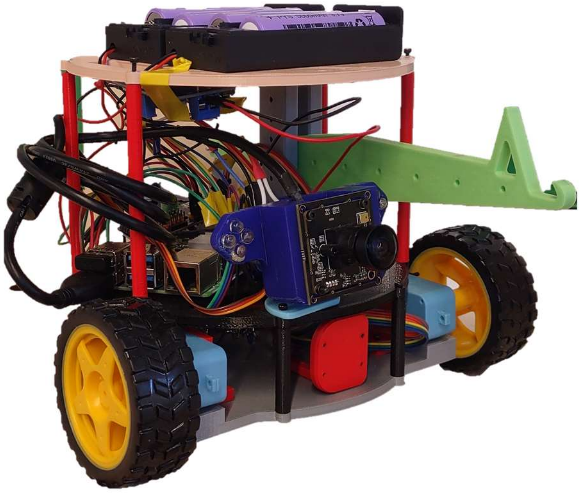
Robot Escape Room v3:

Robot Escape Room Design Engine



A Major Qualifying Project Report submitted to the Faculty of the WORCESTER POLYTECHNIC INSTITUTE in partial fulfillment of the requirements for the Degree of Bachelor of Science

BY:
Iain McEwen (CS)
Conner McKevitt (CS/RBE)
Kaelin Panneton (CS)
ADVISORS:
Professor Berk Calli
Professor Melissa Kagen
Professor Gillian Smith
DATE: April 25th, 2024

Acknowledgements

We would like to thank our advisors Berk Calli, Melissa Kagen, and Gillian Smith for their continued guidance and support throughout the project. We would also like to thank anyone who has playtested with us and provided feedback on our designs. Finally, we would like to thank the previous years' teams for creating a project that this year's team could build upon.

Abstract

This project spans three major areas: Computer Science, Robotics Engineering, and Interactive Media and Game Development. These three disciplines were interwoven to create a modular robot escape room design tool and play experience. The goal of this project was to create a robot escape room system in which a user could design a custom escape room then make and play their escape room with little technical knowledge. This was created in four main parts, the backend, the design wizard, the room, and the robot. The result of this project was a robot escape room system that accomplishes our goal while providing a base starting point for a large amount of feature development.

Authorship

Section	Author
1.0 Introduction	Kaelin Panneton & Conner McKevitt
2.0 Background	Kaelin Panneton
2.1 Escape Rooms	Iain McEwen
2.2 Game Engines	Kaelin Panneton
2.3 Robot & Room	Conner McKevitt
3.0 System Methodology & Design	Conner McKevitt
4.0 Engine & Website Methodology & Design	Kaelin Panneton & Iain McEwen
5.0 Room Methodology & Design	Conner McKevitt
6.0 Robot Methodology and Design	Conner McKevitt
7.0 Evaluation through Playtesting	Kaelin Panneton
8.0 Challenges	Kaelin Panneton & Conner McKevitt
9.0 Next Steps	Conner McKevitt
9.1 Website	Kaelin Panneton & Iain McEwen
9.2 Engine	Kaelin Panneton & Iain McEwen
9.3 Room	Conner McKevitt
9.3.2 Core Redesign	Iain McEwen
9.4 Robot	Conner McKevitt
9.5 Backend System	Conner McKevitt
9.6 Classroom Modifications	Kaelin Panneton
10.0 Conclusion	Conner McKevitt

Table of Contents

Table of Figures

1.0 Introduction

Escape Rooms are an exciting and fun experience in which an individual must escape a room using clues found throughout the room. The series of MQPs that this project follows offered an experience where the players use remote controlled robot to escape a miniature room (Clune et al; Moore et al). These experiences offered a unique story driven escape room with a novel concept.

This year we wanted to provide a unique spin on the robot escape room concept that leveraged the skills of the team. Unlike the previous iterations which were a collaboration between CS, RBE, and IMGD student, our team comprises of three computer science students, one of which is a robotics engineering double major. As such, we decided to focus on the technical aspects of the project and make the project more accessible in the future to those without a technical background.

The goal of this year's MQP was to create a robot escape room system in which a user could design a custom escape room then make and play their escape room with little technical knowledge. To accomplish this goal, we split the project into three parts each with their objectives. These are:

1. **Design Wizard: To create a tool in which a user can design and implement their own robot escape room.** This tool should have functionalities for being able to assemble a unique room layout with customizations for configuration of puzzles and order of completion for puzzles.

2. **Room: To design and implement a modular system that can be adapted to user made designs with little technical know-how.** A new system was needed to accommodate the ability for a user to design their own escape rooms. It should be easily expanded upon while not requiring players to touch code or wiring during assembly.

3. **Robot: To iterate on the robot so that it has the base features required to allow a player to escape the room while offering a platform that can be extended and modified.** The robot inherited from the previous project iteration was a good starting point. We wanted to identify the issues with the robot and resolve them while adding features that would allow more possibilities in an escape room setting. We also wanted to provide a good starting point for future iterations and feature development.

2.0 Background

2.1 Previous Iterations

Before starting research into potential areas of development for the project, the team first needed to reference what had already been completed by the previous two iterations of the Robot Escape Room project.

In the 2021-2022 academic year at WPI, a group of four students created an MQP for different disciplines of study to come together and build a robot escape room. The first iteration of this project explored each aspect from the early stages and made initial decisions, with the goal of creating an early model and framework for the creation of a robot escape room. (Buckingham et all., 2022). This initial version had some complications with playtesting due to COVID pandemic, and that's one of the reasons they decided to have the web interface to control the robot, but this ended up working well for the project. The idea was a modular escape room for the robot that fits together with puzzle-like pieces to build the walls, floors, and puzzles. They additionally built the robot and website that were connected via a locally hosted website. This started the process of having three main components: the robot, the website, and the escape room for the robot. These three core elements continue to be the foundational pieces to the present day.

In the following year, a new group of six students went to pick up where the first team left off and then built upon the project to make it their own. The second iteration of the robot escape room was built off the initial concept, but went into their own designs for the robot, puzzle design, and overall narrative elements (Bell et al., 2023). They developed their own theme with narrative and puzzles to go with it. They expanded upon the initial website design to incorporate the story of being in space and doing puzzles that involve the robot's grabber arm.

2.2 Escape Room Background

Escape rooms are very aptly named games where the player(s) must find their way out of a room or series of rooms by solving puzzles. Unlike most games, which have digital recreations of a previously physical game, escape rooms find their origin in early text-based computer games, with Behind Closed Doors (1988), credited as being the first published game to use the premise of the player being trapped within a room with the goal to escape (Ascalon, 2021). As computer graphics and user interaction rapidly improved, text-based inputs fell out of popularity leading to Crimson Room (2004), a point-and-click game where the player is introduced by waking up within a locked room. Created in flash and shared around the web, it became widely popular and is credited as the inspiration for the genre going forward (Exit the Room, 2022; Ascalon, 2021). They crossed into the physical world in 2007 when Takao Kato started the first escape room in Japan, the REAL ESCAPE GAME (SCRAP Entertainment, 2020). With the desire to create a collaborative event, and inspired by a girl who was hooked on escape games, Tako created a small escape room that was instantly popular, founding the company SCRAP and spreading across Japan with more physical rooms (Kato, 2016). This marked an important change in escape rooms, shifting perceptions from a single-player experience, as most online games were, to a collaborative experience where people have to rely on one another. From there, escape rooms spread across the world, now able to be found on every inhabited continent.

2.2.1 What Makes an Escape Room

The core of a physical escape room is a collaborative puzzle-solving experience, where you are trapped in either a single room, or a series of rooms. There is great variation found between escape room puzzles, both in the style of puzzles and the way the puzzles progress. Most styles of puzzle are fair game to be found in escape rooms, untying rope, word searches, math problems, or even riddles. Occasionally physical puzzles can be found, such as fishing out a piece of metal otherwise unreachable with a magnet, placing an object in the correct spot, or finding secrets behind a painting, so long as they are not physically intensive. This restriction exists for two reasons, to keep the game focused on mental acuity and collaboration, and to preserve the escape room itself. Many escape rooms could be solved by tearing doors from their hinges or breaking open a lock, however many escape rooms are set up very carefully to guide the players. Having a puzzle that requires an excess of force encourages players to find other ways to apply that solution, as you have effectively told them that force is an acceptable solution.

Escape rooms are not a single puzzle, instead a series of puzzles that allow players to spread their focus, not crowding around and inhibiting each other from solving the puzzle. Each puzzle progresses towards the goal of escaping the room but can be linear or nonlinear. A linear puzzle progression is where each puzzle gives a solution to the next, and that one to the one after, until you have escaped. This creates a wonderfully designed experience, yet creates a bottleneck at each puzzle, requiring each player to ignore the rest of the room, if the other puzzles are even presented to them. Alternatively, a nonlinear escape room has one large puzzle in order to escape, but requires other puzzles to be solved first, by hiding physical pieces or information behind other puzzles. This creates an experience that will allow each player to explore the room and play to their differences but does not ensure a collaborative experience until the last puzzle. As with life, the best progression is a balance between the two, and also depends on the goal of the room and intent of the designer.

Finally, escape rooms are an experience. The core of that is being in a room with other people, but people love stories and situations. Ever since Crimson Room (2004) introduced players with a bit of reading explaining that you have woken up in this room and don’t know how you got there, your story in the escape room has been a key piece of the experience. Not only can it allow the players to immerse themselves in the experience, but it can also inform the players as to what puzzles to expect. If it’s themed around a painter you would think to find color-based puzzles, perhaps the background music includes hints in a musically themed room, or if it is set in the 1920s perhaps alcohol or prohibition could play into the clues. Additionally, the theme could allow for secondary goals, if you are a spy you need to steal a diamond before you escape, or as a detective you need to get a piece of evidence, allowing greater freedom in the progression of the puzzles, as they can lead to this secondary goal or to the escape. Perhaps they are one and the same, whereas as a robber, the code to escape the vault is written on the bags of money you collect.

2.3 Game Engine

A game engine is a software framework designed for the development of video games which includes various libraries and programs to support systems needed throughout the game development process. Game engines were designed with reuse and modding in mind and were designed to be highly customizable (Gregory). Typically, a game engine is a piece of software or code that can be used to make a digital game. These concepts can also be thought of as building blocks for physical puzzles and game design, where we can present simplified options to a user and handle the complexities, so they don't have to.

A research study specifically focused on research for game engine design notes that while there may be common language used for programming concepts, there is not such common language for game design (Anderson et al, 2015). If there was to be a more commonplace language for video game design, it may be able to encompass different game engines and make cross–communication more straightforward. Additionally, since game engines are such a customized thing to make, there can theoretically be infinite of them for different games and purposes. It is important to decide if a new engine is really necessary when starting a new project or if it can use existing technology.

There is also the question of why build a game engine instead of just building the game from scratch. The main points are about time constraints, amount of work, and creative potential. The cons of it taking time to create a game engine are outweighed by it being extremely useful and targeted toward your game when finished and it can be used for future projects without having to create a new one. Once the game engine is created, it can be used for similar components differently (Wirtz, 2023).

2.3.1 Modular Game Design Process

There are a multitude of game design processes currently in existence, but our team will be focusing on the modular game design process and applying this to the creation of a game engine. Modular design can be applied to the process of creating a robot escape room, where instead of the block-building method just being applied to each level, it can be supplied separately to the robot, website, room, puzzles, and story elements.

Modular level design works by reusing and varying formation of a defined number of level elements to create entire games, most notable examples being Super Mario and Legend of Zelda Games (Modular Level Design). There are a few main components in modular level design such as choice of module sizes, working with grids, the concept phase, absolute flexibility, the "pattern problem", and asymmetrical elements. Some of these may not appear to be useful for the modular design of a robot escape room game engine, but some of these can directly apply. Module size can be directly related to the size of puzzle or code pieces in order to be combined into a larger final project. The key part of modular design that can be described for the robot escape room game engine is the concept phase. This is the step where the designer looks at how the different modules or pieces will work together to create the game as a whole. The different ways of building will come together and overlap to be able to visualize some final product. Another benefit of using the modular design process is the absolute flexibility it allows. The building blocks established at the beginning of development can be repeatedly put together and taken apart until they accomplish whatever goal was started with.

As seen, modular design can be applied to the more game like aspects of game design, but it can also directly be applied to the software and code aspects of design. Component-based software development is a type of development where each software component is a single entity and can communicate with other components through clearly defined interfaces (Boekhoud).

2.3.2 Accessibility for Web Design

Historically video games and games in general have not had the most accessible components for making sure people of all physical and mental abilities can enjoy them. There is a difference between accessibility for games and accessibility for web design, this section will focus specifically on the accessibility of web design.

There is an existing set of guidelines for accessible web design that comes from the government itself. Section 508 of the Rehabilitation Act "requires federal agencies to ensure that their information and communication technology (ICT) is accessible to people with disabilities, unless certain exceptions apply" (Section 508). This is a more general law for various aspects of IT accessibility, not just the web, but it's helpful for understanding the polices out there and how they benefit those using the internet instead of those developing them. The next set of guidelines are clearer on what is expected for web developers and the content they create.

Additionally, the Web Content Accessibility Guidelines (WCAG), an organization with the goal of providing a standard for making web content more accessible for those with disabilities (WCAG website). These are general guidelines that can be applied to any website to make it more accessible in areas such as colors, text, audio, videos, user input, and more. These specific recommendations and guidelines make it so that websites can also better support user's existing assistive technology like screen readers, screen magnifiers, input/output devices, etc.

The WGAG has ten key guidelines for accessible web development. The first one is to provide sufficient contrast between the foreground and background. This means no yellow text on a white background, or anything similarly hard to read. Then it states don't use color alone to convey information, to ensure those who are color blind can still understand the content.

Ensuring that interactive elements are easy to identify is another key one for making sure websites are easy to navigate. The next one goes along with this and is to ensure that form elements clearly associated labels and to provide easily identifiable feedback. The next is to use headings and spacing to group related content so that the start and stop of sections is clear to the user. Creating designs for different viewport sizes is a good thing to note for people who may be accessing content on different devices such as phones, laptops, tablets. Making sure your content can be accessed by different people and devices is important no matter what kind of site you are creating. Finally, they outline that developers should include image and media alternatives such as captions and descriptions, along with providing controls for content that starts automatically, typically audio or videos. WCAG provides clear examples for each of these points so those designing websites can follow the examples.

Accessibility is one of the most important factors when creating a website for anything, game or otherwise. These key points detailed by Section 508 and WCAG are out there for developers to take advantage of and hopefully make technology more inclusive in this way.

2.4 Robot

2.4.1 Robot Control

A key aspect of making a robot escape room is detaching the users controlling the robot from the physical robot and room. This requires remote, wireless control. This is a well-explored and implemented method of robot control that can be seen all the way from toys to industrial and military applications. The remote control of robots comes with two tradeoffs, information sharing and amount of automation.

When communicating wirelessly and controlling a robot remotely, the more information that is sent between the multiple devices takes up bandwidth. If there is too much information being shared, then the time delay could become restrictive to controls. A way to offset this is by keeping some information locally on the robot and automating tasks and operations. The downside to this is that there would be less user control of the robot (Sayers 4-5).

3.0 System Methodology and Design

In switching away from a premade game that a player completes and towards a system where users can design, construct, and then play a custom game the backend infrastructure needed upgrading. This section discusses the methodology, design choices, and final design of the backend communication infrastructure for the robot escape room system.

3.1 Backend Communication

To facilitate communication between the robot, room, and the website we decided to use a backend system. Rather than making a system from scratch we search for possible industry or open-source system. We wanted a system that was easy to use, usable in python, and relatively low-latent.

MQ Telemetry Transport (MQTT) stood out to us as an option that met all our requirements. This is a lightweight publish/subscribe messaging system that can be used on small microcontrollers. This system splits its devices into two categories, the MQTT broker and the MQTT clients. The MQTT broker oversees handling the messages that the MQTT clients send. Each message is sent to a specified topic. These are used to organize the messages and clients can subscribe to a topic to be updated when it receives a new message. An example system is shown below (MQTT).

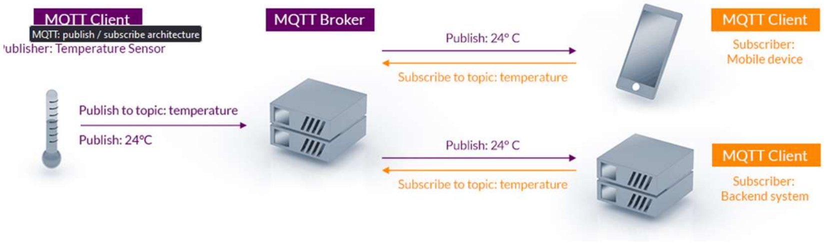

MQTT Client
MQTT: publish / subscribe architecture
'ublisher: Temperature Sensor
Publish to topic: temperature
Publish: 24°C
MQTT Broker
Publish: 24° C
Subscribe to topic: temperature
MQTT Client
Subscriber:
Mobile device
Publish: 24° C
Subscribe to topic: temperature
MQTT Client
Subscriber:
Backend system

Figure 3.1: MQTT Broker/Client System (MQTT)

We implement the Eclipse Mosquitto MQTT broker, and open-source and free broker, to run our backend. This was set to run on start up on the robot's Raspberry PI 4. The robot, puzzle controller, and room devices are all set as clients. The implementation of these clients is discussed in detail in sections 5.0 and 6.0. Each client is responsible for publishing and subscribing to key information that keeps the system running. A final system overview is shown below. More details on the specific topics and commands can be found in the documentation linked in Appendix A.

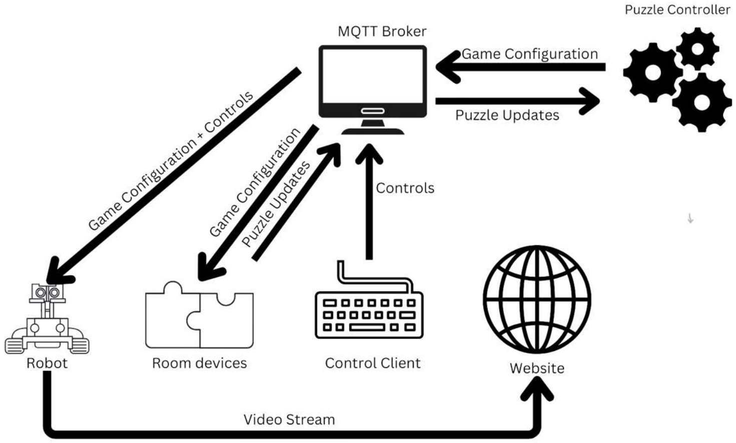


Figure 3.2: MQTT System Overview

4.0 Engine and Website Methodology and Design

The final goal experience can be described by two parts: the Robot Escape Room Designer Engine and the Modular Robot Escape Room Website. The designer has three pages: a room layout page, a puzzle progression page, and a final output and instructional page. The modular website has a few different parts: the sample HTML and CSS pages that are setup using Flask, the web_stream.py file that contains the instructions for creating new pages for the website, and the control files which are what makes the robot and room work. We wanted to provide an experience with set steps to follow for the user to build their own Robot Escape Room.

4.1 Engine and Website Methodology and Design

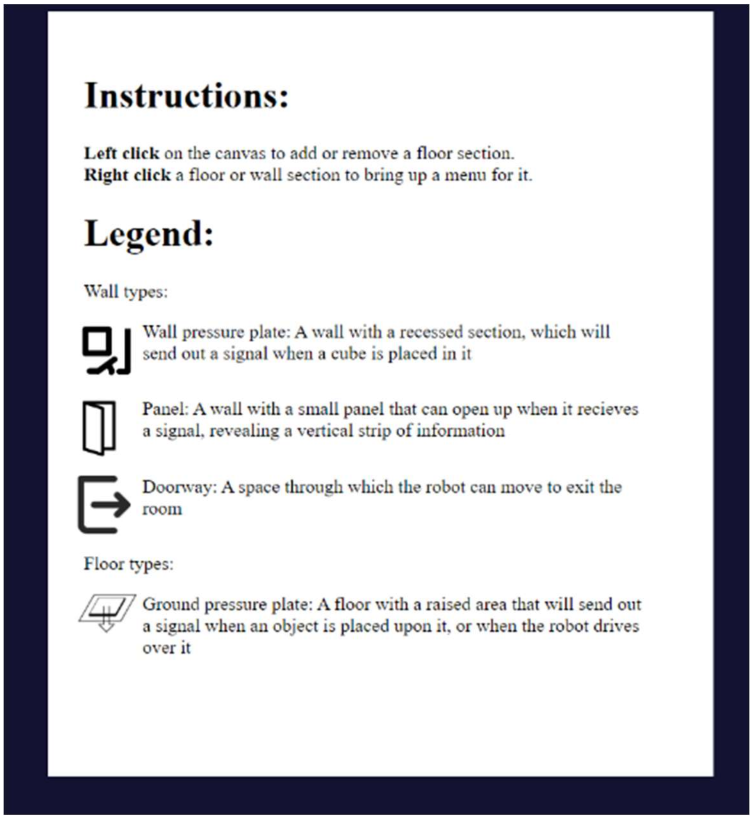
This chapter will discuss the process that was used to get to the final product for the designer engine and the website.

4.1.1 Wizard Design

An escape room has two main components: the puzzles and the room they are in. Escape rooms can custom build the layout of the rooms they use, which allows the layout and puzzles to play off each other into the desired experience of the escape room, for instance a series of branching corridors can create a sense of tightness or claustrophobia, and the designer can put clues or interactions at the ends of them to give them a need to be explored. Furthermore, they can get another sense of tightness by adding a timer, as the player would need to spend time traveling down these corridors, and potentially waste time should they need to do so again.

The design wizard facilitates the design by giving a two-dimensional floor plan, to which they can add or remove floor sections to change the layout of the room, with walls dynamically drawn. Additionally, through context menus, or right clicks, the designer can change the variation of a floor (such as a ground pressure plate) or a wall (such as a doorway) from a dropdown selection. This is stored as a two-dimensional array of space objects, which themselves are composed of the floor and connected wall piece types. This array gets saved to the session storage when the designer navigates to a different page, so that it could be redrawn when they revisit or be used for other steps, like the puzzle progression.

When presented with this page, the user can interact with the canvas by either left clicking or right clicking anywhere on the canvas. Left clicking adds or removes a floor section, while right clicking brings up a menu that allows the user to choose between different floor and wall types. The options for wall types are wall pressure plate, panel, or doorway, while currently there is only one option for floor types which is a ground pressure plate. There is an instructions panel to the left of the canvas that describes the controls and what each of these pieces will do when implemented in the room (Figure 4.1).



Instructions:

Left click on the canvas to add or remove a floor section.
Right click a floor or wall section to bring up a menu for it.

Legend:

Wall types:

Wall pressure plate: A wall with a recessed section, which will send out a signal when a cube is placed in it

Panel: A wall with a small panel that can open up when it recieves a signal, revealing a vertical strip of information

Doorway: A space through which the robot can move to exit the room

Floor types:

Ground pressure plate: A floor with a raised area that will send out a signal when an object is placed upon it, or when the robot drives over it

Figure 4.1: Room Layout Instructions and Legend

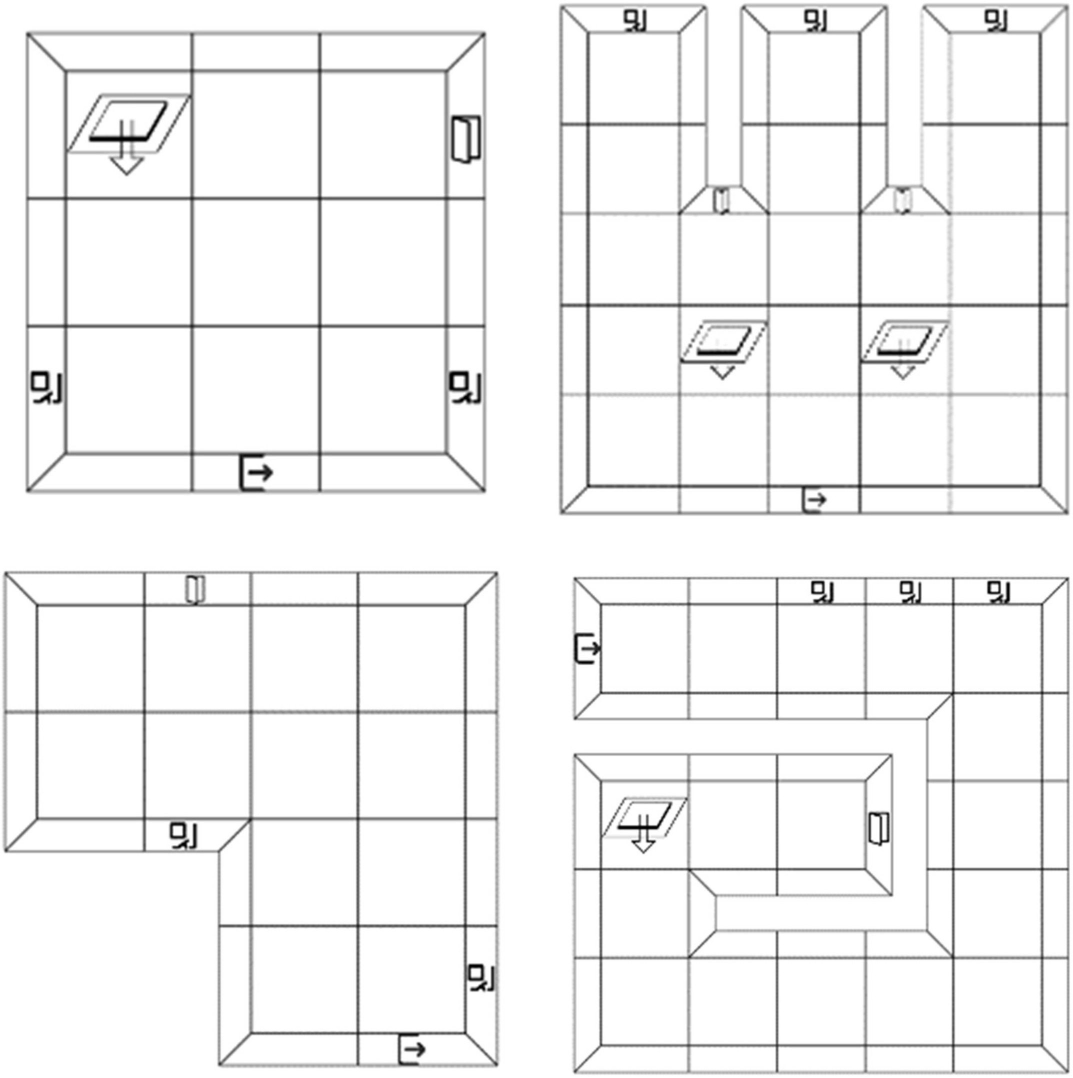
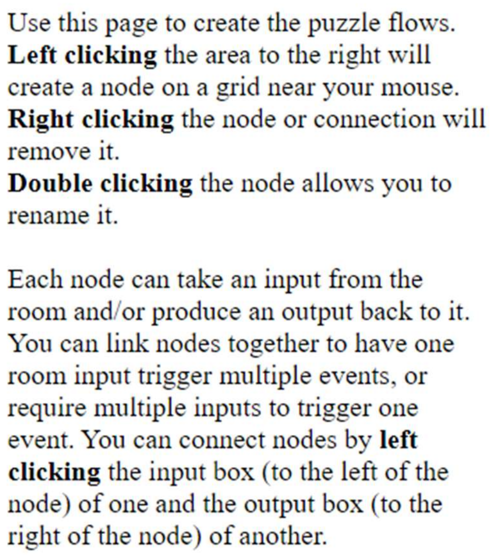


Figure 4.2: Possible Layout Variations a Designer Could Make

The puzzle progression is the cause-and-effect flow of the puzzle pieces, steps represented as nodes. These nodes are comprised of four major parts: the node input, the room input, the node output, and the room output. The distinction between node and room input and outputs is very minor conceptually, both outputs will wait for the inputs to be completed. The puzzle progression could have been a collection of one input-one output nodes; however, by adding connections between them, the designer can create more dynamic puzzles. They could have a button press open a panel and a door, have the door wait for a block to be placed and a code to be input before opening, or create a chain where a code input will open a door but only if a button were pressed to open the panel that contained the code.



Use this page to create the puzzle flows.
Left clicking the area to the right will create a node on a grid near your mouse.
Right clicking the node or connection will remove it.
Double clicking the node allows you to rename it.

Each node can take an input from the room and/or produce an output back to it. You can link nodes together to have one room input trigger multiple events, or require multiple inputs to trigger one event. You can connect nodes by **left clicking** the input box (to the left of the node) of one and the output box (to the right of the node) of another.

Figure 4.3: Puzzle Progression Instructions

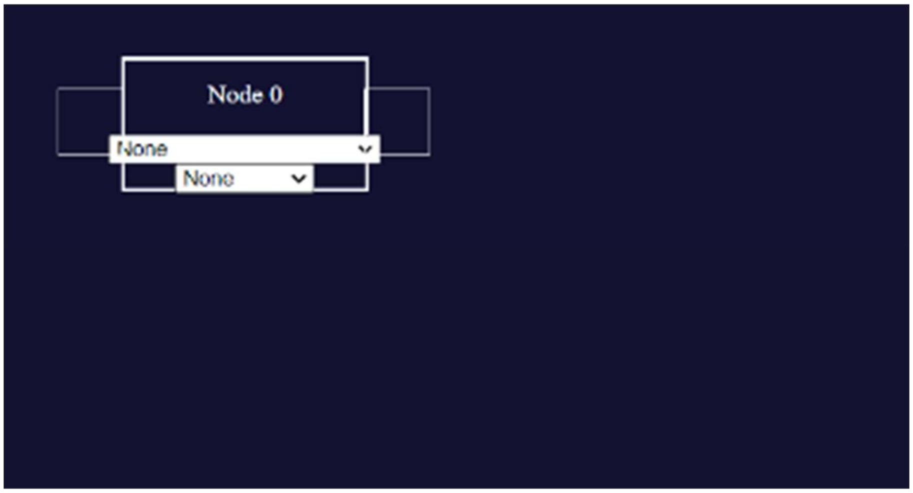


Figure 4.4: Adding a Node

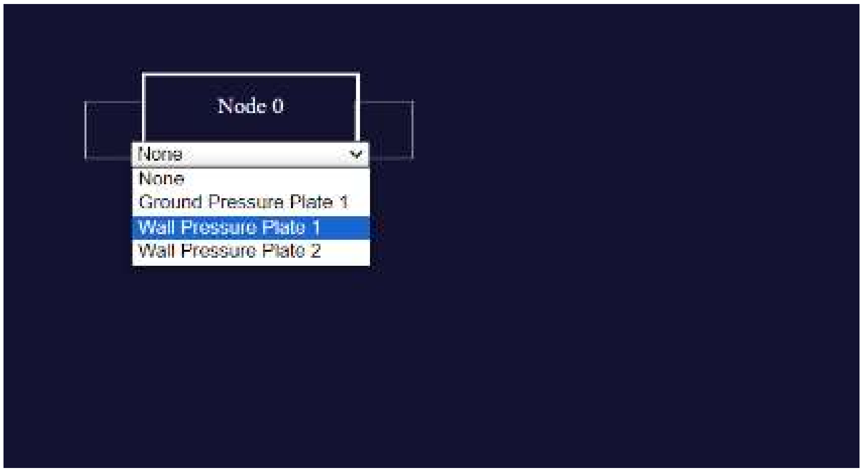


Figure 4.5: Changing the Room Input of a Node

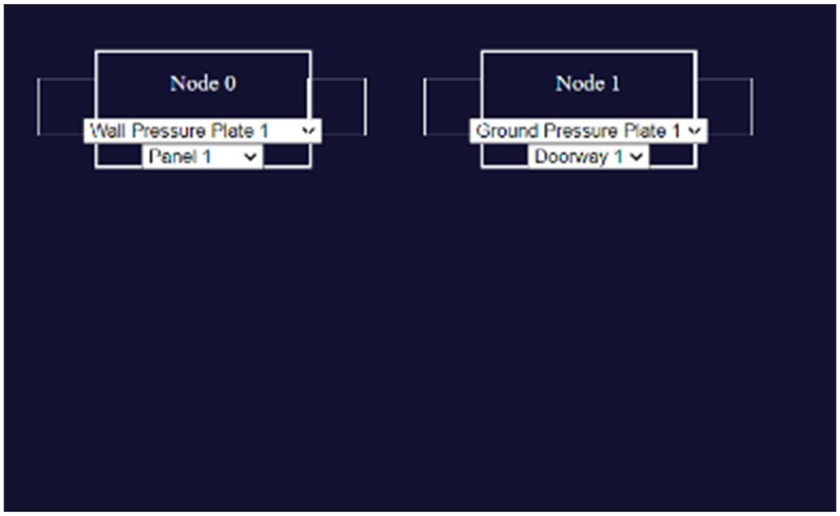


Figure 4.6: Adding a Second Node

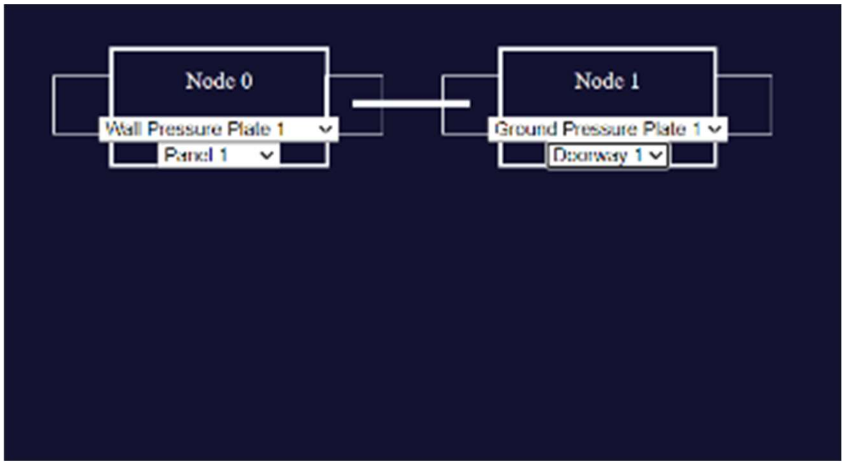


Figure 4.7: Connecting the Two Nodes

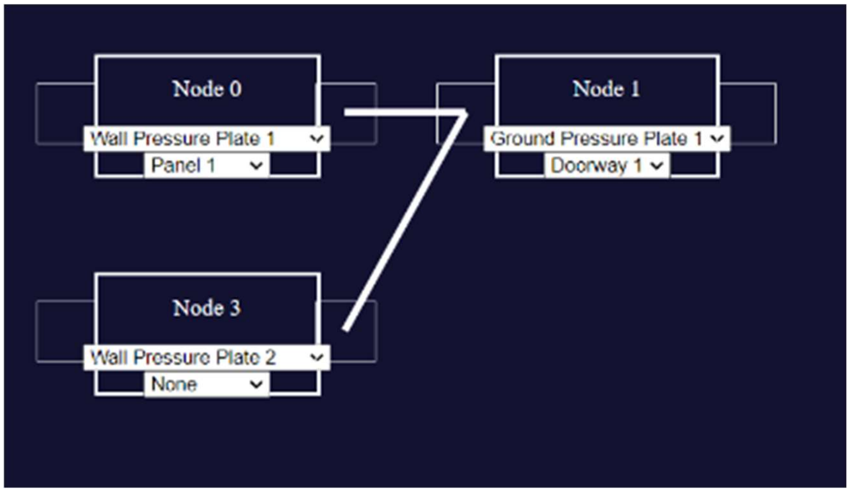
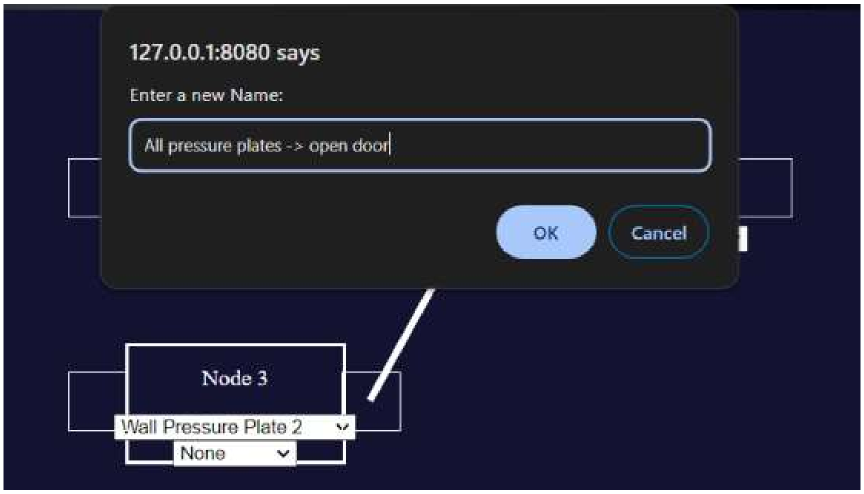


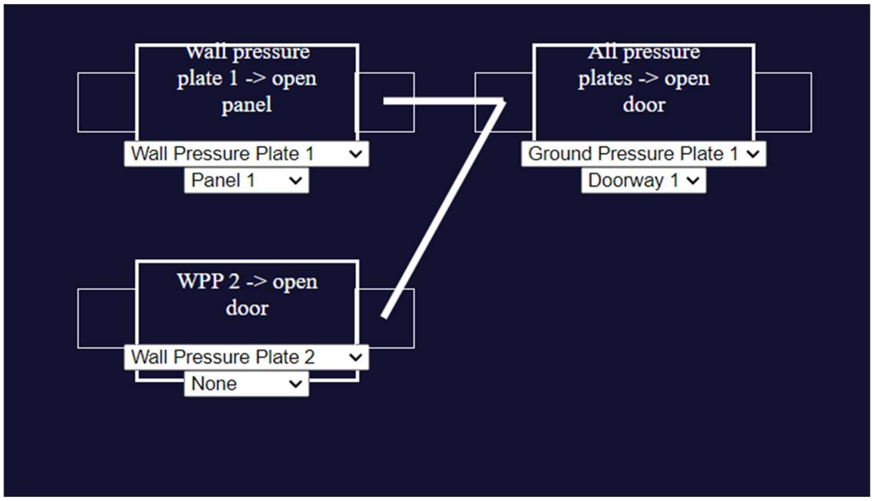
Figure 4.8: Adding a Third Node and Connecting It


127.0.0.1:8080 says

Enter a new Name:

All pressure plates -> open door

Figure 4.9: Changing the Name of a Node



Wall pressure plate 1 -> open panel

Wall Pressure Plate 1

Panel 1

All pressure plates -> open door

Ground Pressure Plate 1

Doorway 1

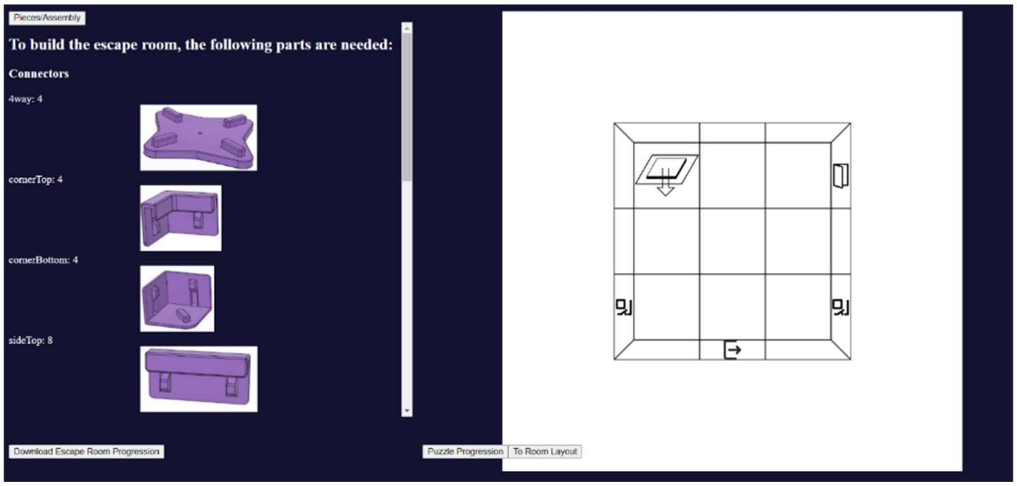
WPP 2 -> open door

Wall Pressure Plate 2

None

Figure 4.10: Final Node Progression

This page is where designers get to figure out the room mechanics and puzzle transitions without needing to understand the hardware behind them. The design of the wizard gives them the opportunity to be creative and figure out the vision now, before the room is created. Similar to the first page, there is an instructions panel on the left of the page with the controls (Figure 4.3). The user can add a node on the grid by left clicking anywhere on the page or remove a node by right clicking an already existing node. Additionally, by double clicking the title of a node the user will be able to rename it, which can be helpful because then the user can describe things in a way that is understandable to them. Figure 4.4 demonstrates this process of adding nodes and configuring them to certain components from the room layout.



To build the escape room, the following parts are needed:

Connectors

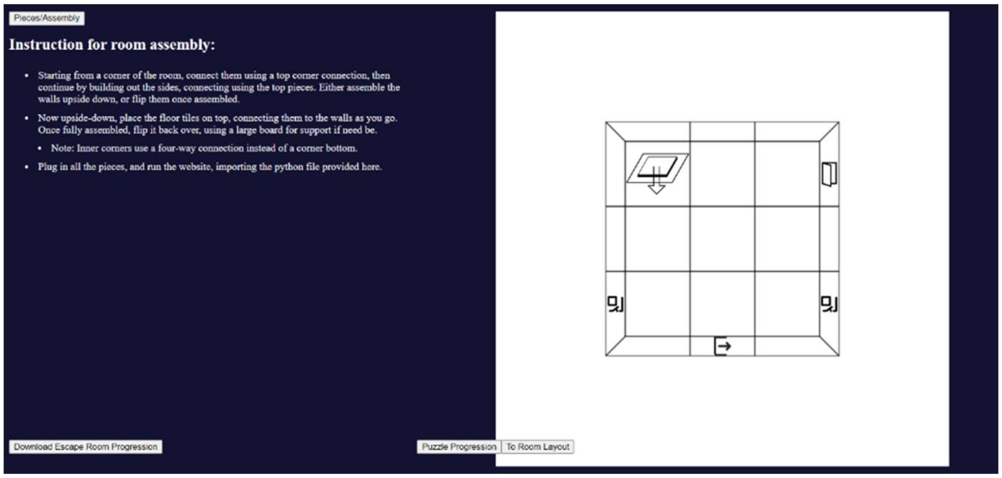
4way: 4

cornerTop: 4

cornerBottom: 4

sideTop: 8

Figure 4.11: Final Output Page with Piece Counts



Instruction for room assembly:

- Starting from a corner of the room, connect them using a top corner connection, then continue by building out the sides, connecting using the top pieces. Either assemble the walls upside down, or flip them once assembled.
- Now upside-down, place the floor tiles on top, connecting them to the walls as you go. Once fully assembled, flip it back over, using a large board for support if need be.
  - Note: Inner corners use a four-way connection instead of a corner bottom.
- Plug in all the pieces, and run the website, importing the python file provided here.

Figure 4.12: Final Output Page with Assembly Instructions

The final room layout page contains a similar layout to the first page, with the room layout displayed on the right side of the page. There is also a display on the side of the page showing the piece counts needed to build the room designed by the user (Figure 4.10). There is another option to view the instructions for how to assemble the room, as shown in Figure 4.11. This final output page ties together all the pieces from the previous steps of the design engine, so the user can then begin building the physical room and working with the hardware components.

We wanted to design the wizard as a design tool, not something to be used in place of the already existing website and system for the room. However, we did want to produce some output that can be used by some website to run the puzzles and the room itself using MQTT. Our initial design for this was a PDF with the assembly instructions, counts of all the pieces needed, and a picture of the room that was designed by the engine with a file that could be uploaded to the escape room to set it up. This design has since changed to a simpler output: a python file that contains the code required to run the room. Doing the output this way ensures the designer has what they need to run the puzzles according to their specifications in the nodes page.

4.1.2 Last year's Website Design

The website from last year had a simple design with pages for each element of the story and a UI the included the video stream, a controls setup, and some buttons on the side for things like a hint page (Figure 4.12). The problem with this system was that it was directly specific to the experience that year's team had created and would be difficult to modify for a different play experience and story without having knowledge of how web development, networking, and the robot itself.

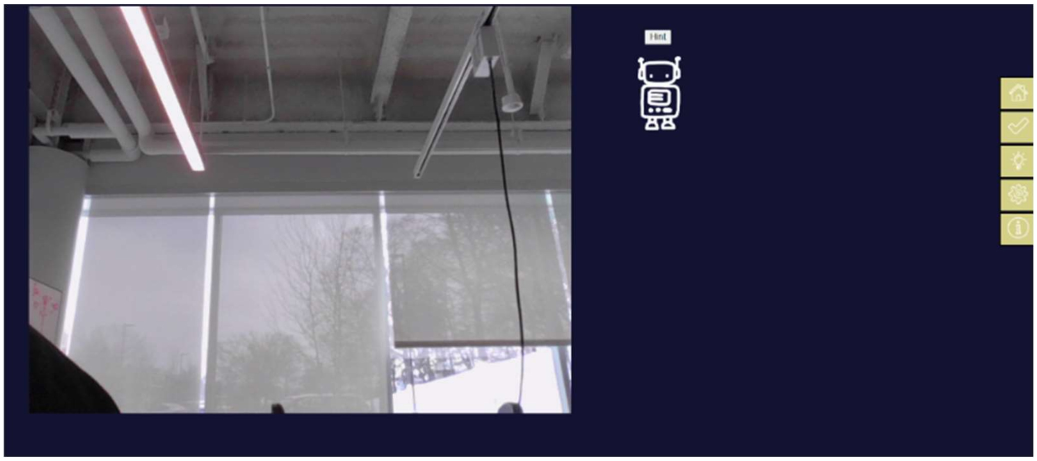


Figure 4.13: 22-23 Website Main Page

Considering this, this year's team wanted to modify the existing system to make it simpler and more customizable for different play experiences.

4.1.3 Modified Website Design

In comparison to the design wizard, the website is the actual means by which a player can interact with whatever story and puzzles the designer has created. For the wizard to be a helpful design tool, the website needs to follow this modular structure in which an inexperienced person can take what they do know and learn how to build a robot escape room.

The previous iteration of the website had been specifically created to follow the narrative and puzzles produced by last year's team, but we wanted the website itself to be a customizable element similar to the wizard that can be developed by someone without a computer science background.

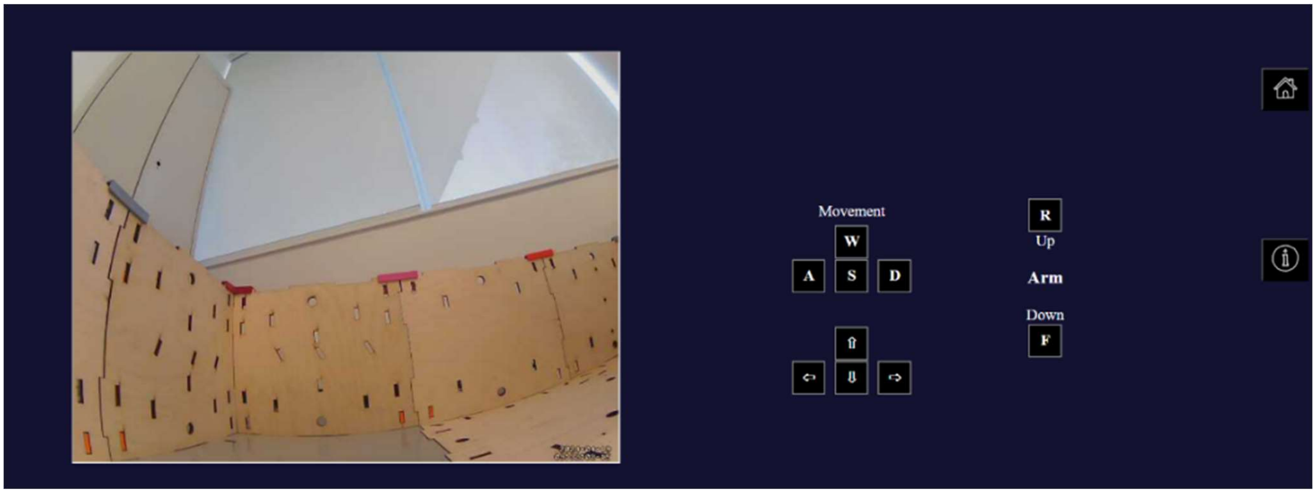


Figure 4.14: 23-24 Website Main Page

We did this by stripping the website of extra pages that were specific to the experience designed by last year's team. We kept some of the base pages and underlying flow, but tried to keep only what was necessary. Once the user has an example of one flask page, it would not be hard to replicate for other pages as needed (Figure 4.13).

5.0 Room Methodology and Design

This chapter discusses the process used to implement the Room design from the 2022-2023 Robot Escape Room team. It will then go over the changes made and the design aspects of the new modular system.

5.1 Last year's Room Design & Changing Priorities

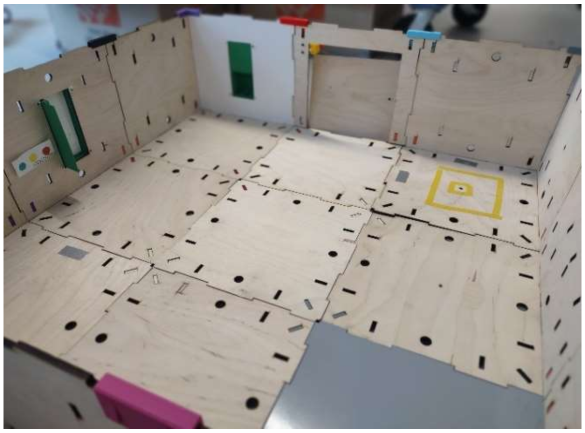


Figure 5.1: An Assembled Escape Room

As shown above, the room design from last year was made up of square floor and wall pieces that could be interconnected in various patterns using plastic clips. As shown in Figure 5.2, there are multiple game pieces that make up the puzzle aspects of the room. There were the door segment, cabinet segment, and the pressure plate segment. Through connecting these in a certain way and putting story aspects on their website the 2022-2023 team was able to make their narrative driven escape room.

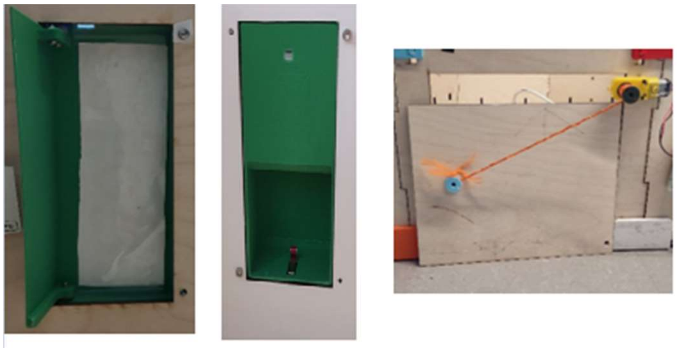


Figure 5.2: Three types of Game Segment. Cabinet Segment (Left), Wall Pressure Plate (Center), Door Segment (Right)

Coming into this project, the priorities for the room design were much different than previous years. Early decisions shifted priorities away from story and designing an escape room and to making a modular system. The initial design with the various segments and possible arrangements was a good starting point, but the room also needed the ability to adapt to the variety of combinations that a game designer could choose. We also wanted to make the room as “plug-and-play” as possible, meaning the user would not need to rewire or reprogram the room.

5.2 Designing the Room System

The door segment and the cabinet segment were chosen as the main game elements that the modularity would be based on. The plan was that each one of these game elements would have a microcontroller attached to it. These would serve as a node for communication via the MQTT system described in section 3.0. Each microcontroller would be responsible for controlling the main game element as well as any attached peripherals such as limit switches or LEDs.

The Raspberry Pi Pico Wireless was identified as a prime candidate for the micro-computer due to its wireless capabilities, inherent micro-python support, form factor, and price. Additionally, this board was 50% to 70% cheaper than alternatives which was the primary motivation to settle on this board.

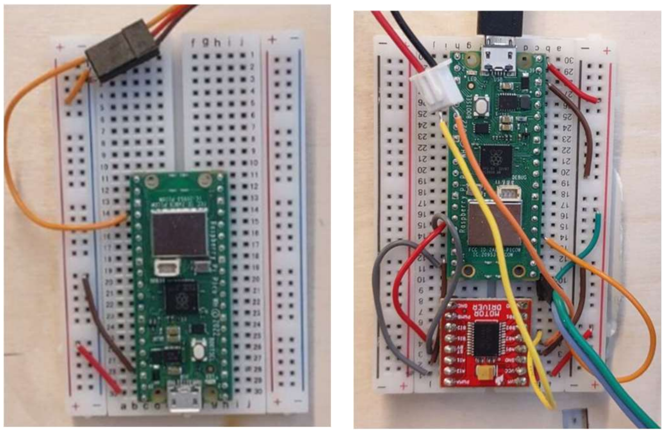


Figure 5.3: Cabinet Pico Design (Left), Door Pico Design w/ Limit Switch (Right)

As shown above, two designs were created using these boards, one for the door segment and one for the cabinet segment. These are called segment controllers. Each of these segment controllers were also coded to control external limit switches which could be attached to select pins on the board. The door segment controller also used a dedicated motor controller to aid in the control of the door motor. Each Pico board is powered via a micro-USB cable with the potential to be completely wireless through a dedicated AA battery pack. Optimally, 3 AA batteries wired in series would provide the 4.5V needed to power the Pico and would make the system entirely wireless. Due to time constraints in ordering this was not done.

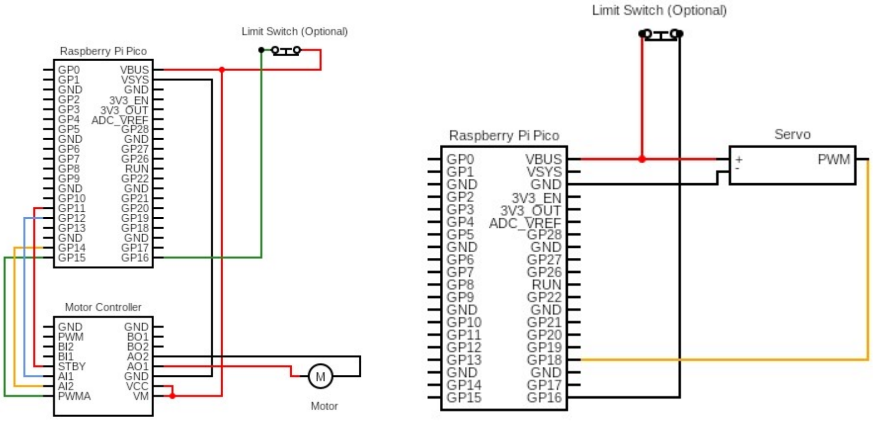


Figure 5.4: Wiring Diagrams for the two segment controller types, Motor Segment Controller (Left) and Cabinet Segment Controller (Right)

5.3 Integrating the Room System

The next step in making the modular room system was to integrate the new segment controllers into the MQTT network described in section 3.1. We set up three stages during which certain actions would take place.

The first stage is when the segment controllers are powered up and connected to the MQTT broker. When connected to power the controller will automatically start the connection process. This is shown visibly by a green LED on the Pico microcontrollers. When the LED is off the controller is not connected. When the LED is solid the controller has connected to the broker and is ready to proceed to the next stage. This will also be shown on the MQTT broker through a publish message indicating the segment controller is waiting for setup. This looks like *room/setup/door_0/status: waiting:door_0,0,0* where *room/setup/door_0/status* is the topic and *waiting:door_0,0,0* is the message.

The second stage is the setup. During this stage the puzzle controller, the script generated at the end of the design wizard, would publish a series of messages to a topic dedicated to each segment controller. An example message would be in the form of *room/setup/door_0: 1,1* . The message is being treated by the segment controller as a comma separated value. The first value is setting an ID for connected limit switch. A value of 0 signifies t no limit switch. The second is a place holder for an LED feature that was not fully developed. Each segment controller receives a message in this format to set it up for the next stage. Once each device receives its setup message, it publishes a message saying that it is connected and identifying limit switch ID and LED type. This looks like *room/setup/door_0/status: playing:door_0,1,0*".

The final stage is gameplay. During this stage the segment controllers wait for the player to trigger a game event. In the current setup, the only game event that can be triggered is the toggle of a limit switch on the pressure plate segment. When triggered, the segment controller will publish a message updating a the "*room/switch_{ID}*" topic with either a "*True*" or "*False*" message. The puzzle controller will read this and publish a corresponding command message as previously determined by the designer using the wizard. A command message tells a segment controller a specific action to take such opening a door or opening the cabinet to specific position.

Both segment controllers come with prebuilt responses to commands that the puzzle controller can send. For the door segment controller these commands come as a comma separate value in the form "*move_type,speed,time*", or either an "*open*" or "*close*". The speed is on a scale of 0.0 to 1.0 and time is in seconds. Both are float values. During certain move types some of the values will be set to 0. An example command would be "*forwardTime,0.5,5.0*". The list of move types and required inputs are shown below.

Move Type	Inputs
forward	speed
reverse	speed
forwardTime	speed, time
reverseTime	speed, time
open	N/A
close	N/A

Figure 5.5: Move Types and require inputs

The cabinet segment controller takes inputs in the form of a position to move to. The predetermined positions are "MAX", "MID", "MIN", "OPEN", "CLOSE". No other input is required.

6.0 Robot Methodology and Design

This chapter discusses the process that was used to get to the final robot product and explains the changes made to last year's robot. It then explains the technical and design aspects of the robot.

6.1 Evaluating Last Year's Robot

The robot design and code of the 2022-2023 Robot Escape Room team offered the core starting point and provided a base to build on for our robot design. Coming into this project the team was aware that they would not have the manpower needed to completely redesign the robot and overhaul the escape room at the same time. As such it was decided that the team would identify key problems to fix or improve so that future projects could have a solid base robot and could instead focus on gameplay aspects. This section provides an overview of the robot received and then will focus on the problems identified with the robot and the solutions used to resolve them.

Shown below is the 2022-2023 robot as we received it.

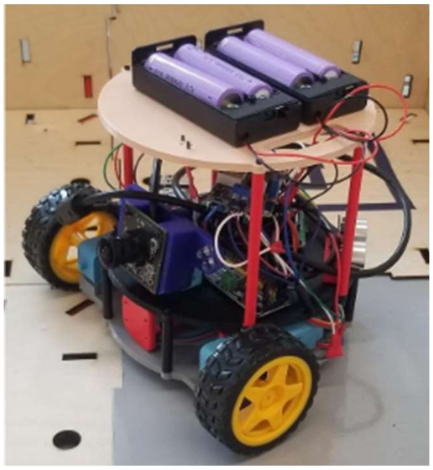


Figure 6.1: 2022-23 Robot

On the bottom layer is the motor driver and the motor controllers. A key point about the controllers is that they have magnetic encoders attached; however, these were not being used. The second layer housed a Raspberry PI 4 which served as the robots control computer. This layer also housed a camera/LED mount, a servo control board, a voltage regulator, and an unused ultrasonic sensor. The third holds two battery housings. Each contains two 3.4V batteries wired in series to output 7.4V. More details about the design specifics and intentionality can be found in the Robot Escape Room v2 MQP paper.

The first thing we did upon receiving the robot was attempt to get it running using last year's code base and hardware. This was a confusing and complex process that required multiple scripts and devices. The robot required two python scripts running, one for robot control and one for streaming the camera and interacting the MQTT server that last year's team used. Additionally, the web stream was only hosted locally so a user had to use windows remote desktop to view the website, camera stream, and control the robot. An exception was that a USB keyboard could be used separately to control the robot. The Raspberry PI was also getting undervolted when running other hardware such as the servos and drive motors. The largest problem that we encountered was network stability and delay issues. These occurred in both the robot controls and most notably in the camera stream. During initial testing, we noticed that the camera was largely useless, and users would instead watch the physical robot due to the large amount of lag in the stream.

The previous team also had plans to implement an arm mechanism to provide more interaction with the game environment and more options for puzzles. To our understanding the implementation was not finished so there was no way for the robot to interact physically with the room and objects within it.

6.2 Robot Solutions

Finding solutions to the identified problems was done by prioritizing a balance between criticality and an estimated time cost. The Raspberry PI not being supplied enough voltage was the immediate priority because it limited the usability of the PI and caused occasional crashes. The second priority was the control lag and simplifying the process to start the robot. These were accomplished together and required the new MQTT system as described in section 3.0 to be completed first. The third priority was fixing the camera lag. This drastically decreased the experience of the escape room and made the user play in unintended ways. The final task was to add an arm mechanism that would allow interaction with the environment and enable a new variety of game options.

6.2.1 Undervoltage

The 2022-23 robot used two banks of two 3.4-volt batteries wired in parallel to supply ~7.4 volt to the robot system. As shown in the wiring diagram in figure 6.2 these ran through a voltage regulator to limit the voltage to ~5.3 to power the Raspberry PI. This was then split to both the motor control for the drive motors and a servo dock which led to the Raspberry PI. The result was the ~5.3 volts coming out of the voltage regulator supplied the PI along with the LEDs, motors, and servos.

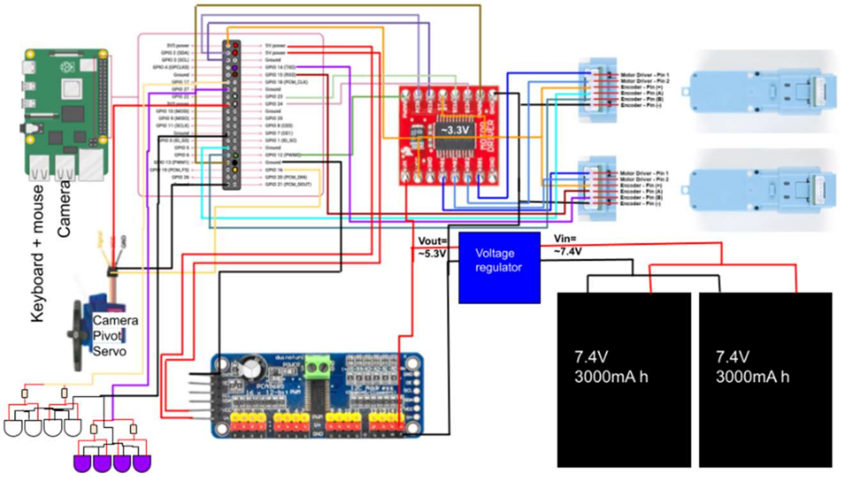


Figure 6.2: 2022-23 Wiring Diagram

Through tests we also discovered that the wrong wire gauge was used to connect the voltage regulator to the Raspberry PI, this caused steady drops in amperage due to wire overheating. This would eventually result in the Raspberry PI crashing and was another source of the PI giving undervoltage warnings.

To resolve these issues, the power supply was split into two, and all peripherals except the LEDs were rewired to run off one. Additionally, all wires in the power supply system were switched to 22 AWG which removed the overheating wire problem. This is shown in figure 6.3. The PI being given a dedicated power supply allowed more stable voltage and in combination with the rewired power supply resolved the undervoltage problem.

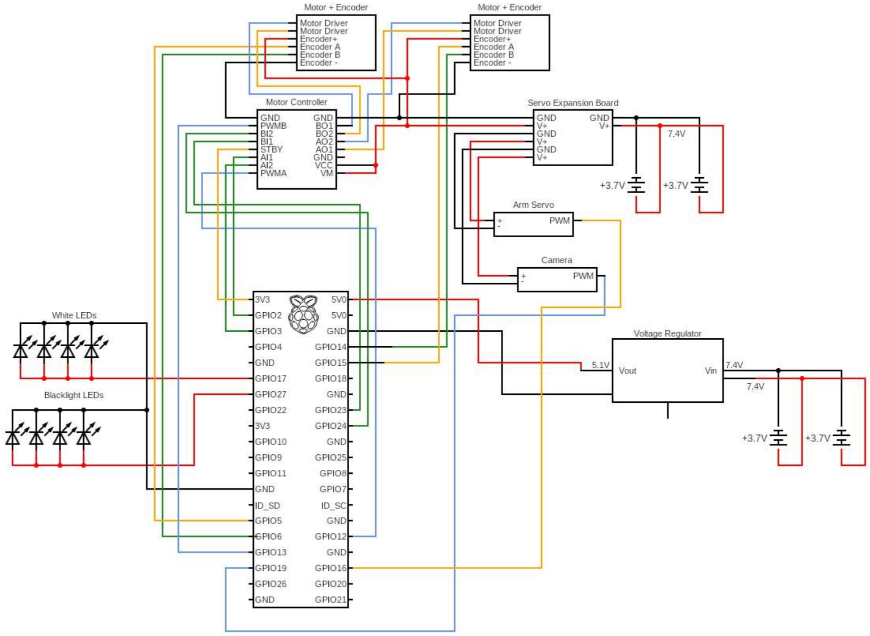


Figure 6.3: 2023-24 Wiring Diagram

6.2.2 Robot Controls

As previously described, the 2022-23 system used a remote desktop connection to the PI to control the robot and view the video stream. This required running multiple scripts and knowledge of the Windows and Linux command line systems as well as knowledge of the Raspberry PI's IP address. This was a major problem as non-technical usability was important to the project.

Using the MQTT server implemented, as described in section 3.0, the controls of the robot were able to be transmitted over a wireless local area network. This was accomplished by creating two python scripts, one for the robot and one for the user. The user script would publish messages to specified topics that the robot script subscribed to. When the user presses a button to actuate or move the robot and the corresponding message was published. The robot script would immediately recognize that a do action specified. An example of a message sent by the user script is *robot/motion: forward* where *robot/move* is the topic and *forward* is the message.

Advantages of this system are less latent controls, easier setup, and more abstraction of robot controls. Since the robot script subscribes to control messages from the MQTT broker and not a local program there is also a lot of potential for feature development. It is now possible for the puzzle controller to move the robot based on game events or interact and modify with the control system in a variety of ways. This gives rise to a lot of potential game applications that would help make the game more complex and provide a more unique experience.

6.2.3 Camera View

In the 2022-2023 Robot Escape Room project the robot's camera view was hosted locally on the robot which was then viewable by the user through a remote desktop connection. Since removing the remote desktop requirement was a goal there needed to be a new approach to getting the video stream to the website.

Motion is a Linux program the is designed to monitor cameras and detect motion. This program is also able to stream the camera view to a wireless local area network. This feature was used to publish a jpeg stream to the network which is accessible to the user through the website. This method drastically reduced the lag and made controlling the robot through solely the camera view feasible.

6.2.4 Arm Mechanism

The decision was made to move away from the arm designed by the 2022-2023 Robot Escape Room team. Due to the limited manpower available and the difficulty of accurately programming and implementing a multi-depth of field servo actuated gripper continuing this design was deemed unfeasible.

Multiple different designs of manipulators were considered, each with their own advantages and disadvantages. A forklift mechanism would provide a wide end effector area, a large range of motion, and a relatively large load capacity; however, it would require gearing and additional CAD modeling. A servo articulated arm as designed during the 2022-2023 Robot Escape Room MQP offered good control; but was ruled out due to its complexity. A four-bar is a system of four linkages that can be designed to move between specified points by modifying the lengths of each linkage. This type of design was chosen due to the ease of design and the simplicity of coding and actuating it. Since moving one linkage moves the entire system, the entire system could be run using a single servo motor. This design would allow the robot to actuate the limit switch and pick up objects placed in the room while being time-efficient and cheap to make.

Using a pre-built kit, we designed our four-bar in such a way that it could be used to both pick up and move objects in the room as well as hit the limit switches in the wall pressure plate game segments. This was mounted inside the robot's body such that the end effector is visible to the camera view and the width of the robot remains unchanged. The mounted four-bar is shown below.

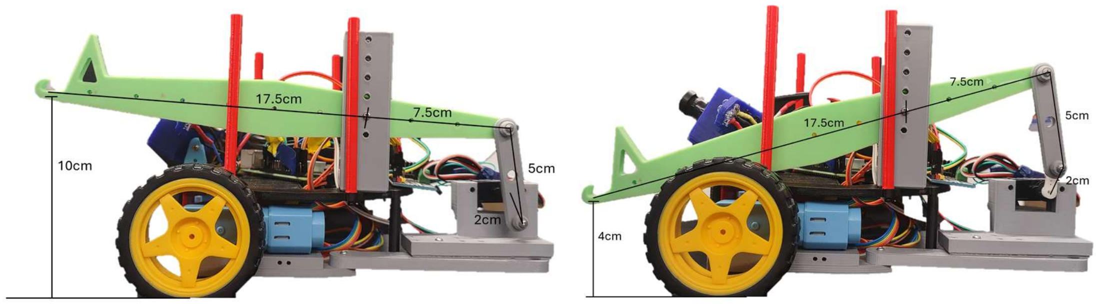


Figure 6.4: four-bar in up position (left) and down position (right)

The four-bar is made of three connected links that are attached to 2 origin points on the robot. One origin is on the frame of the robot while the other is on the servo motor. These links are 2 cm, 5 cm, and 22.5 cm long. The long linking is divided into two parts by an attachment to the frame. This attachment serves as a fulcrum which allows the end effector (hooked end) to move up and down. The end effector has a 6 cm range of motion, enough to reach into the cabinet segment.

7.0 Evaluation through Playtesting

The playtesting process was necessary for testing if our design was usable and intuitive by someone unfamiliar with the project. Before playtesting began the team outlined a plan for the experience we wanted to give users, which ended up being a thirty-minute session in which we conducted a five-minute pre-interview, gave participants twenty minutes to interact with the escape room design wizard, followed by a five-minute post-interview. Participants were given the following prompt: “Please design an escape room that a human-controlled robot can complete and talk us through your thought process.” The pre and post-interview questions we have can be found in Appendix B. This style of playtesting has three main research methods: qualitative data in the form of open-ended questions and discussion, observing participants during the session and encouraging them to think aloud, and reviewing the final output of the website when the session is over.

We designed the experience to last thirty minutes because we wanted it to be something simple, but also could be built upon in multiple sessions if we had gotten to the point where the room could also be assembled according to user instructions. In the design wizard the participant tested, there are three pages: the room layout page, the nodes puzzle page, and the final output page. The interaction between participants was on a laptop we provided, and the only input needed was via keyboard and mouse.

Through playtesting we were able to get specific feedback about how intuitive the designer interface was and additionally were able to identify some bugs. The main problem we ran into during our playtests was an issue where the puzzle progression nodes would not save when moving between the pages, so if you went back to check the room layout or add more wall types you would have to start from scratch. We were then able to fix this for the final implementation. Also, there were some visual issues with the pictures for the room layout pieces and the final room page listed the connectors needed but didn’t have any visual representation for each piece. Overall, most of the improvements needed were small changes and most people enjoyed the fact that the interface was easy to understand, and the controls were also relatively simple with only left and right clicks needed.

8.0 Challenges

While throughout the project we achieved a great deal, there were times of struggle. Overall, we learned a lot of lessons during the making of Robot Escape Room v3. This section covers a few of the big takeaways that we as a team got from this project.

8.1 Importance of Communication

Communication was a surprising problem that we encountered during the project that we did not expect due to our small team size. Notably, technical communication was an area we needed to improve on because the project was split into such distinctive parts. These parts separate our work, but due to the nature of what we were trying to accomplish all the parts needed to intercommunicate. This became a problem during testing the puzzle controller and the segment controllers. Each of these were developed separately by different team members so when it came time to test, we encountered failures. Commands and process were either not communicated or were not clear. While this did not stop our team from completing our objectives, it did take additional time to resolve these issues.

The solution to this problem would have been implementing a clear system of communication for technical details as well as maintaining updated documentation. This would have allowed team members to know that the information they were working with was up to date.

8.2 Design Limitations

As seen through the background research, escape rooms are a complex concept that involves behind the scenes work and things falling together in just the right way to create a compelling experience. Some of these things are hard to achieve for a robot escape room such as player teamwork, establishing a setting and context for the story, and having a complex narrative. Things that are commonplace in a normal escape room like red herrings, clues that don’t help the player progress through the game and are meant as a distraction, are harder to achieve with that layer of separation between the user controlling the robot and visualizing everything through the camera stream on the website. Since the room is not equipped with extra visual information outside of the different wall pieces for different puzzle types, there is not currently an opportunity to set up these distractions.

9.0 Next Steps

At the end of our project, we had accomplished a lot, but there were a lot of features and ideas that we did not have time to design or implement. In this section we will cover the next steps that this project can take.

9.1 Website

The website did not get worked on as much this year, because our primary focus was the designer. However, we did condense a lot of the unnecessary pages that were specific to the experience crafted by last year's team and modified the pages to be more modular. For future implementations on the website, the team imagines there could be some sort of webpage generation based on the information output by the design engine after the user goes through that process.

Another future implementation that would also correspond to the hardware is to establish direct communication between the website and the MQTT server for text input and potentially other types of input. An example of this would be a puzzle that requires a password or keycode input to solve it.

In running the website, there are two python scripts that need to be run, the Flask app and the robot control script. Currently they must be run separately to have both run at the same time. This could be improved by implementing multiprocessing, an avenue we had started to investigate, but were unable to properly implement.

A large part of the user experience is the visuals of the room, and for a modular room that can become other escape room experiences, a physical set dressing for one game would be a large amount of effort. However, if an AR overlay could be created for a room and applied to the video stream of the robot, that would vastly increase the immersion an escape room could provide.

A shortcoming in our project was we did not fully adhere to the web accessibility guidelines, such as alternative text or audio descriptions for elements in the webpages. In the future, implementing measures to address these guidelines would be useful for those who are sight impaired. Beyond those, the main response from the robot being a video stream is not accessible to those visually impaired, so audio responses to robot and room interactions could be another consideration for accessibility.

9.2 Engine

The designer itself could be modified in many ways depending on the context of how it is being used, but the team has brainstormed some modifications that can be made in the future to improve upon the existing product. Some of these would be relatively easy to do but are nicer to have than necessary, while some of these would be more involved and require more time than is in scope for the product. The biggest problem the team ran into when planning out this designer engine was to use this technical website to produce a more creative output in the form of a Robot Escape Room.

The main improvements that can be made on the design engine website are adding more visual options to denote objects on the page such as color coding or numbering of the objects in the room. An example of this would be a red pressure plate opening a red doorway. There could also be functionality added to the room layout page where the user can mark directly on the room what pressure plate opens what panel or doorway. Another idea that could be implemented is controls that allow the designer to select an action for a piece such as opening or closing a door and activating or deactivating a pressure plate. This layout would also be useful for the puzzle progression page to allow the designer to easily see where pieces are without switching between pages. Another improvement would be dynamic resizing of the room, adding another row or column when the designer reaches an edge, would remove the arbitrary constraint of the 5x5 maximum dimensions currently imposed. However, there would need to be some consideration for the legibility of the layout once it is at the final page or included in a file for another person to assemble the room, so an overall maximum may still be needed.

Additionally, for improvements to the puzzle progression page there could be options for more dynamic node creation and organization, such as the user being able to click and drag nodes to reposition them on the screen. Allowing the designer to choose how the room outputs respond to an input instead of being locked into a set output, for example a panel could be made to open or close instead of just opening. That idea could also be applied to the room inputs, for example the designer chooses that something occurs when something is placed into a wall pressure plate or when something is removed from a wall pressure plate. Furthermore, akin to room inputs and outputs, website and robot inputs and outputs could be implemented, website features encompassing effects like displaying text or applying a screen shake, and interactions like password inputs and keypad entry, while robot features would encompass effects like slowed or erratic movements. As more inputs and outputs are made implemented into the designer, a concise, general-purpose node may become cluttered and messy for the user interface, and creating multiple node types based around which facet of the experience the designer is interacting with would become necessary.

9.3 Room

9.3.1 Segment Additions

Currently, there are only 3 game features that make the puzzle part of the room. The next step to making more complex and involved escape rooms is to develop additional features that can be put into the room. Some of these can be implemented via just code while others would need physical additions. A list of suggestions is shown below.

Addition	Description
Limit Switch Pattern Reader	The ability to recognize patterns via toggling the limit switch. Could be morse or other patterns.
Multi-position switch	A system that returns the value of a multi-position switch.
Floor Servo	Servo the makes a wall go up/down
LEDs	Patterns or colors signifying something, toggle on/off/color
Floor Limit Switch	Limit switch placed on the floor

Figure 9.1: Ideas for Room Additions

9.3.2 Core Redesign

The room is not easily assembled or disassembled, as it was designed for a single experience. Creating a new design would help with the accessibility of the room assembly, and considerations towards not requiring multiple people to flip the room during the process or having connections that are a difficult to take apart, would also decrease frustrations a user might have if they make a mistake during the assembly. Additionally, if pieces could be switched out without major disassembly of the surrounding room, that would be beneficial to the design process, as sometimes there may be issues only realized with the physical room.

More flexibility in room design could be implemented with a rework of the room build if the new system allowed for more than two walls to connect at an intersection. This could lead to multi-room escape rooms or tighter designs with nooks that do not need a three-foot-wide arch for a U-turn. An issue with this would be the Raspberry PIs' location that controls room pieces when those would be placed as a dividing wall, or pieces that have protrusions on the back side to accommodate motors and switches.

9.3.3 Wireless Power

The segment controller currently needs to be powered via a micro-USB connection. We recommend that this is switched to a battery pack containing three AA batteries. Each AA supplies 1.5V of power so when wired series they would provide 4.5V, enough to power a Raspberry PI Pico.

9.4 Robot

9.4.1 Kinematics and Automated Control

With the implementation of MQTT to all aspects of the escape room system there are a lot of possibilities. Since the robot receives base movement commands over MQTT it is possible for any MQTT client to send commands, not just the user's control script. We recommend implementing multiple features from this idea.

The puzzle controller could have preset movements that happen during game events and could be part of a story. This could be something like the robot spinning to simulate being grabbed. These could be preprogrammed in and added to the game design via the design wizard.

The puzzle controller can also communicate with the user control script. This could make game events modify controls. Controls could intentionally stop working or even be reversed.

The robot could also have a variety of movements preprogrammed. The robot currently has functional encoders that are not being used. These in combination with the camera or an ultrasonic sensor could be used to navigate the room autonomously.

Overall, we recommend developing features around the MQTT system and robot control.

9.4.2 Improved Servo Control

The arm and camera servo controls are currently coded to move based on duty cycles. This causes stuttering when moving small increments. We recommend switching to a servo library that abstracts this and may resolve the stuttering issue.

9.4.3 Code Reading and Vision Expansion

Currently, the camera is only used to provide the players with a first-person perspective of the robot. We recommend implementing a visual reader that can read visual codes such as april tags or QR codes. This is a feature that offers a lot of potential in navigation and game design along with multiple avenues to increase user interaction.

9.5 Backend System

9.5.1 Automatic Address Finding and Network Interchangeability

The IP addresses are currently setup as hardcoded values that work with our current router and wireless setup. This does not suit all use cases and work should be done such that the segment controller, robot, puzzle controller, and user controller scripts can connect to the broker without knowing beforehand what local IP address the broker will be on.

9.6 Classroom Focused Modifications

A goal for the project that was developed towards the end of our time was to make it so that the Robot Escape Room could be modified from an MQP project to something that can be used in IMGD classes here at WPI. These lead to us needing to change our mindset and think about how this would and would not work for a college classroom environment.

The biggest consideration in adapting this project to work for a class specifically in the IMGD department would be that it would be valuable for the students not having to worry about the coding or technical aspects of the project, instead being able to focus on the creation of the room aspects such as story and puzzle design. This was already one of our goals at the beginning of this process, so to get to this point the project would not have to have much work done.

One of the challenges that comes up is that there is still a period of predesign work that would need to be done before the creation in the wizard itself and this would need to be scaffolded for a class. However, some amount of effort will need to be made by the professor in order to adapt this to whatever class it is being taught in, so this is not as big of a hurdle as it might appear.

10.0 Conclusion

We have created a robot escape room system that allows a user to design a custom escape room then make and play their escape room with little technical knowledge. After spending the first term understanding and getting the existing system to work, we were able to determine our personal goals for the project and where we wanted to go next. Development was focused on three main objective areas: the design wizard, the room, and the robot.

We created a design engine where users can design and implement their own robot escape room using the room layout tool and puzzle progression page. For future iterations, we recommend adding more visual indicators for objects, along with more text and audio features.

We designed and implemented a system to make the room modular such that a user can easily assemble a variety of designs. There remains a lot of possibilities with this system that can make the escape room much more engaging, challenging, and fun. We recommend creating more segment types and removing the need for wired power.

We also iterated on the robot to provide the features needed to allow a player to escape the room. Through identifying and solving a variety of problems we have also created a platform that can be expanded on. We recommend the work is put adding new features such as automated control and code reading. There is also room to improve the control of servos and to replace the arm with a better manipulator design.

Overall, the Robot Escape Room v3 team was able to provide a unique take on the Robot Escape Room MQP series that leveraged our strengths. We created an escape room infrastructure that can be iterated on in future MQPs or classes.

11.0 Appendices

Appendix A: GitHub Link

Located in this GitHub link is the code and documentation regarding the programming aspects of the room, robot, design wizard, and website. There are also instructions regarding setup.

https://github.com/NInjaQuarrior/RobotEscapeRoom

Appendix B: Playtesting Questions

- Warm-Up Discussion (5 min)
    - What is your experience designing games?
    - What is your experience playing escape rooms?
    - What is your experience designing escape rooms?
- Prompt for the user (20 min): "Please design an escape room that a human-controlled robot can complete and talk us through your thought process."
- Discussion of Experience (5 min)
    - What did you dislike about the experience overall?
    - What was uncomfortable about the experience?
    - What would you want to implement for future experiences?
    - What were some interface specific problems that you thought could be improved upon?

12.0 Bibliography

"Accessible Web Design." Accessible Web Design | DO-IT, www.washington.edu/doit/accessible-web-design. Accessed 10 Oct. 2023.

"Adaptive Accessibility." W3C Workshop on Web Games Position Paper: Adaptive Accessibility, www.w3.org/2018/12/games-workshop/papers/web-games-adaptive-accessibility.html. Accessed 10 Oct. 2023.

Anderson, Eike Falk. "The Case for Research in Game Engine Architecture." Proceedings of the 2008 Conference on Future Play: Research, Play, Share, 23 Nov. 2015, www.academia.edu/191806/The_Case_for_Research_in_Game_Engine_Architecture.

Ascalon, Anecia. "The History of Escape Rooms (2021): Blog: The Escape Game." Cosmo Waving, The Escape Game, 22 June 2022, theescapegame.com/blog/the-history-of-escape-rooms/.

Bogue, R. (2022). The role of robots in entertainment. Industrial Robot: The International Journal of Robotics Research and Application, 49(4), 667–671. https://doi.org/10.1108/IR-02-2022-0054

Buckingham, Owen, Edward Matava, Alyssa Moore, and Matthew Nagy. "Robot Escape Room," April 27, 2022. https://digital.wpi.edu/pdfviewer/b2774000v.

Clune, Nathan, Liang Lu, Olivia Bell, Grace O'Reilly, Lillie DeHaemer, and Zachary Sarrett. "Robot Escape Room V2," April 25, 2023. https://digital.wpi.edu/concern/student_works/8c97kv108?locale=en.

Exit the Room. "A Brief History of Escape Rooms." A Brief History Of Escape Rooms | Exit The Room, https://www.exittheroom.com, 31 Mar. 2022, www.exittheroom.com/blog/a-brief-history-of-escape-rooms.

Gregory, Jason. “Game Engine Architecture, Third Edition, 3rd Edition.” O’Reilly Online Learning, A K Peters/CRC Press, learning.oreilly.com/library/view/game-engine-architecture/9781351974271/. Accessed 10 Oct. 2023.

“Guidance on Web Accessibility and the Ada.” ADA.Gov, 6 Oct. 2023, www.ada.gov/resources/web-guidance/.

Kato, Takao. “Thoughts on Real Escape Game.” REAL ESCAPE GAME Created by SCRAP, 8 Aug. 2016, realescapegame.com/thoughts-reg/.

Moore, Alyssa, et al. *Robot Escape Room.* : Worcester Polytechnic Institute, 2022.*MQTT - The Standard for IoT Messaging*. (n.d.). MQTT. Retrieved April 22, 2024, from https://mqtt.org/

Sayers, C. (1999a). Basics. In C. Sayers (Ed.), Remote Control Robotics (pp. 9–24). Springer. https://doi.org/10.1007/978-1-4612-1430-4_2

Sayers, C. (1999b). Introduction. In C. Sayers (Ed.), Remote Control Robotics (pp. 1–8). Springer. https://doi.org/10.1007/978-1-4612-1430-4_1

SCRAP Entertainment. “About Scrap” REAL ESCAPE GAME Created by SCRAP, 10 Oct. 2020, realescapegame.com/about/.

“Section508.Gov.” Home, www.section508.gov/content/guide-accessible-web-design-development/. Accessed 10 Oct. 2023.

“The Impacts of a Modular Design Strategy on Modern Video Games.” Vdocument.In, vdocument.in/the-impacts-of-a-modular-design-strategy-on-modern-video-essay-versionpdf.html. Accessed 10 Oct. 2023.

(WAI), W3C Web Accessibility Initiative. “Designing for Web Accessibility – Tips for Getting Started.” Web Accessibility Initiative (WAI), 10 Oct. 2023, www.w3.org/WAI/tips/designing/#main.

(WAI), W3C Web Accessibility Initiative. “WCAG 2 Overview.” Web Accessibility Initiative (WAI), www.w3.org/WAI/standards-guidelines/wcag/. Accessed 10 Oct. 2023.

Wirtz, Bryan, et al. “How to Make Your Own Game Engine (and Should You?).” Video Game Design and Development, 7 Sept. 2023, www.gamedesigning.org/learn/make-a-game-engine/