera with the mouse in the car
    - Strange that we can’t see ourselves in the car
    - “No idea what DD is.”
    - The car subtitles should be dialogue
    - Romance route choice made
- Final thoughts
    - Game was boring
        - Nothing to do during dialogue
        - Visual stimulation is needed
        - Dialogue photos were not good
    - More interactable items needed
    - Walking was far too slow

Playtester 5:

- Day 1
    - Tried to talk to the professor
    - Tried clicking through Morgan’s dialogue
    - Add prompt for how to advance text
    - Explored living room & Tom’s room
    - Was guided well through day 1
    - Tried exploring the apps on the phone
    - “Is this representing Jacob’s phone”
    - Didn’t click on photos
    - Device orientation prompt was out of place (being tested on iPhone)
    - Had difficulty knowing what to do on the phone
        - After being hinted to click on photos by the observer, it still took a while to click on the necessary photo
    - Tried using the computer during minigame 01 to progress
    - “I probably should have paid attention to what he said.”

  - The mix of subtitles and dialogue is strange
  - Chose the good option
- Day 2
  - Found vending machine easily
  - “Day1” text appeared when entering the classroom
    - (I think this happens when using the GameStart script)
  - Progressed through day 2 easily
  - Didn’t notice subtitles started after opening the album in minigame 02
  - Waited for subtitles to end before continuing
- Day 3
  - Didn’t use mouse for door minigame since he didn’t see the cursor
  - Didn’t look around in car scene
  - Invested reaction for the dialogue
  - Found store items after some searching
    - Difficulty finding the last item
  - Tried talking to Emily while searching for items
  - “Did I miss something?”
  - Didn’t walk far enough back for the handbag item to be highlighted
  - “She’s so far away from her house!”
  - “Yo, you are not good at lying!”
  - Non-romance choice made

Playtester 6:

- Day 1
  - “I like the art style.”
  - Explored classroom before sitting down
  - “Can I go faster?”
  - Noticed grammar issues in dialogue
  - “Damn, Jacob looks good!”
  - “Why is some dialogue just going when others I have to click through?”
  - Subtitles go a little fast

  - Confused about what’s happening in the Morgan scene
  - Walk out of map using the far away classroom door
  - Morgan perspective change was shocking
  - Tried to click though Morgan dialogue instead of using “E”
  - Commented on grammer errors
  - “I didn’t see what they said.”
  - “Is he talking to himself audibly or in his head?”
  - “Why’s he so passive aggressive?”
    - Referring to Tom talking to Jacob in the living room
  - Subtitles are too fast
  - Tried exploring the apps on phone
  - Contacts should also have last names
  - Found photo by randomly clicking
  - Chose the Tom death ending
    - Assumed Jacob killed Tom in his sleep
  - Restart to continue the playtest
- Day 2
  - “I saw Rachel in the contacts. Maybe I should be a little more concerned.”
  - The cafeteria scene is a little hard to see
  - Found Emily to talk to easily
  - “Fuck, Jacob killed her.”
  - “Day 1” text appeared when entering classroom
  - Tried to look at photo on phone while in classroom (outside of minigame)
  - Guided through minigame 02 easily
  - Connected on disconnect between phone UI in the browser vs computer dialogue for the browser
  - Dialogue repeated on the computer for the end of minigame 02
- Day 3
  - Figured out the door minigame (success)
  - Confused at the end of the food minigame
    - Ended up talking to Jacob regardless

- Continually tried to use phone outside of minigames
- Found bag item first in clothing store
- Found pants easily next
- Found last item after only a little more searching
- “I feel like the man should be the trigger for “What about this one?” rather than me needing to be there.”
- “What’s DD?”
- Chose non-romantic route

➢ Final Thoughts

- “I don’t want to use my mouse, especially if my hand is on the phone.”