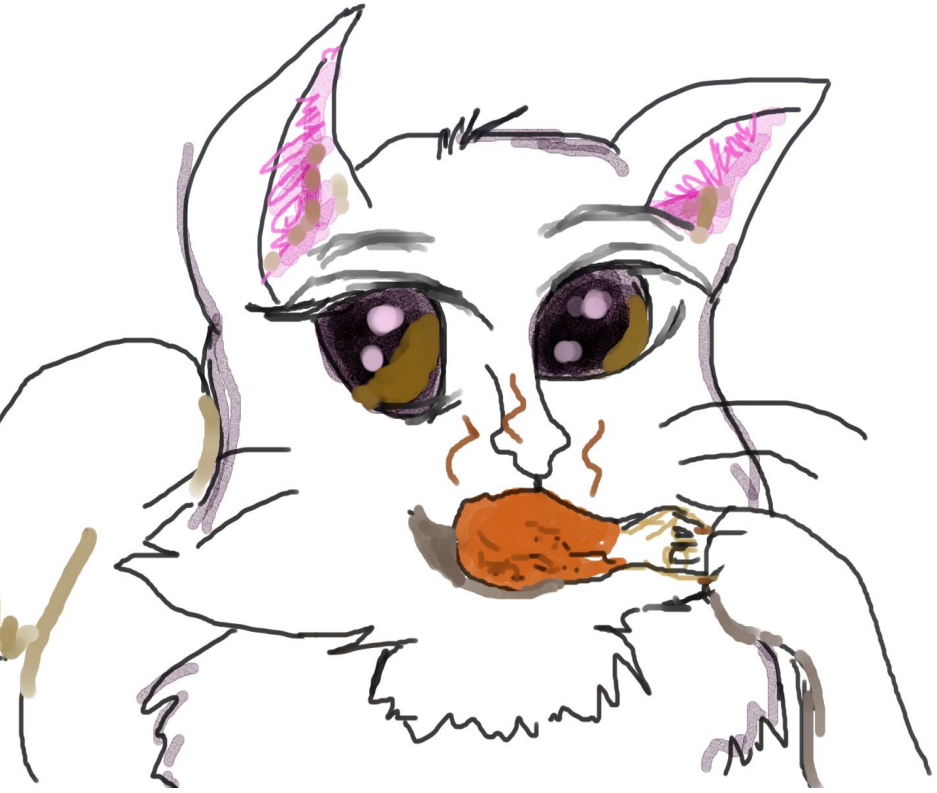
MAKI Studios



Long ago...

Cats battled to the death over magical chicken wings in the sky

And then they brought the chicken wings to the humans…

This sparked the creation of a new restaurant, called

MONOLITH

Our Goal

The goal of Monolith is to teach the components of making a proper meal amidst the busy-ness and cha of life by setting objectives that the player has to rea and allowing them to interact to complete them.

MAKI STUDIOS

Our team:

KHADEEJA

- Sound effects and music connoisseur

IBRAHIM

Dishwasher feature designer/animator

MAURICIO

Food and game progression designer

NAGA

Player movement and animation

BIBEK

Screen text, Menu Help, User Interaction

ALEX

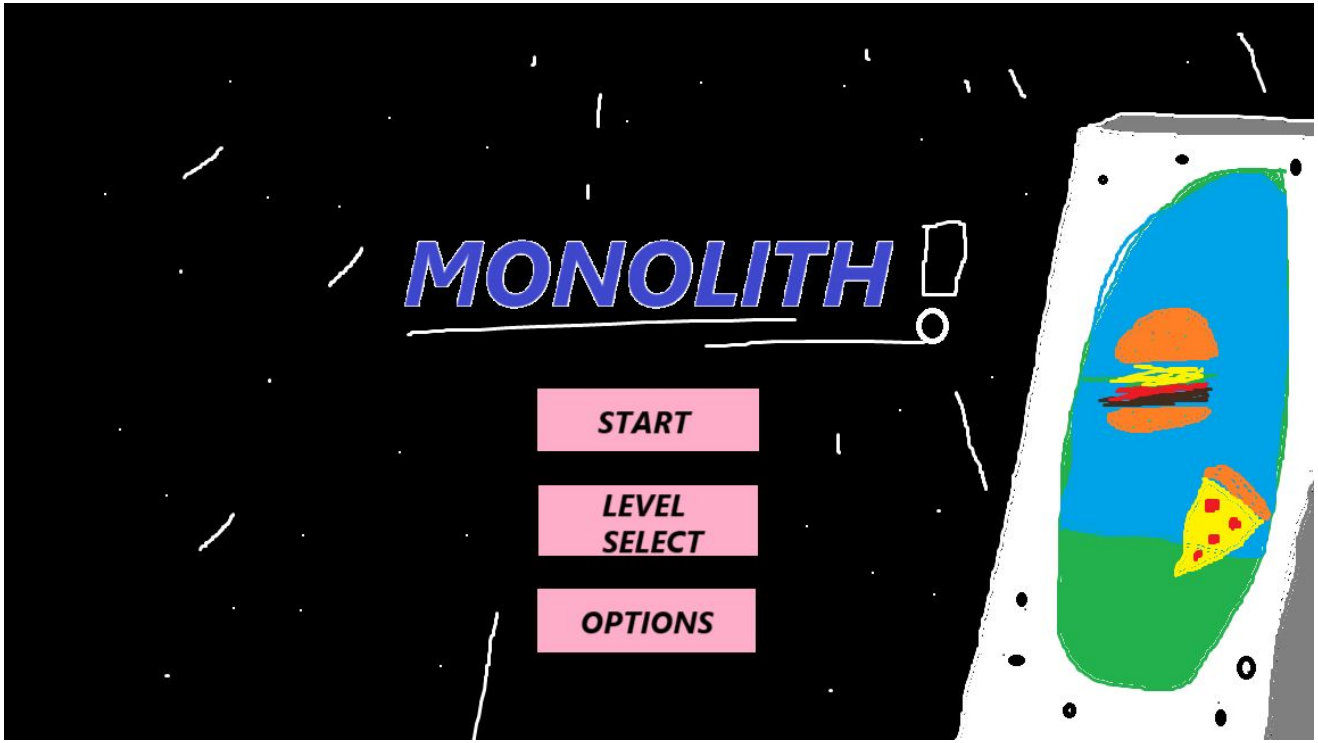
Customer character designer/animator

JENNIE

Level designer

MAKI STUDIOS

SCENE 1: TITLE SCREEN

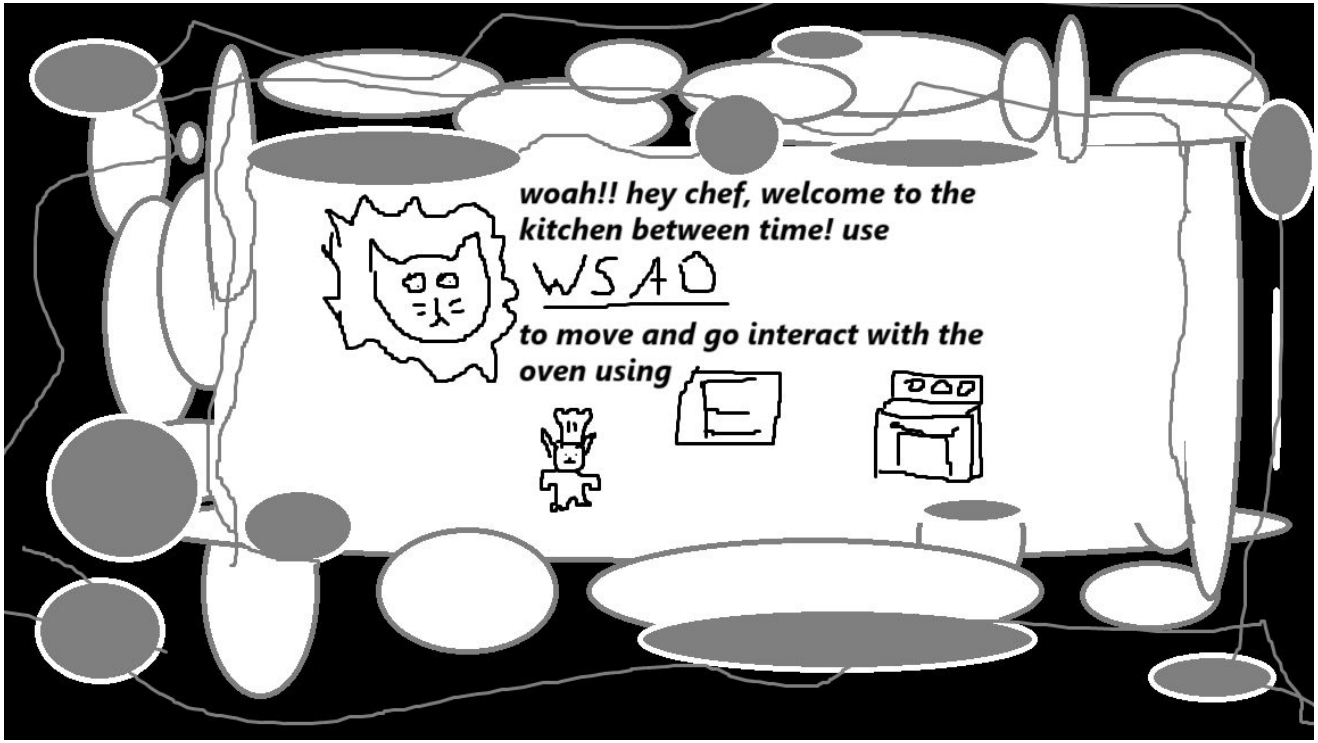


Action: Select to start, browse levels, or change options

Notes:

Potential animated background and music

SCENE 2: SHORT TUTORIAL

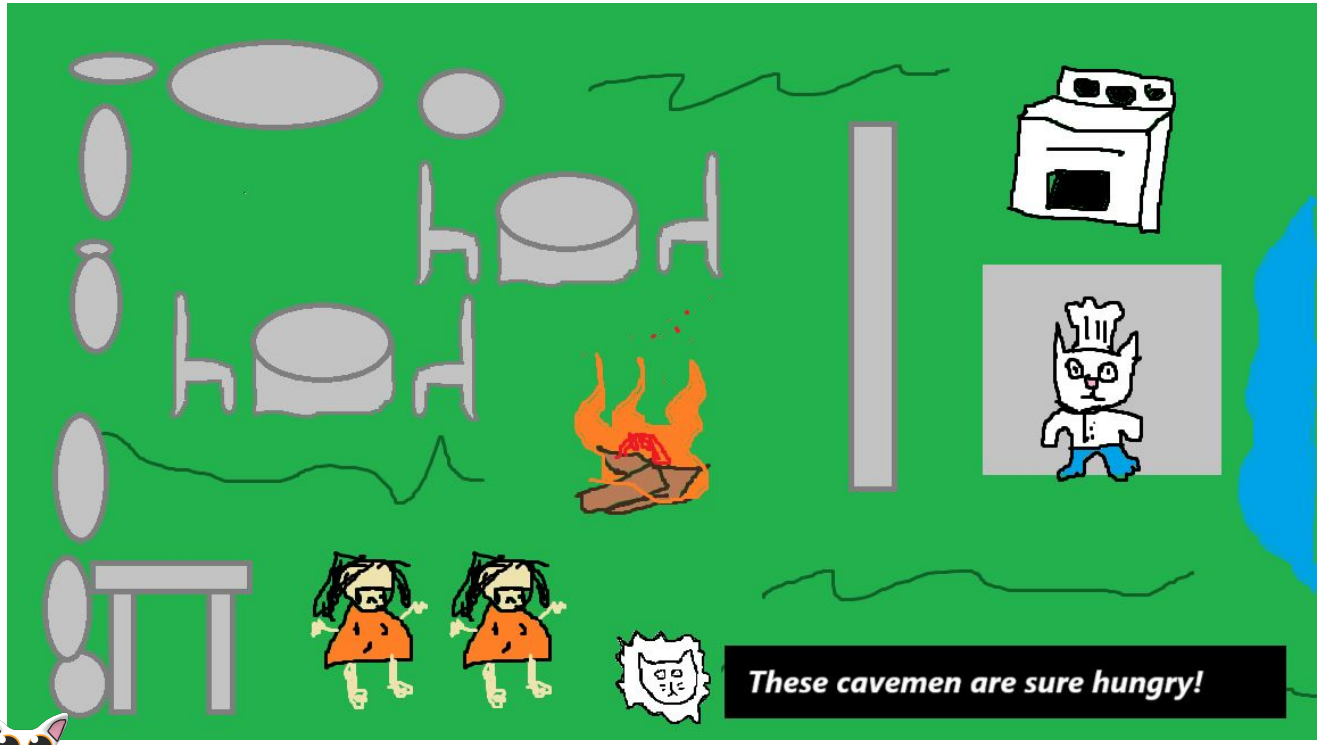


Dialogue: Cosmic cat guides player

Action: Player is in control, learns mechanics

Notes: Top down perspective. Player is small cat chef!

SCENE 3: FIRST STAGE (STONE AGE)

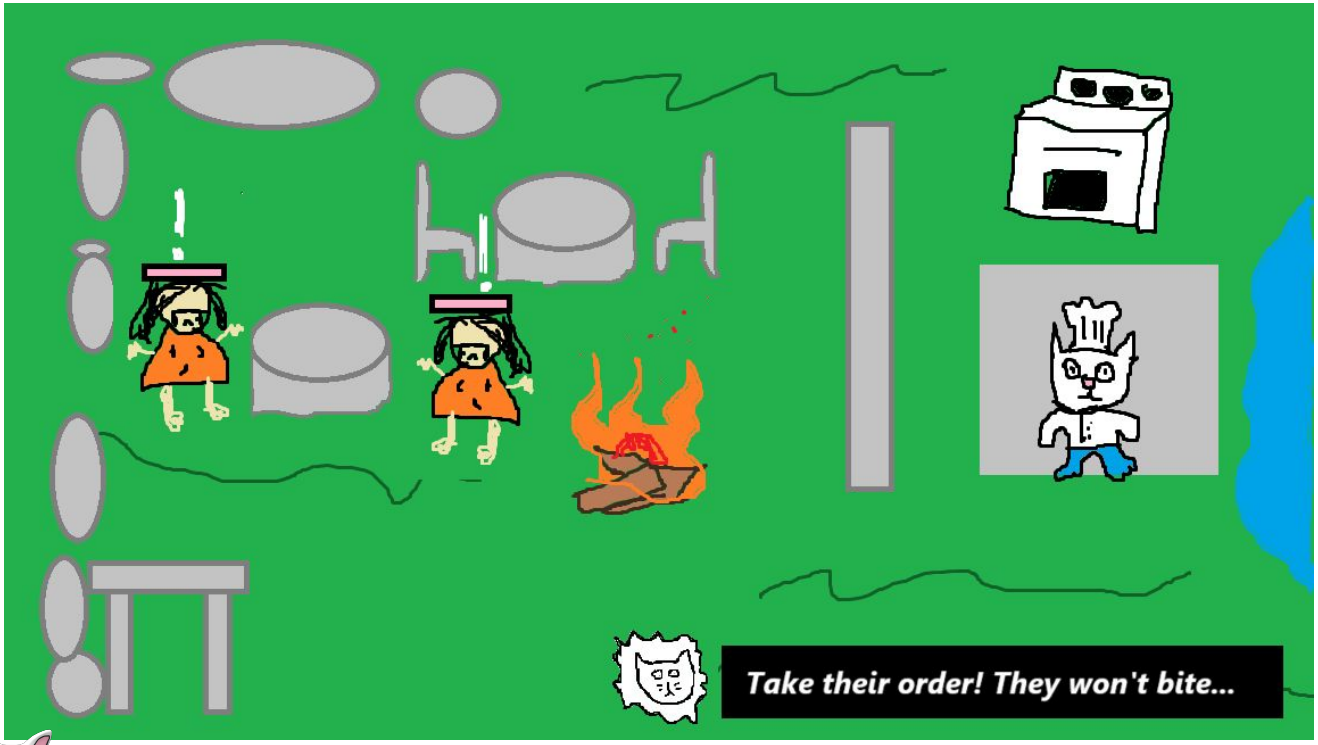


Action: Scene is loaded, player has control and can move an interact with environment

Notes: Hints from cosmic cat. Music and systems implemented in stage

MAKI STUDIOS

SCENE 4: CUSTOMER SEATING



Action: Customers generated sit and wait for player interaction. Satisfaction meter determines scoring and drains slowly

MAKI STUDIOS

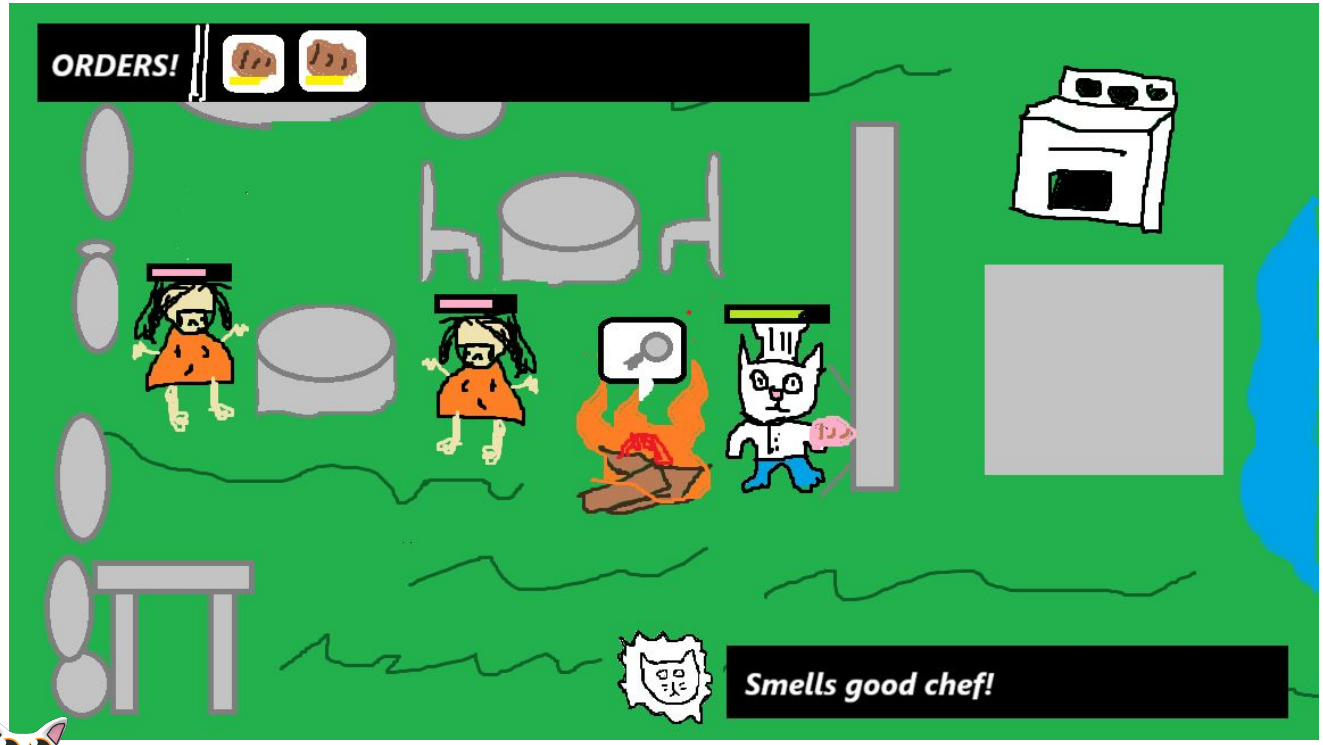
SCENE 5: TAKING ORDERS

Action: Player interacts and orders appear. Customers continue getting impatient

Notes: Impatience is show on order and on customer

MAKI STUDIOS

SCENE 6: GATHER / PREPARE FOOD



Action: Player gathers ingredients and places food at cooking station to let cook

Notes: Progress meter above player to show cooking progress. Customers bars decreasing with time.

MAKI STUDIOS

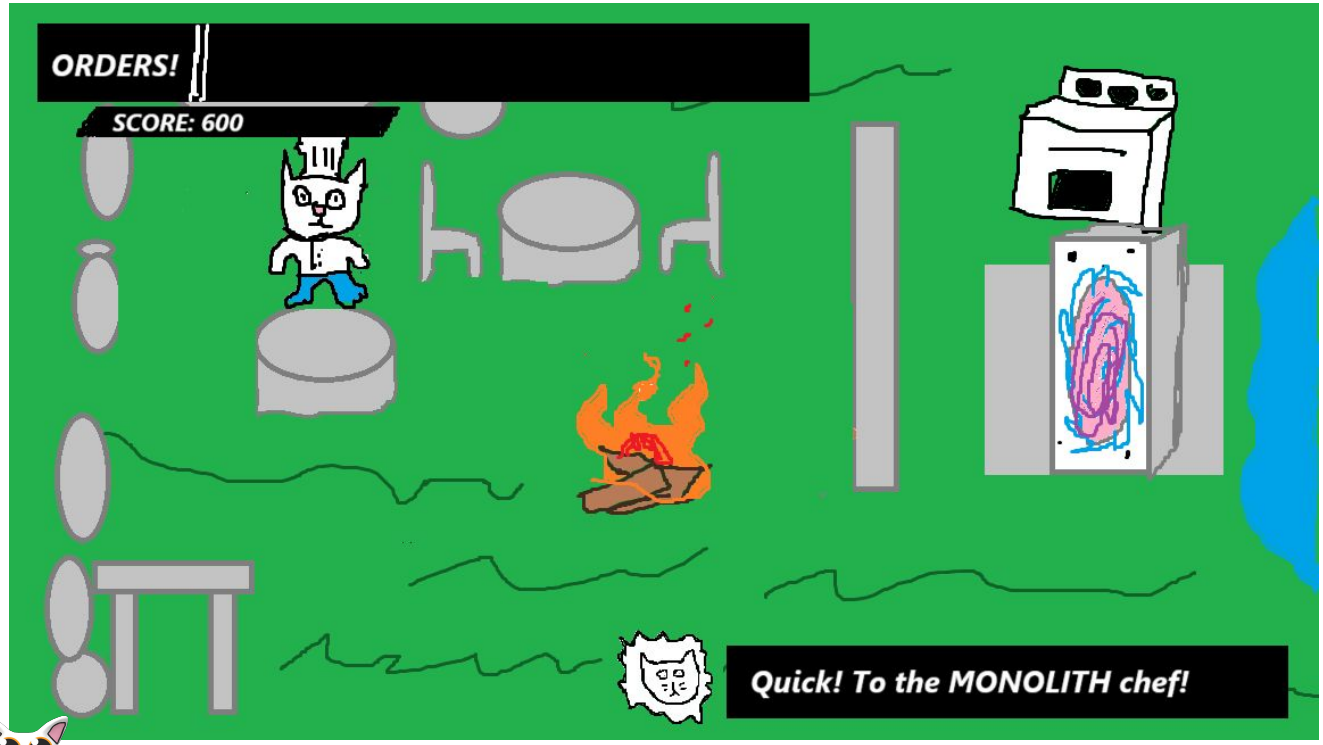
SCENE 7: SERVING FOOD

Action: Player interacts with customers after the dish is prepared. Gain points for correct order

Notes: Customer will penalize players for taking too long or serving the wrong order

MAKI STUDIOS

SCENE 8: LEVEL PROGRESS

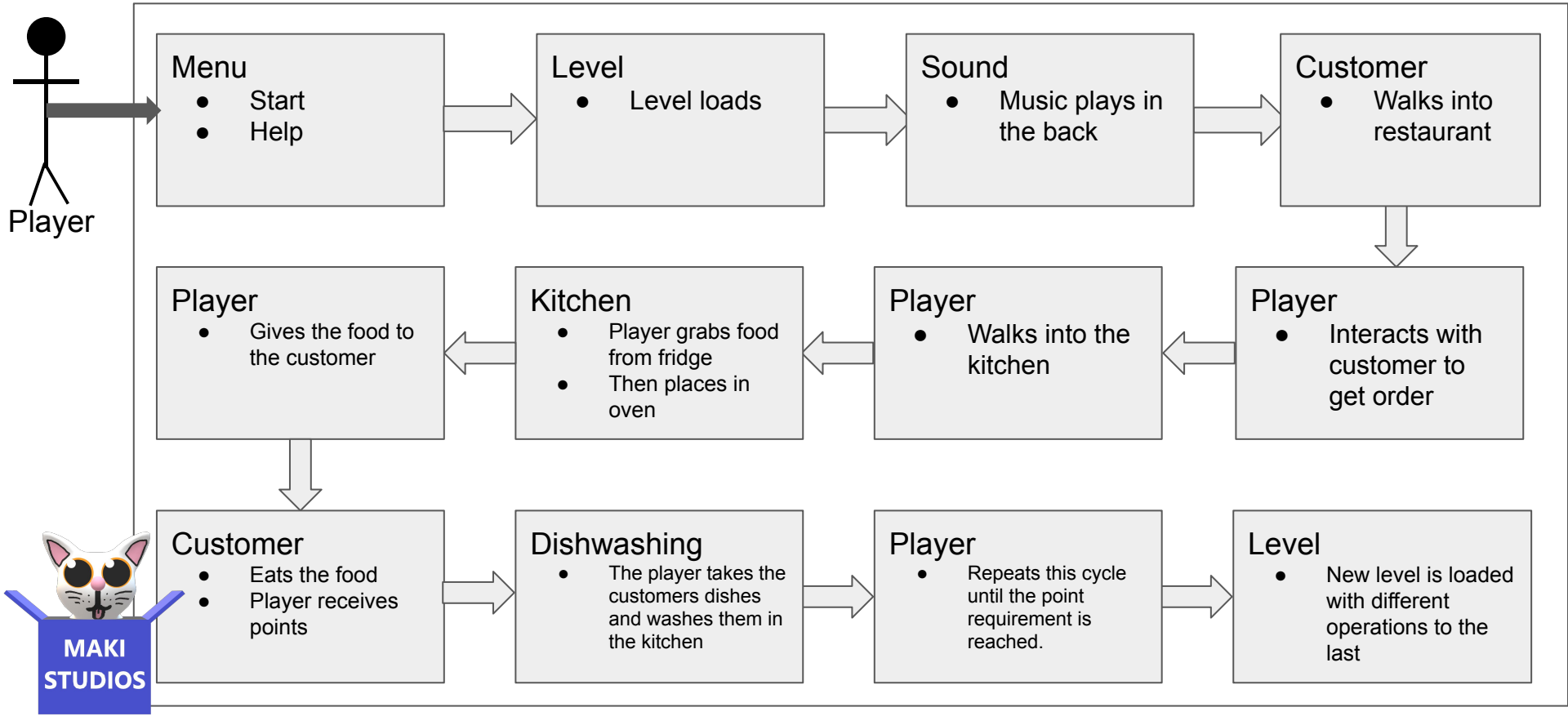


Action: Player can enter the portal that appears to go to the next stage. Game continues by stage until the end

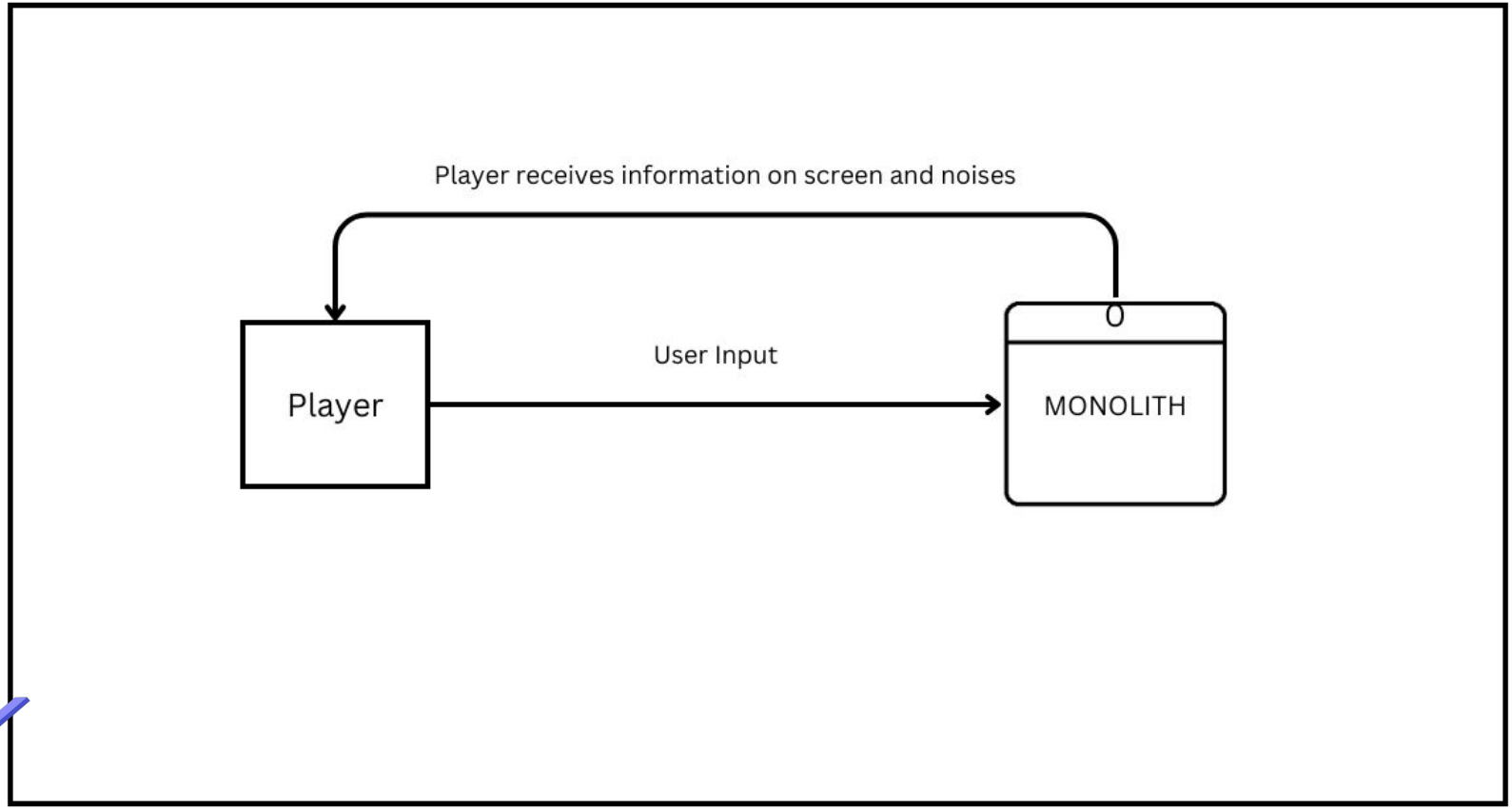
Notes: Players must survive the shift / score enough to progress

MAKI STUDIOS

Global use diagram

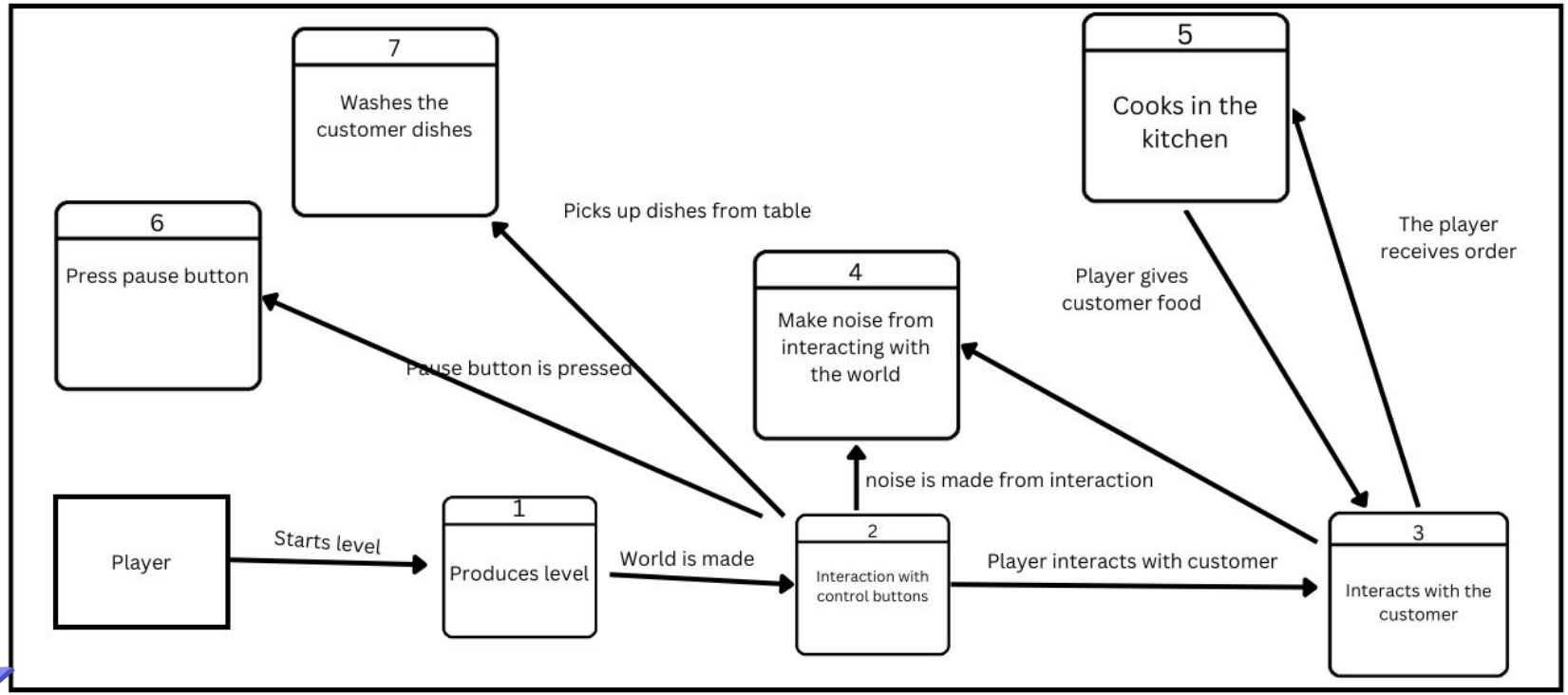


Context Diagram



MAKI STUDIOS

Diagram 0



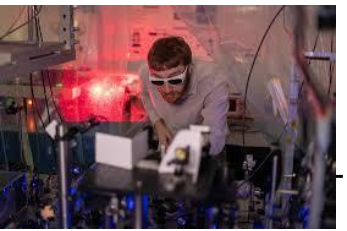
MAKI STUDIOS

Jennie Tafoya - Certified Plasma Level Designer

- I do level design.
  - This sets down the platforms, spaces out the components of gameplay, keeps track of difficulty, builds intriguing and awesome levels

- My component is both **high priority** and **not high priority**

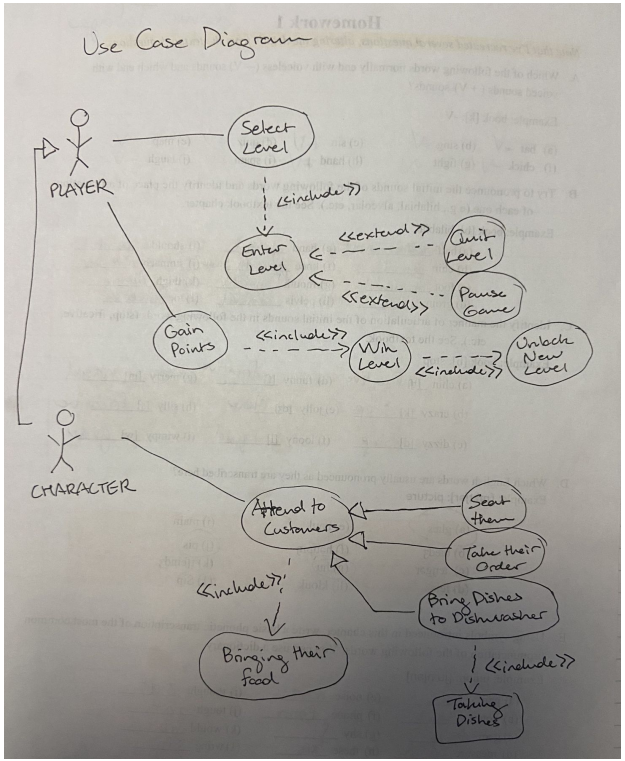
  - Here's what I mean:
    - Base-level design (the layout of the general game) determines the components that are implemented



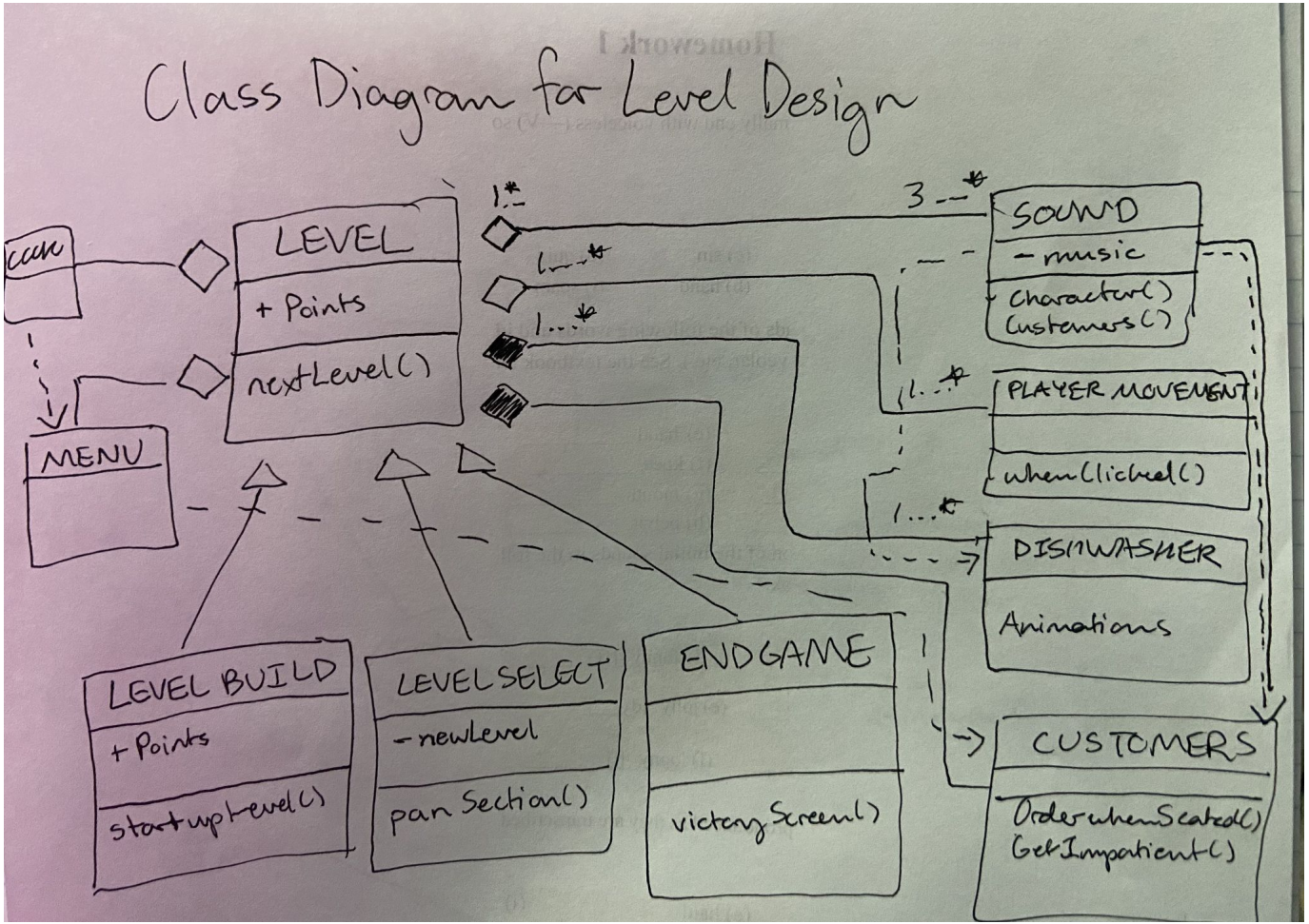
    - Placement/design of individual levels less of a priority. Can be completed once everything else is completed

- Not super complex

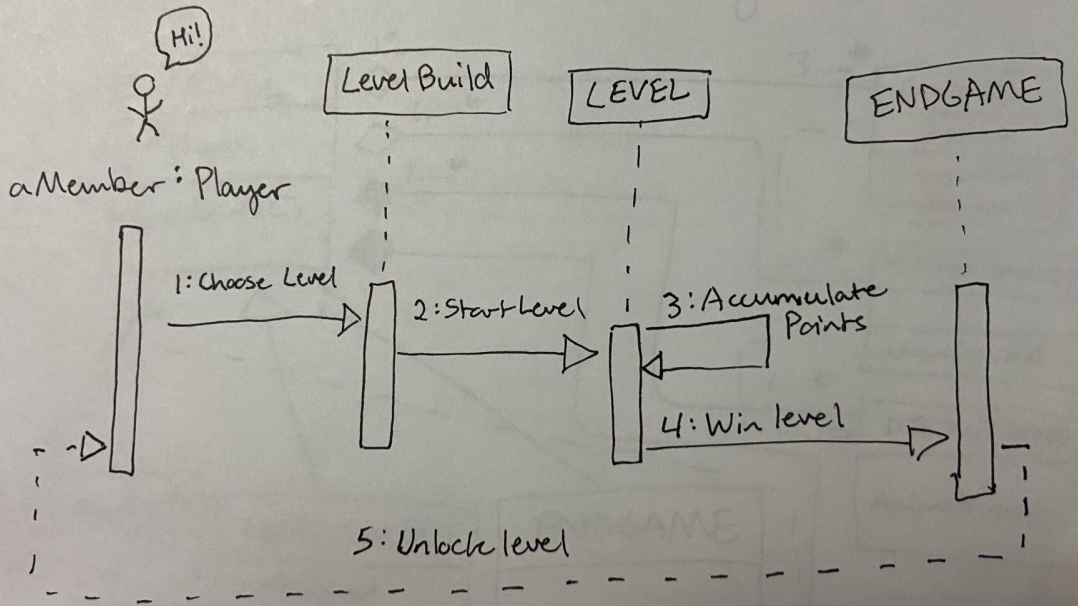
Any questions?



Class Diagram for Level Design



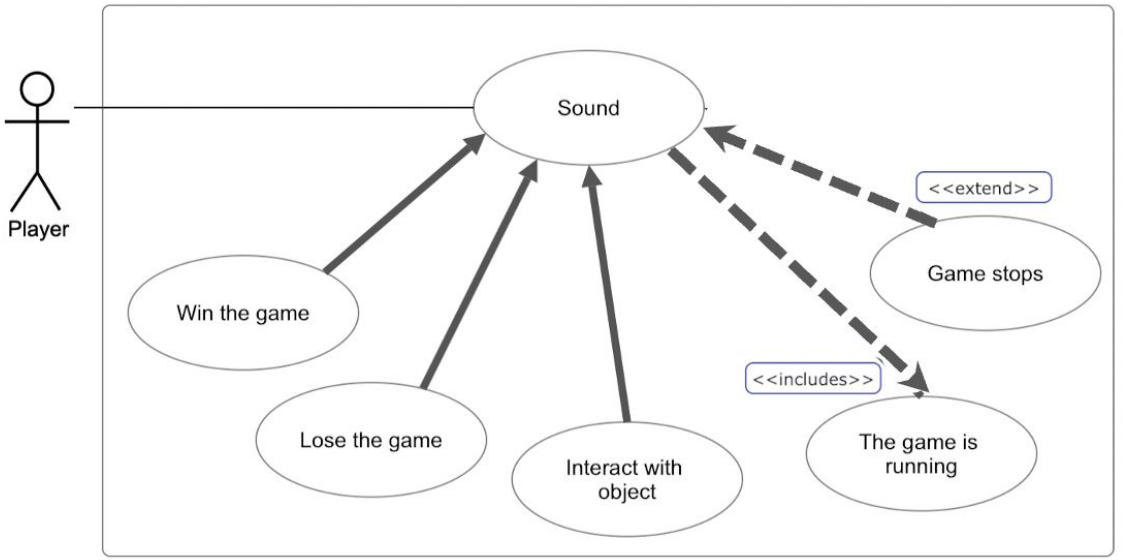
Sequence Diagram



Feature: Sound

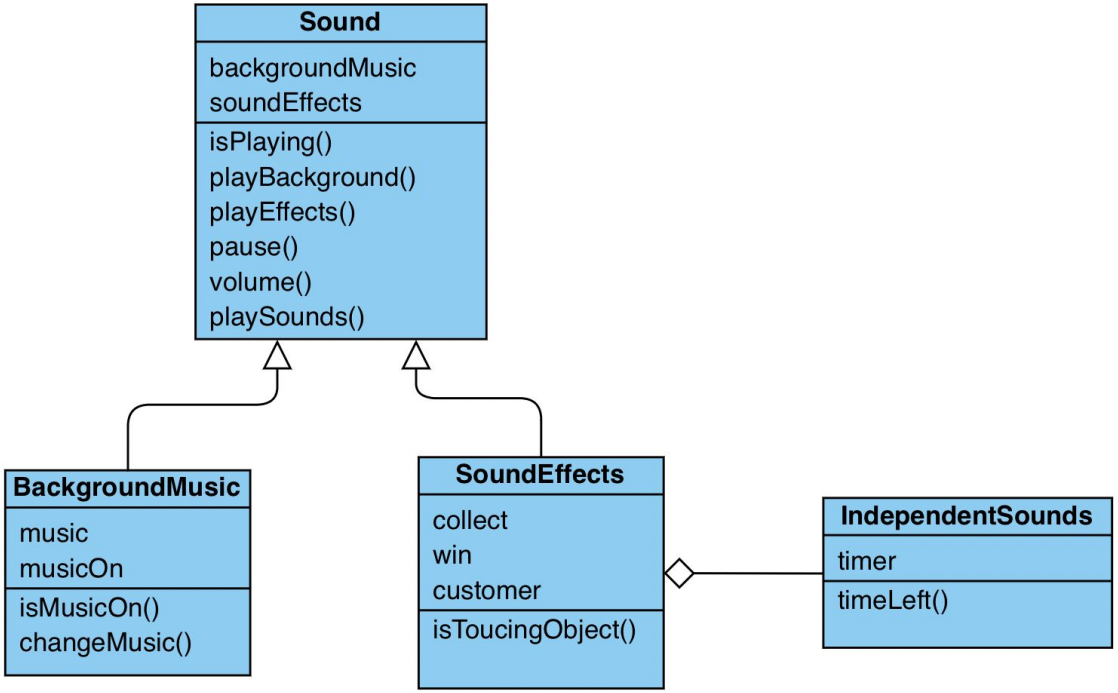
- Role:
  - Background Sound
  - Interacting with game wo
- Priority: 2* (Essential)
- Complexity: Average

Use Case Diagram



Khadeeja

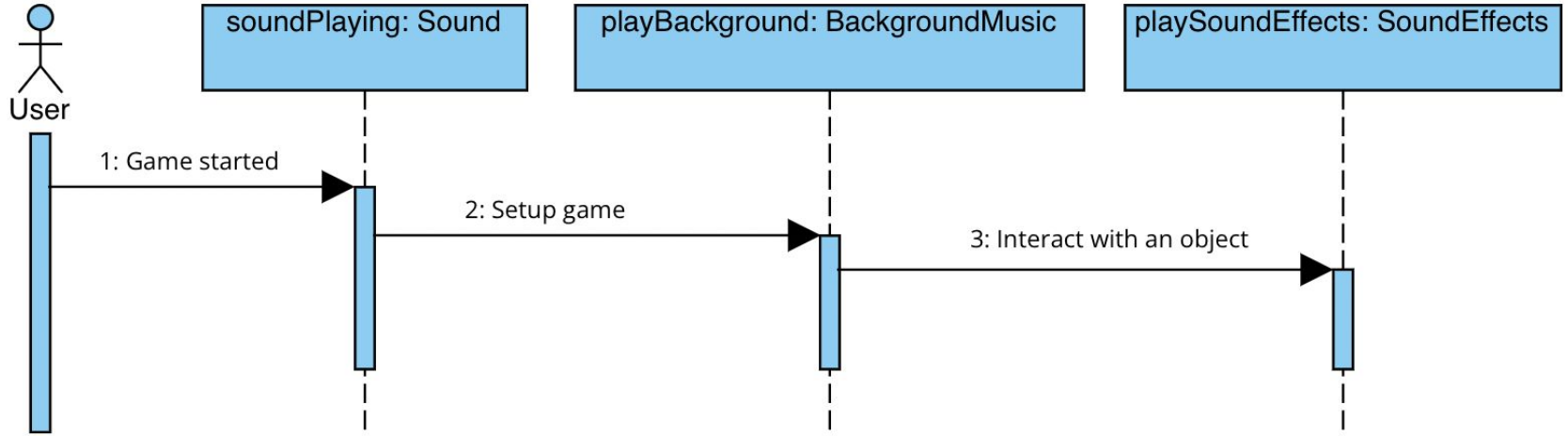
Class Diagram



MAKI STUDIOS

Khadeeja

Sequence Diagram



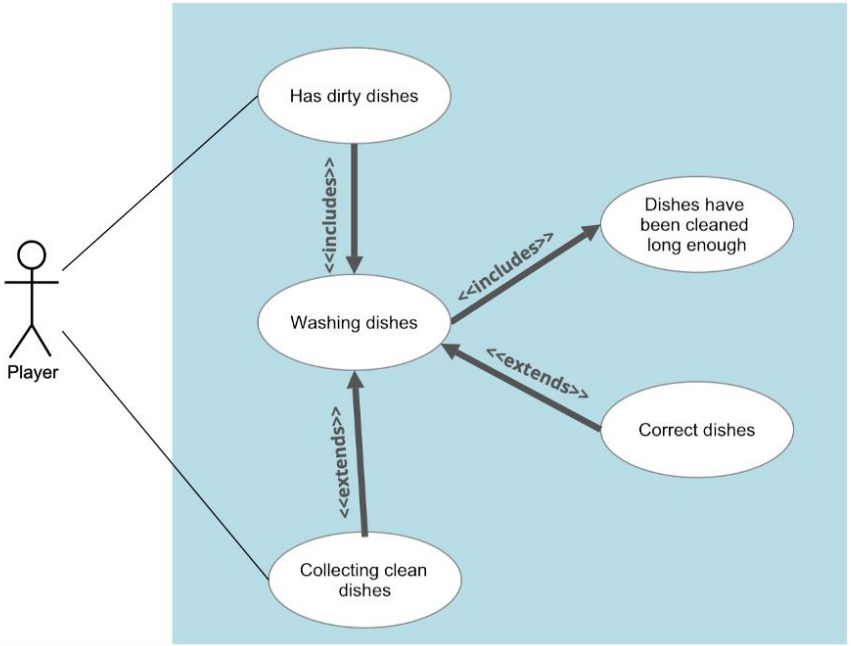
MAKI STUDIOS

Khadeeja

Feature: Dishwasher

- Role:
  - Create the dishwasher
  - Allow players to wash and reuse dishes
- Priority: 3* (nice to have)
- Complexity: Average possibly slightly more complex

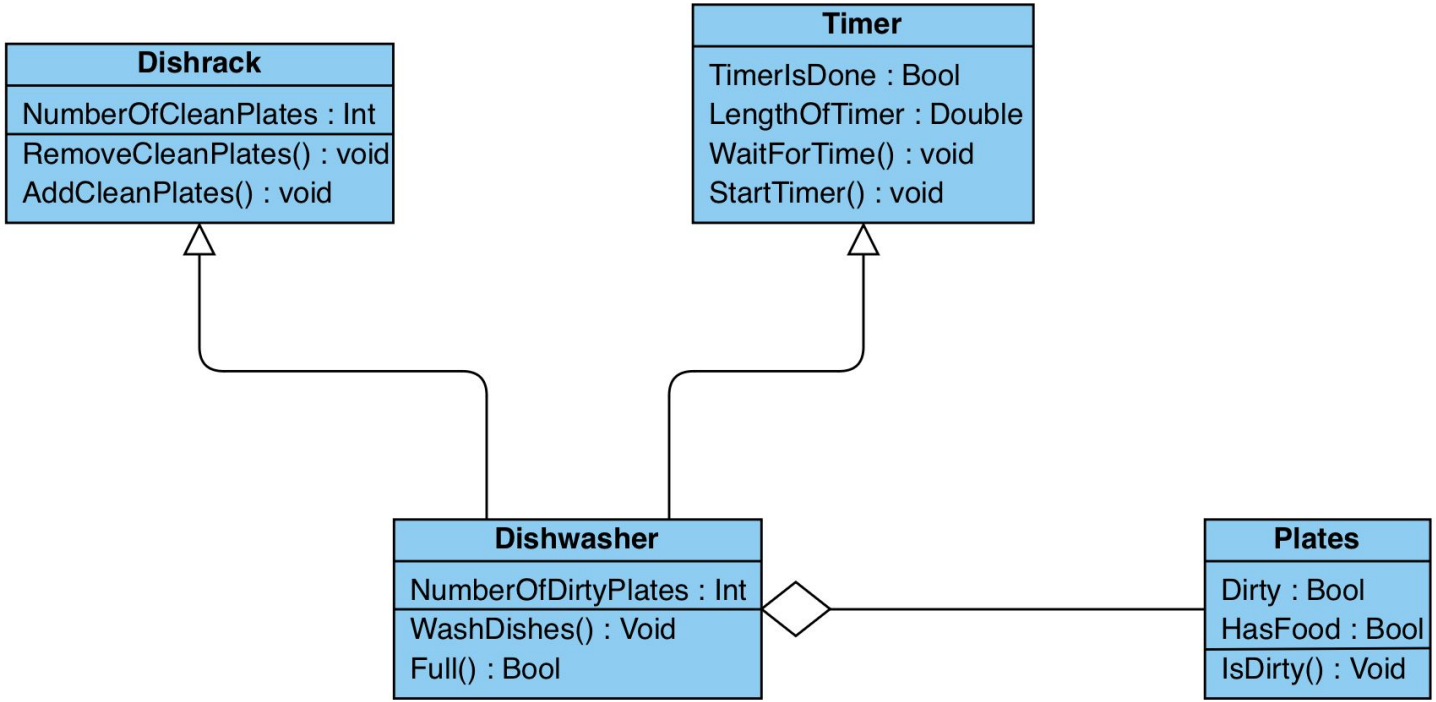
Use Case Diagram



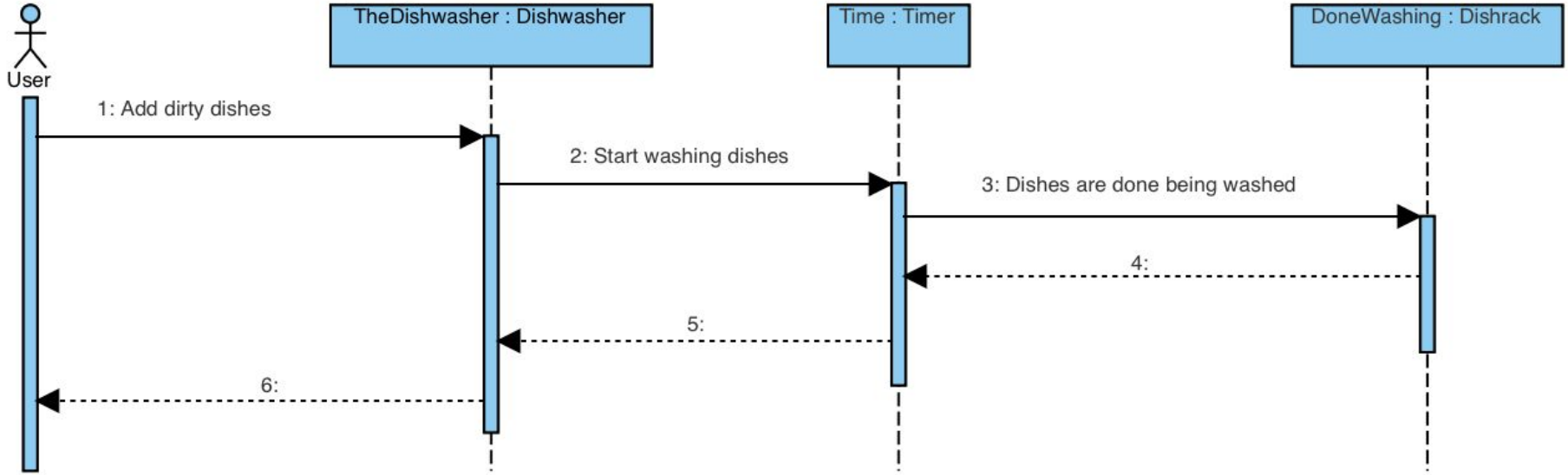
Ibrahim

MAKI STUDIOS

Class diagram



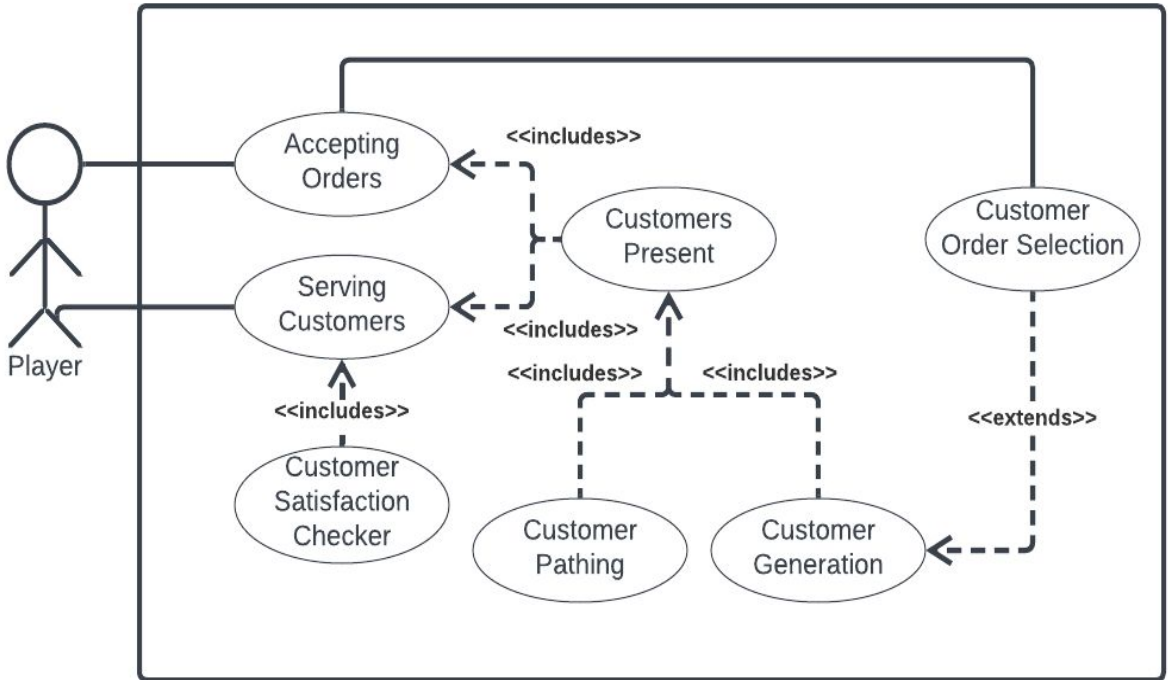
Sequence Diagram



Feature: Customers

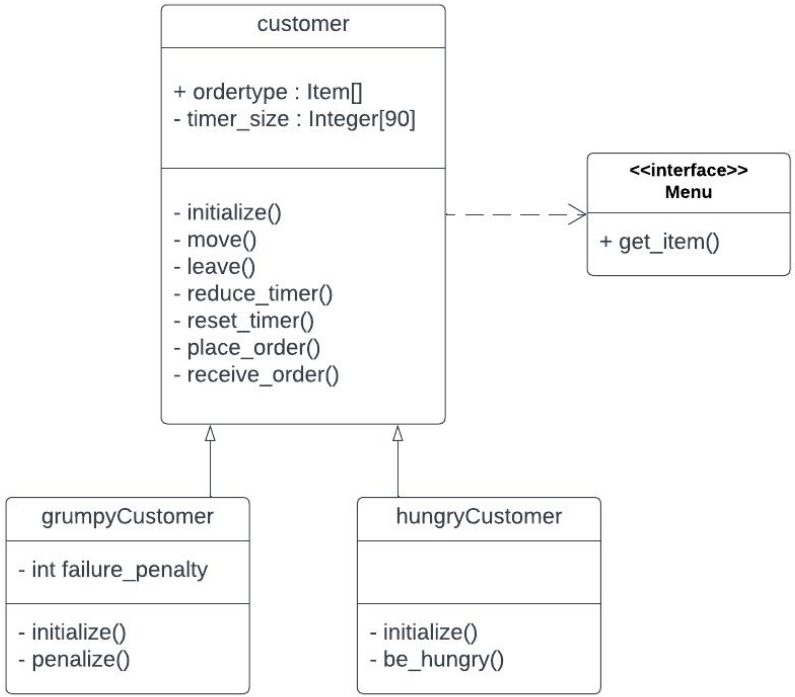
- Role:
  - Create / design the customers
  - Integrate the interactions involved
  - Taking Orders / Serving
  - Make game demo (TL 5)
- Priority: 1* (Must have)
- Complexity: Slightly more complex. Could end up being very difficult.

Use Case Diagram



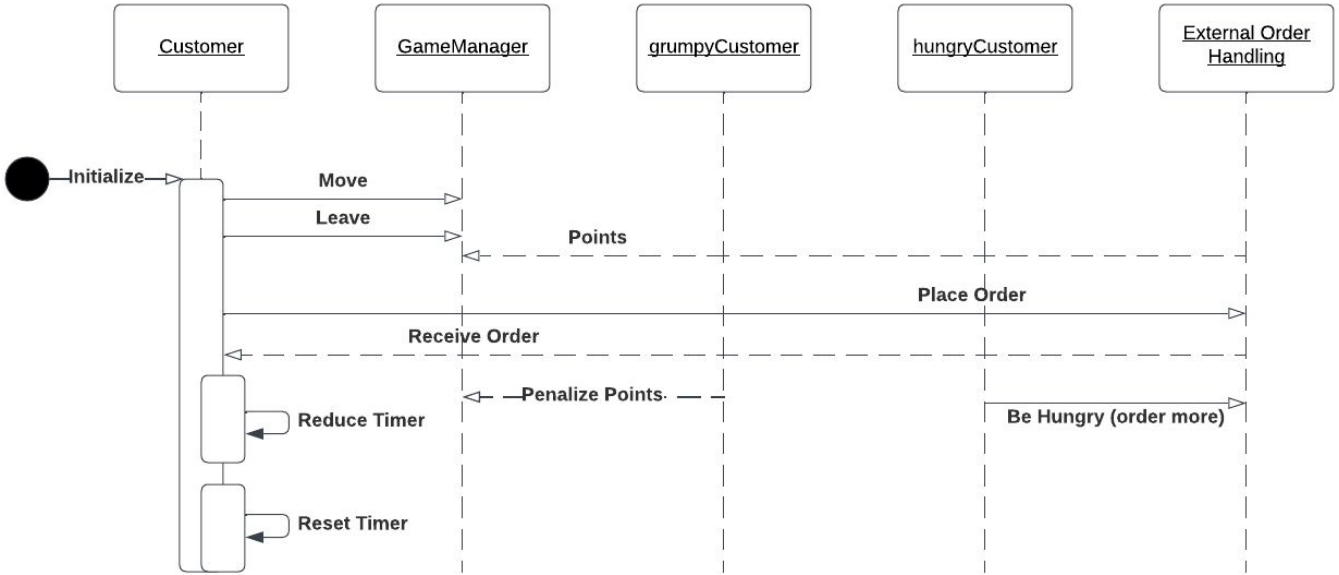
Alex

Class Diagram - Customers



Alex

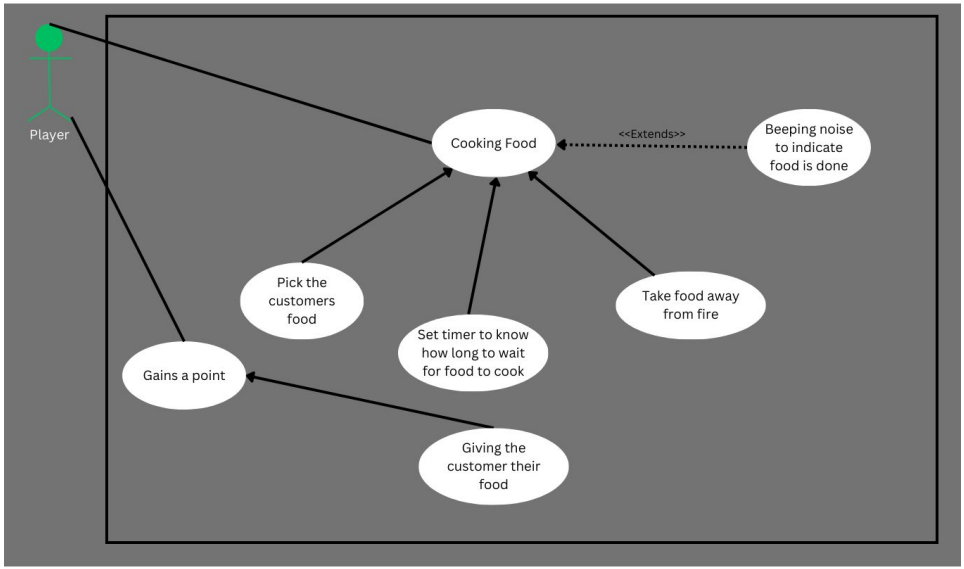
Sequence Diagram - Customers



Alex

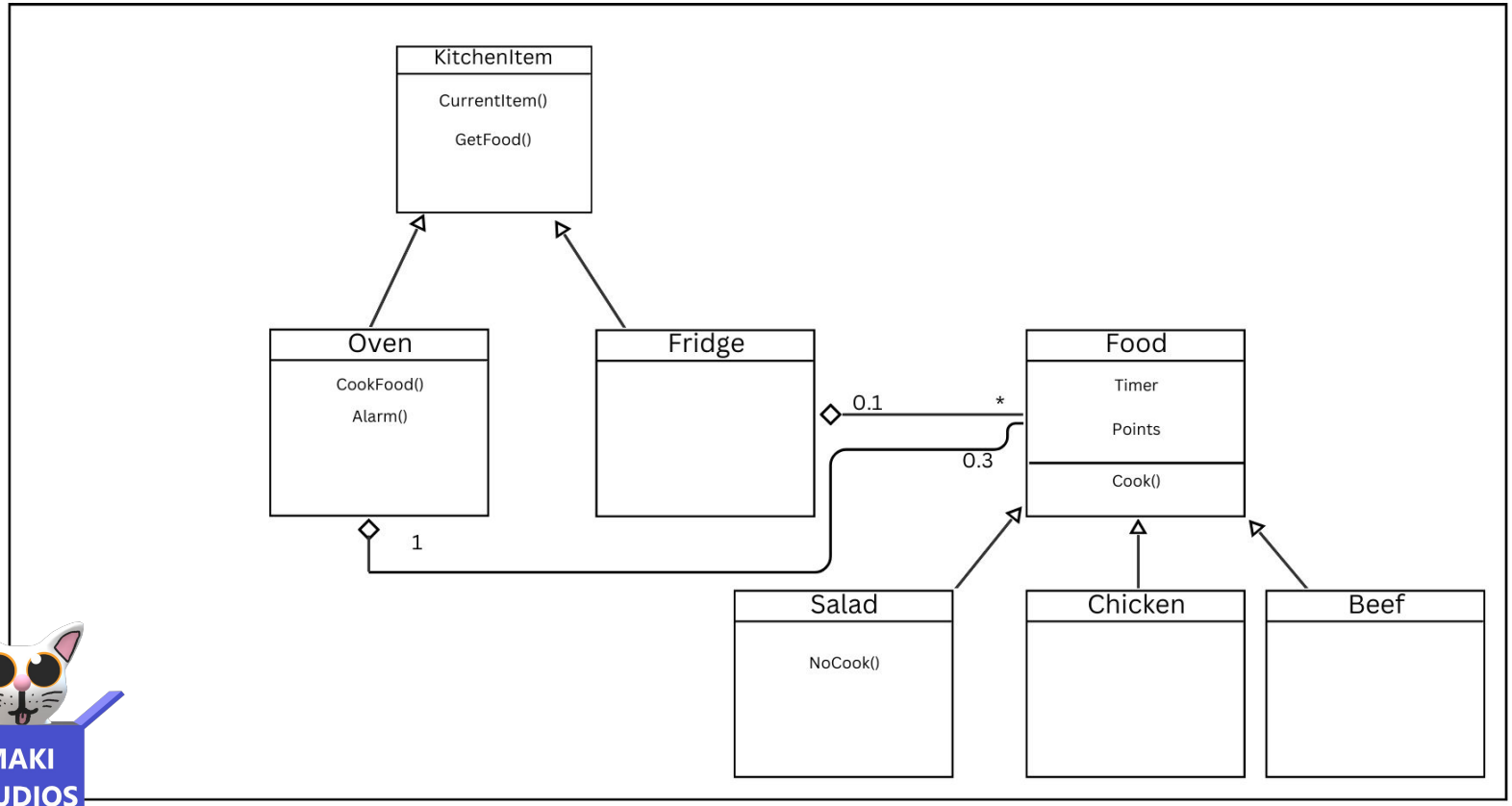
Feature Food/Points - Mauricio

- Playing mechanic- relatively important
- Somewhat complex because many different parts need to interact with each other.



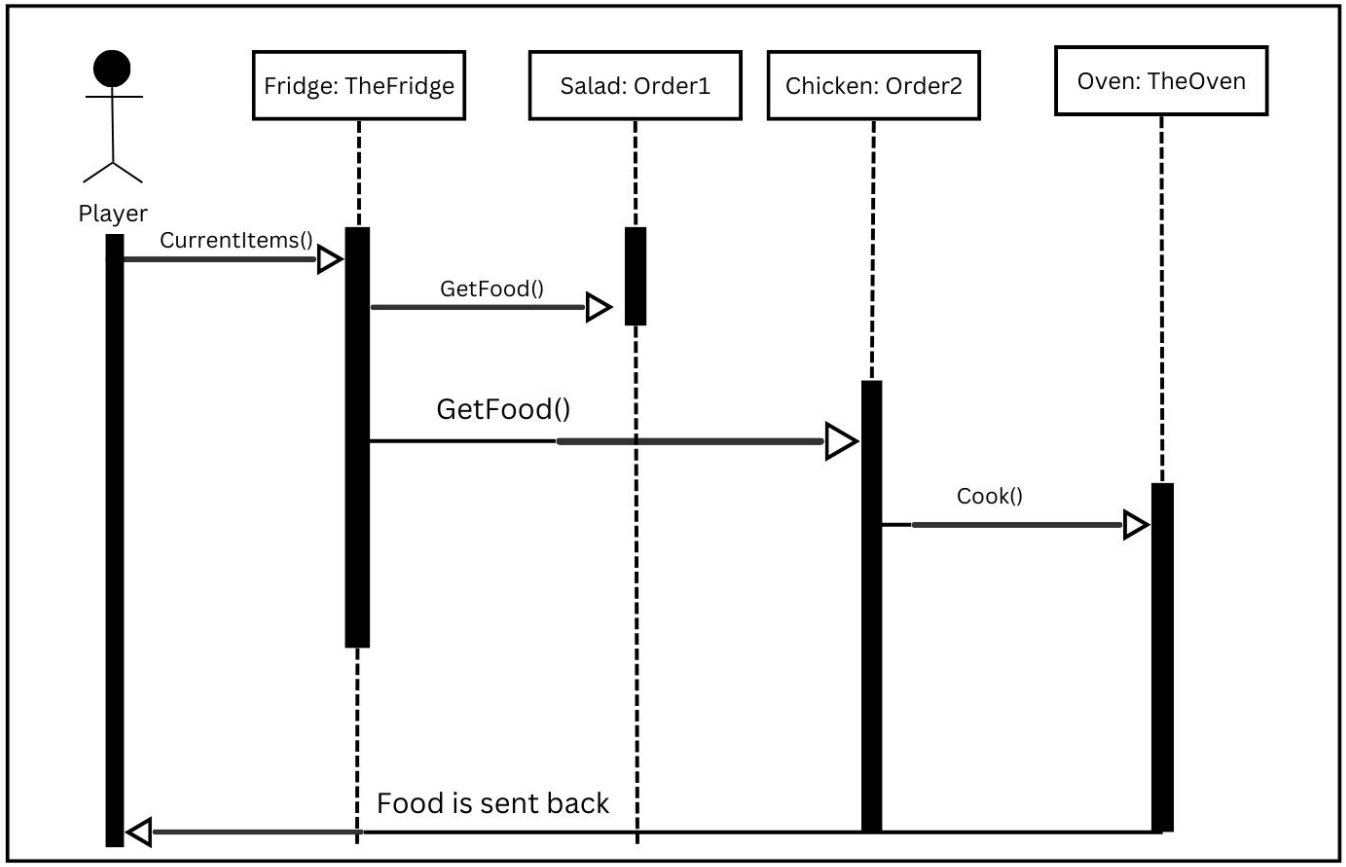
MAKI STUDIOS

Class Diagram



MAKI STUDIOS

Sequence Diagram



MAKI STUDIOS

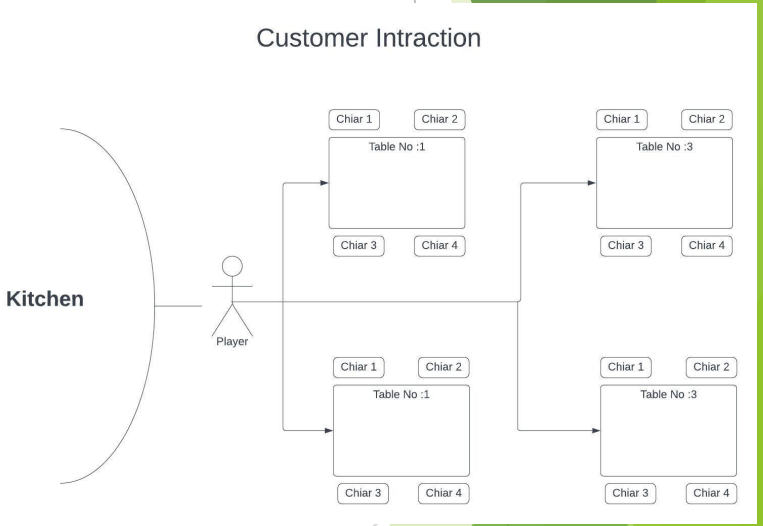
Naga Kadarla

ID: kada7630

Role: Player Movement

Role: Player Movement

- **Player Engagement:** Player engagement hinges on the ability to control your character's every move, from walking and running to attend to customer requests.
- **Gameplay Strategy:** Players must strategically navigate the restaurant to maximize their serving efficiency, minimize customer wait times, and optimize their earnings.
- **Challenges and Obstacles:** Monolith game often include various challenges and obstacles, such as Tables, Chairs, and tight space dining area. Player movement becomes a key component in overcoming these challenges and maintaining a sense of urgency and excitement.



Priority of Player Movement Feature

- **Central Gameplay Element:**

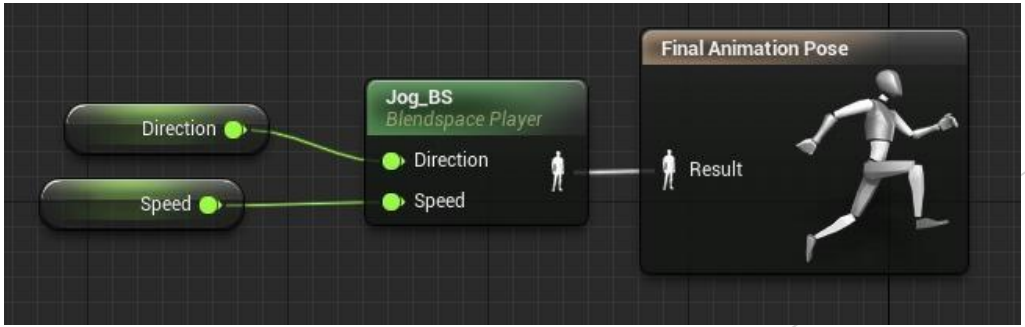
  The primary gameplay revolves around the player's ability to navigate the restaurant, serve customers, and manage tasks efficiently. Player Movement is at the heart of these actions, and it directly influences the player's ability to excel in the game.

- **Engagement and Immersion:**

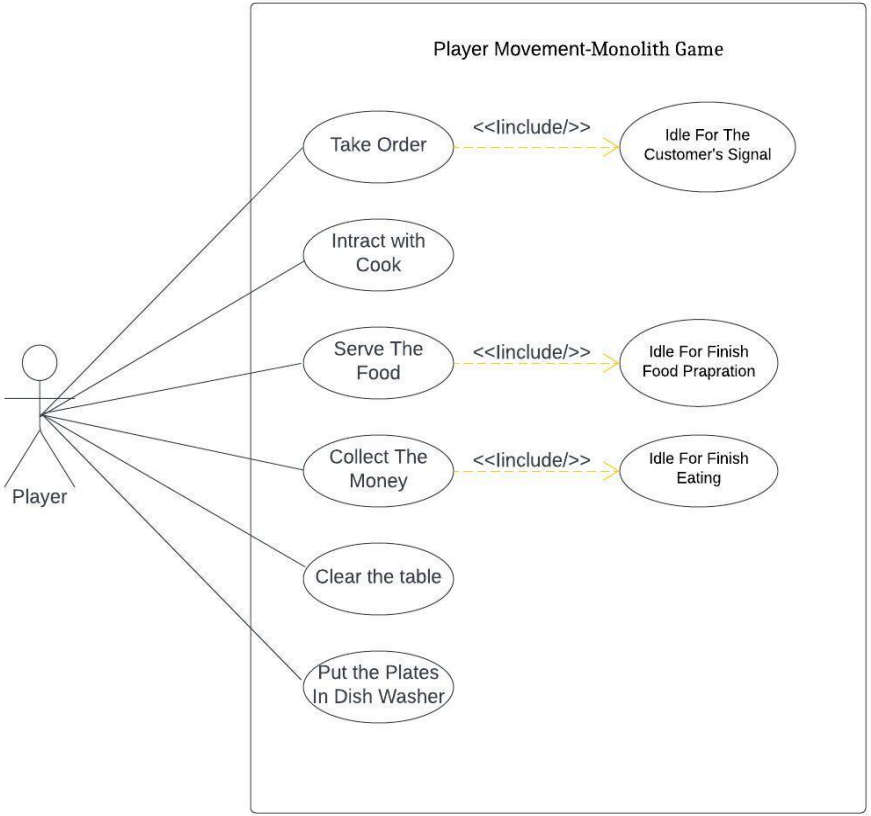
  Players need to feel connected to their in-game character and environment, and the quality of movement controls greatly impacts this connection.

- **Strategic Gameplay**

  Players need to make quick decisions about where to go, which customers to serve first, and how to optimize their actions. The quality of Player Movement controls can significantly affect a player's ability to execute their strategies effectively.



Use Case Diagram

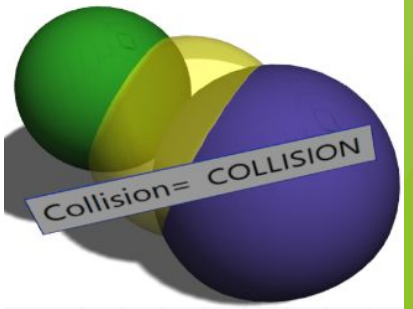


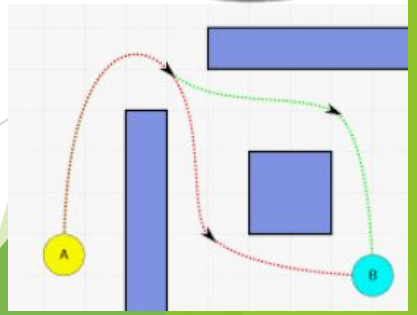
Walk Through in My Use Case Diagram

- **Player Moves Towards Customer:** This step involves the player character navigating through the restaurant to reach a customer.
- **Taking Customer's Orders:** Once the player reaches a customer, they need to interact with them to take their order.
- **Interact with Cook and Place the Kitchen Order List:** After taking orders, the player character needs to move to the kitchen area to submit the orders to the cook.
- **Serving Food to Appropriate Customer:** After the food is ready, the player must deliver it to the correct customer.
- **Collecting Payments from Customers:** To complete the dining experience, the player character needs to collect payments from customers.
- **Take All Plates and Clear Table:** After customers finish their meals, the player must clear tables efficiently.
- **Put the Plates to Wash:** Finally, the used plates need to be taken to the dishwashing area. Player movement ensures that the plates are transported efficiently for cleaning

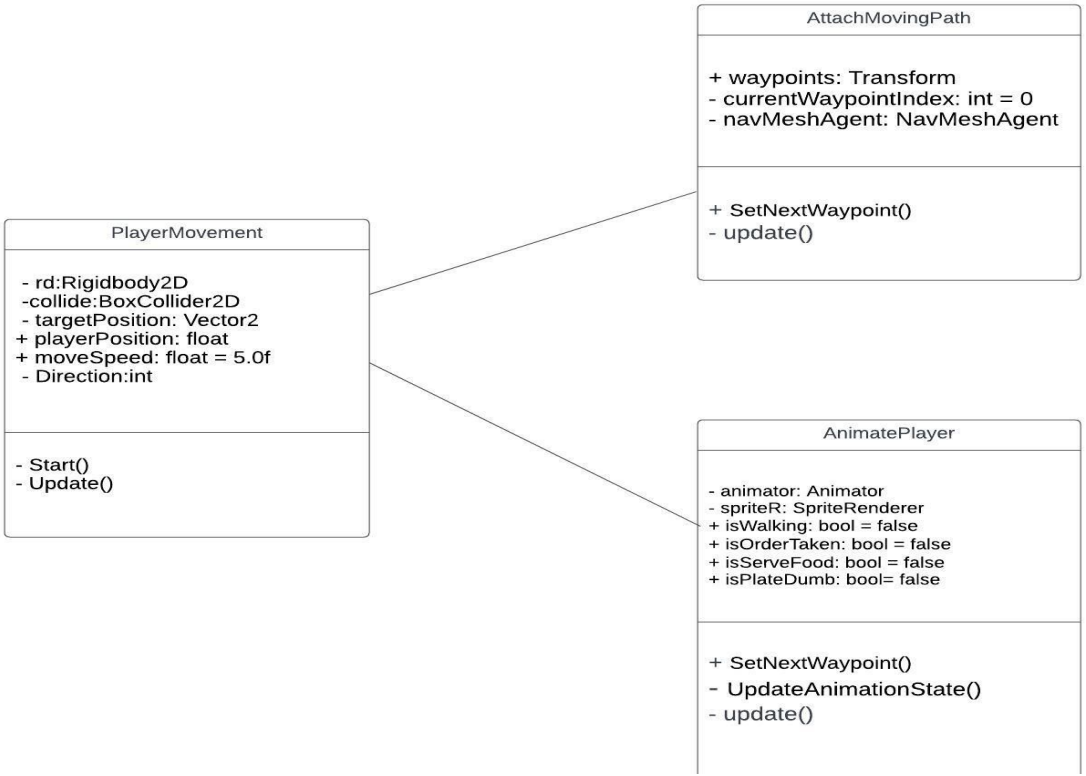
Complexity of the Work

- **Basic Movement:** Basic Movement typically involves implementing simple controls for character navigation, such as moving in all directions (left, right, up, down), possibly with variations in speed. The complexity might be relatively low.
- **Collision Detection:** Our game requires collision detection to prevent the player character from walking through tables, chairs or customers, this adds complexity. Implementing accurate collision detection can be moderately complex, especially in congested restaurant environments.
- **Pathfinding:** Our game includes pathfinding algorithms to navigate around obstacles or find optimal routes, this can significantly increase complexity. Implementing pathfinding algorithms can be complex and may require additional development time.
- **Animation:** If we want the player character's movement to be visually appealing with animations like idle, walking, carrying trays of food or dump the used plates for wash, this adds complexity to the work. Implementing animations can be moderately complex, especially if you need smooth transitions between different states.
- **User Interface Integration:** If the movement controls need to be integrated into a user-friendly interface with various input methods (keyboard, mouse, touch, game controller), this can increase the complexity.
- **Error Handling:** Dealing with edge cases and error handling, such as when the player character gets stuck or encounters unexpected obstacles, can increase complexity as it requires robust error-handling mechanisms.

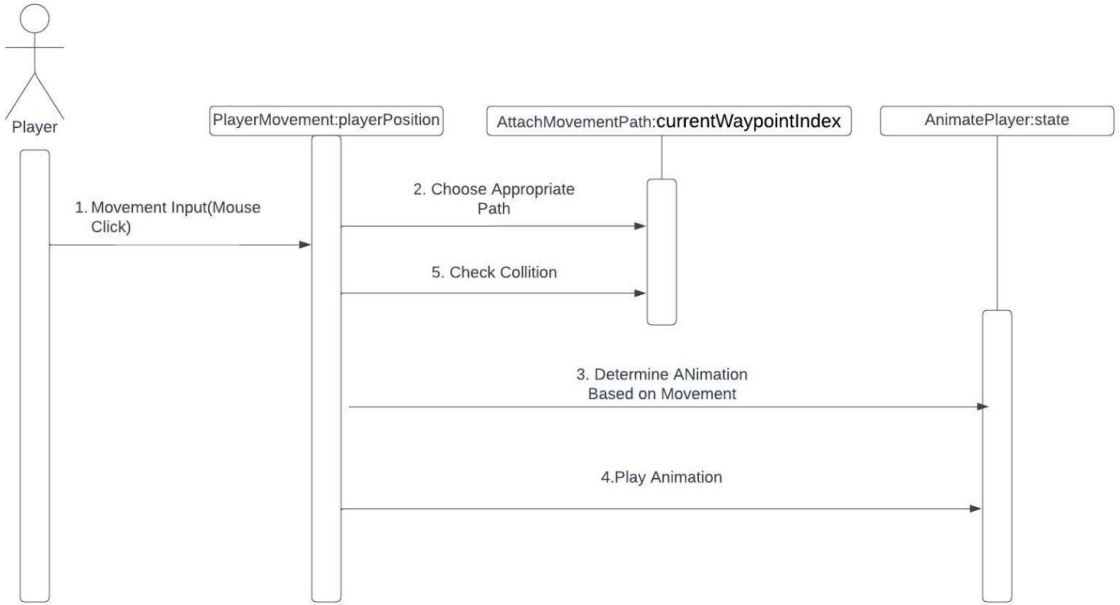




Class Diagram



Sequence Diagram



Monolith

Bibek K Sharma

Role:-Menu/Help/UI

Menu/Help/UI

Player Engagement:

- **Intuitive UI:** User-friendly and visually appealing interface to enhance navigation and gameplay.
- **Interactive Tutorials:** Step-by-step guidance integrated within the game to foster player understanding and engagement.
- **Reward System:** Bonus and reward structures that encourage players to explore menu options and accomplish tasks.

Game Strategy:

- **Adaptive Difficulty Levels:** Gradually increasing difficulty levels to challenge players and encourage strategy development.
- **Varied Menu Options:** Diverse and complex menu options necessitating strategic planning and quick decision-making for optimal service.

Challenges and Obstacles:

- **Time Management:** Players must juggle multiple tasks on time, simulating a real-time dining experience.
- **Resource Allocation:** Strategic allocation of resources like ingredients and kitchen appliances for efficient order fulfilment.
- **Customer Satisfaction:** Continual strategising to maintain high customer satisfaction levels through efficient service management.

Technical Challenges:

1. Designing an Intuitive UI:

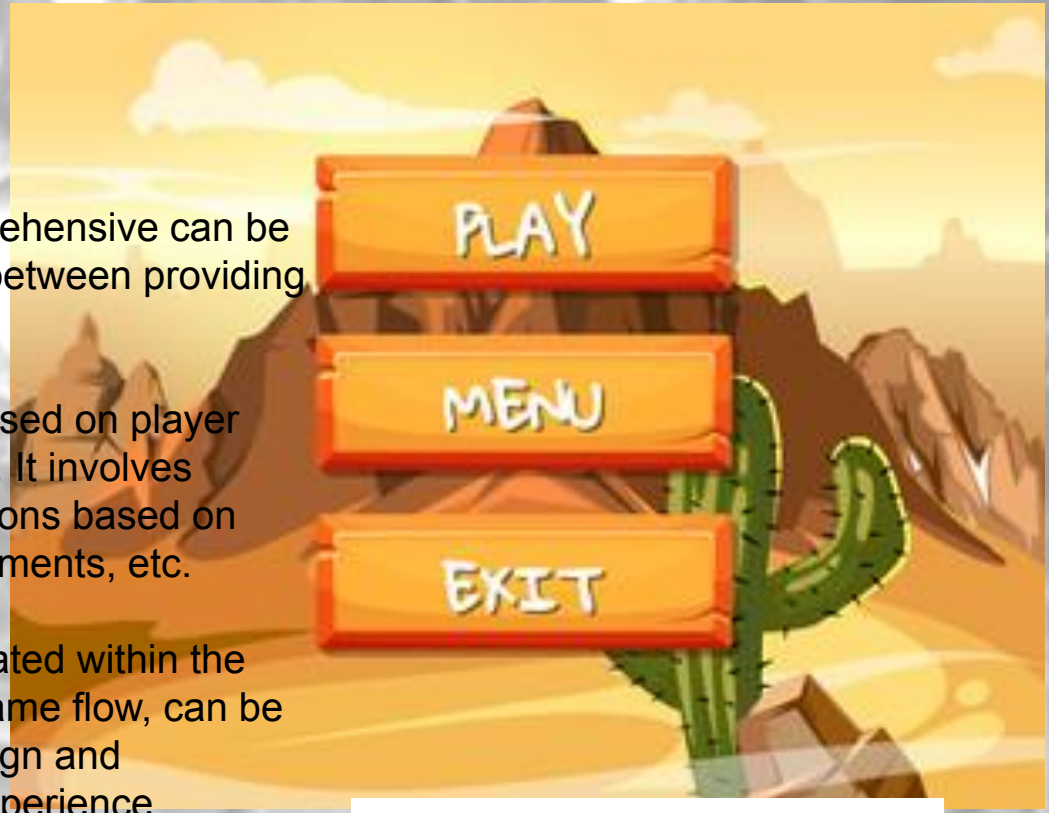
Creating a UI that is both intuitive and comprehensive can be a challenge. It must strike the right balance between providing necessary information and avoiding clutter.

2. Creating Dynamic Menus:

Designing dynamic menus that can adapt based on player progression could be technically challenging. It involves creating algorithms that adjust the menu options based on various parameters like player level, achievements, etc.

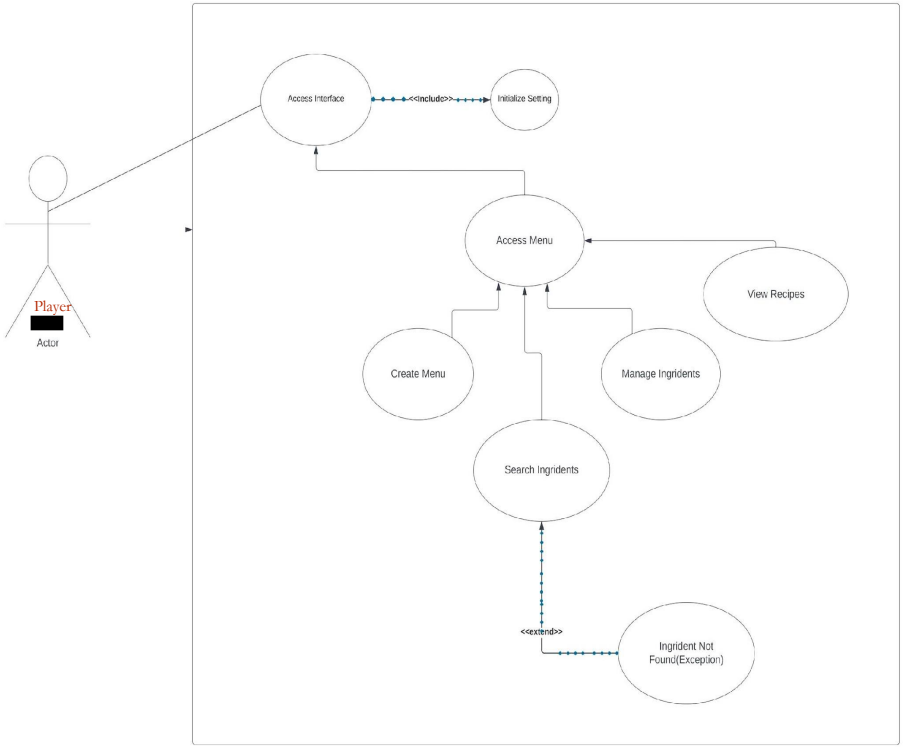
3. Integrating Help Section:

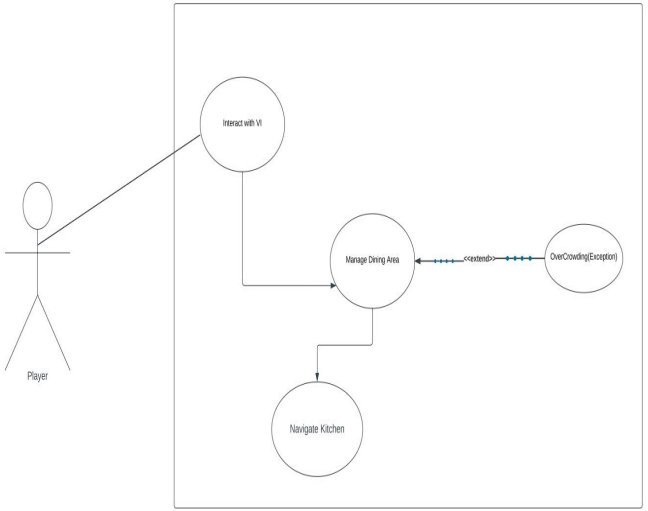
Developing a help section seamlessly integrated within the game, providing help without breaking the game flow, can be a technical challenge. It requires careful design and integration to avoid disrupting the player's experience.



MAKI STUDIOS

USE CASE DIAGRAM





Walk Through Use Case Diagram

Use Case 1: Menu Selection and Help

Summary: This feature allows players to navigate through different menu options and access the help section for guidance.

Actors: Player, Game System

Preconditions: The player has initiated the game and is at the main menu.

Basic Sequence:

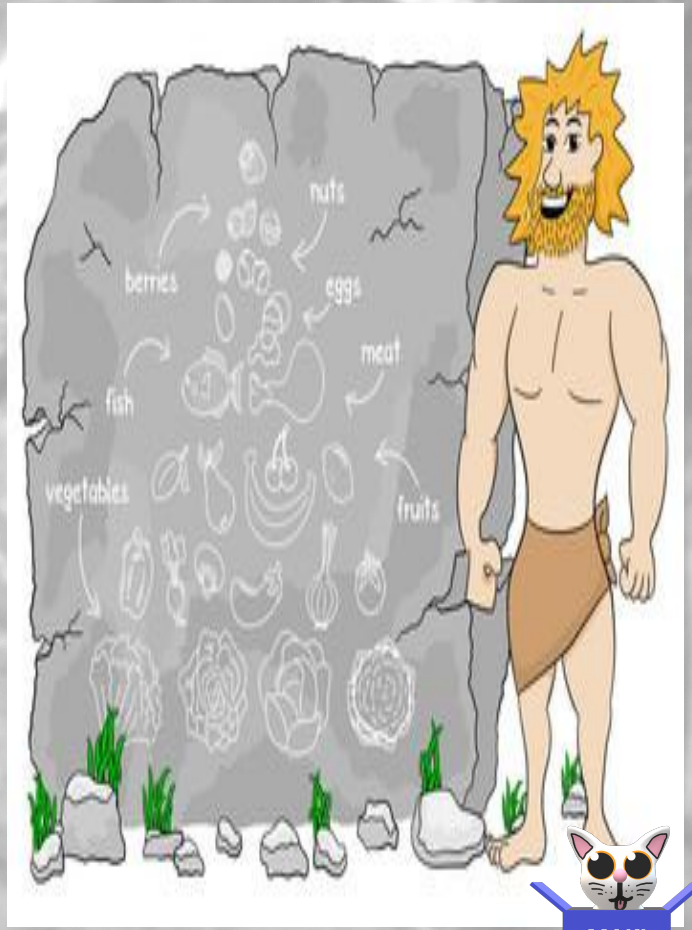
1. **Step 1:** The player views the various menu options displayed on the screen.
2. **Step 2:** The player selects a preferred menu option to explore (e.g., Start Game, Settings, Help).
3. **Step 3:** The game system responds by navigating to the selected menu option.
4. **Step 4:** If the help menu is selected, the game system displays a series of guides or tutorials.
5. **Step 5:** The player interacts with the chosen menu or help guide and may navigate back or make further selections.

Exceptions:

- **Step 3:** In case of an incorrect selection or system error, the game system displays an error message and prompts the player to make a selection again.
- **Step 4:** If the help menu is unresponsive or fails to load the content, the game system attempts to reload the content or shows an error message.

Post Conditions: The player successfully navigates through the menu and accesses the required sections without issues.

Priority: 1 (Must-have feature)



MAKI STUDIOS

Use Case 2: User Interface (UI) Interaction

Summary: This feature entails the graphical and interactive elements that players encounter while navigating the game, enhancing the user experience and gameplay.

Actors: Player, Game System

Preconditions: The player is actively engaged with the game, interacting with various UI elements.

Basic Sequence:

1. **Step 1:** The player engages with the game's visual interface, including navigating through different scenes or menus.
2. **Step 2:** The game system displays dynamic content based on the player's progression and choices.
3. **Step 3:** The player interacts with different UI elements, such as buttons, sliders, etc., to make choices or take actions in the game.
4. **Step 4:** The game system responds dynamically to the player's interactions, updating the game state accordingly.

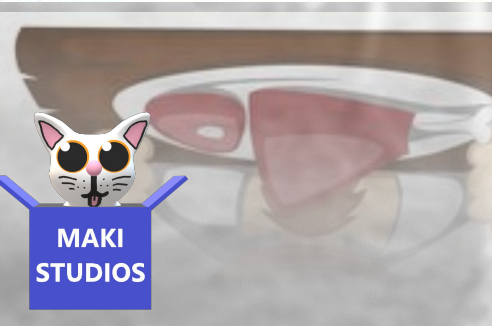
Exceptions:

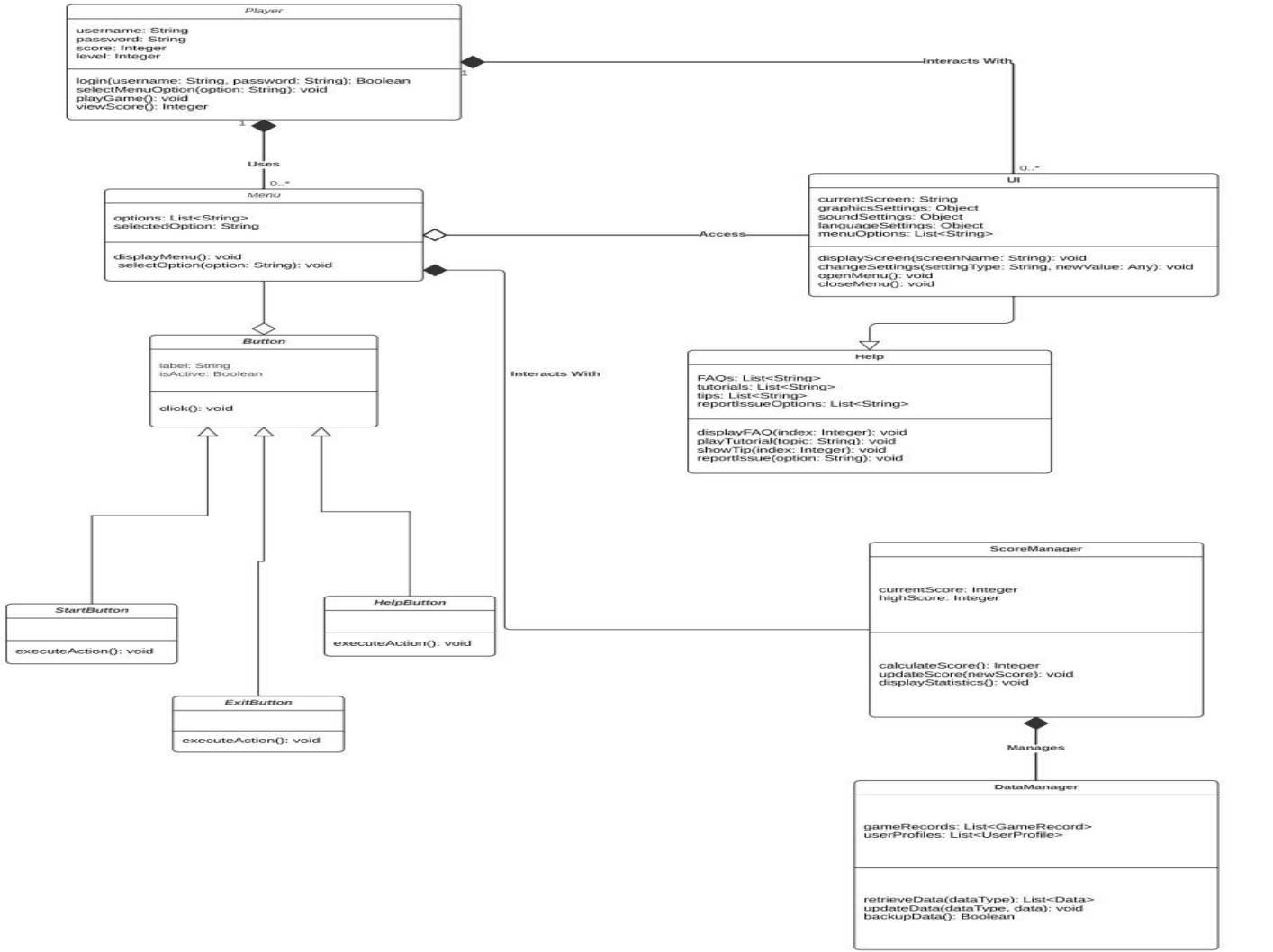
- **Step 3:** If the player encounters a glitch or bug, the game system tries to rectify the issue or prompts the player to restart the game.
- **Step 4:** If the game system fails to update the game state correctly, it may log the issue for further analysis and notify the player.

Post Conditions: The player has a seamless and engaging experience interacting with the visual interface.

Priority: 2 (Essential feature)

Class Diagram





MAKI STUDIOS

Walk Through Class Diagram

.Player Class(superclass)

1. Manages user info & and gameplay interactions.
2. Methods: Logging in, selecting menu options, initiating gameplay, viewing scores.

2. Menu Class(superclass)

1."The Menu class controls the display and selection of various options available to the player, offering a dynamic interface.“

3. Button Class(subclass)

1. "Functioning as a superclass within the Menu class, the Button class introduces primary attributes and methods while being specialised further into StartButton, HelpButton, and ExitButton, each having specific actions.

4. ScoreManager Class

1. Manages gameplay scores & statistics.
2. Methods: Calculating & updating scores, displaying statistics.

5. DataManager Class

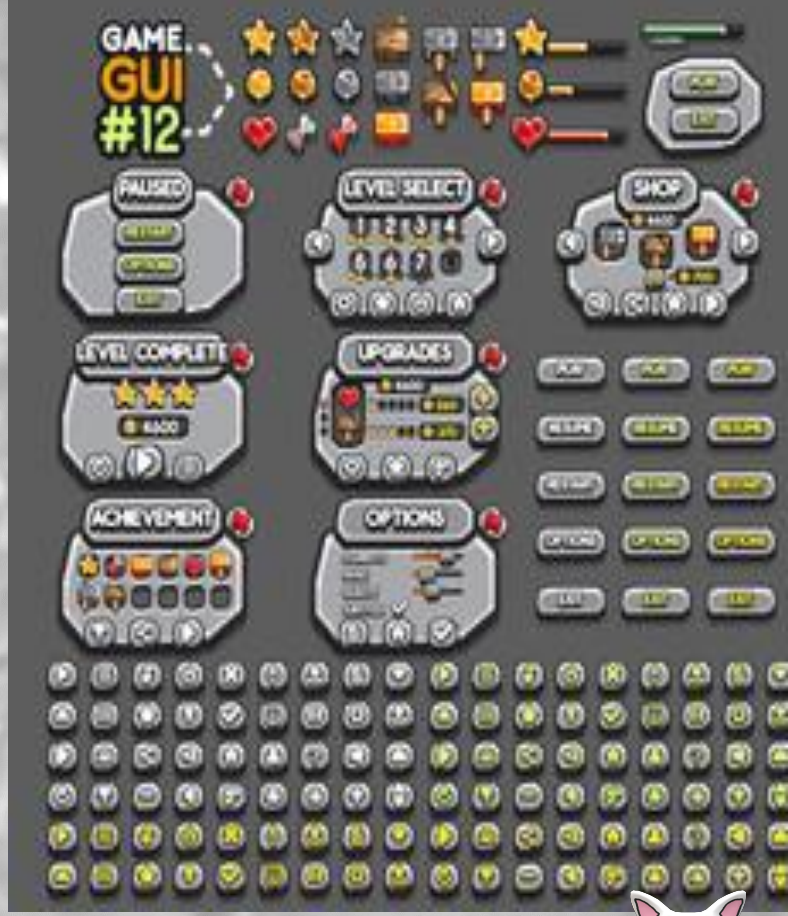
3. Manages data retrieval, update, and backup.
4. Methods: Retrieving & updating data, backing up data.

6. UI Class

5. Manages user interface elements & interactions.
6. Methods: DisplayGameScreen, UpdateUserSettings

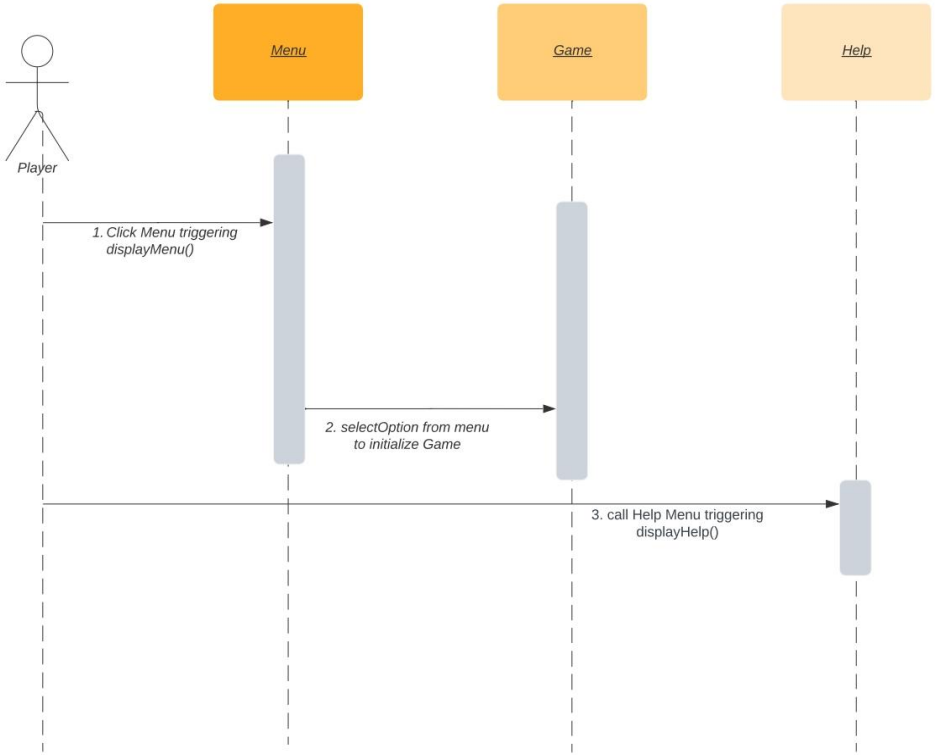
7. Help Class

7. Provides help & guidance to players.
8. Methods: DisplayHelpContent, SearchHelpTopics



MAKI STUDIOS

Sequence Diagram




Thank
you!