Gem Maker Tycoon: A 3D low-poly idle video game for Android devices

Víctor Izquierdo Alegre

Final Degree Work
Bachelor's Degree in
Video Game Design and Development
Universitat Jaume I

May 21, 2023

Supervised by: Raúl Montoliu Colás.

Acknowledgments

First of all, I would like to thank my Final Degree Work supervisor, Raúl Montoliu Colás, for guiding and supervising the creation of this document.

I also would like to thank Sergio Barrachina Mir and José Vte. Martí Avilés for their inspiring LaTeX template for writing the Final Degree Work report, which I have used as a starting point in writing this report.

Abstract

The purpose of this document is to capture the efforts involved in the development of Gem Maker Tycoon, an idle genre video game designed for Android devices. It also serves as a final degree project report that reflects the knowledge gained throughout the degree and personal projects over the years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

The project consists of developing an idle video game for Android devices. In the idle genre, gameplay continues to progress even when the player is not actively interacting. Players initiate the game's progress, and it continues to run in the background, generating resources and rewards over time. This genre offers a relaxing and passive gaming experience, allowing players to witness their progress grow without requiring constant attention. Here there is a trailer of the game: https://youtu.be/wDcoHXXg3VE.

1.1 Work Motivation

The main motivation behind developing a video game in the idle genre is the opportunity to explore new aspects of game development that we haven't previously experienced. By venturing into uncharted territory, we can discover and learn about new game mechanics, design principles, and player engagement strategies. This presents a chance for personal and professional growth, expanding our skill set and knowledge base in game development.

Additionally, the idle genre has gained significant popularity in the mobile game industry. By adding an idle game to our portfolio, doors are opened to collaboration and employment opportunities with companies that specialize in this genre, such as

Codigames [3]. Inspiration is drawn from their successful games, specifically Idle Supermarket Tycoon — Shop [5] and Idle Burger Empire Tycoon—Game [4], which have captured the interest and enjoyment of players.

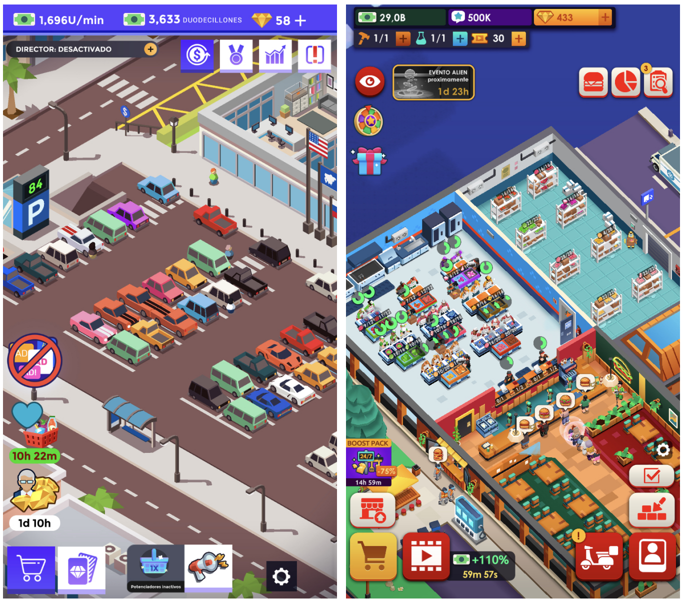


Figure 1.1: Games used as inspiration

Another important motivation is for the knowledge acquired during the studies to be applied and further developed. This project provides an opportunity to delve deeper into specific areas and enhance expertise. For example:

- **VJ1227 - GAME ENGINES:** Improve proficiency in Unity3D during the development of the idle game.
- **VJ1216 - 3D DESIGN:** Refine modeling skills to create visually appealing and engaging assets for the game.
- **VJ1215 - ALGORITHMS AND DATA STRUCTURES:** Put knowledge into practice by developing efficient algorithms and utilizing appropriate data structures to optimize game performance.

By undertaking this project, we strive to not only create an enjoyable idle game but also to push the boundaries of our abilities, gain industry recognition, and make practical use of the knowledge and skills acquired during our studies.

1.2 Objectives

The main objective was to develop a polished game and document it in this report to fulfill the requirements for obtaining the Video Game Design and Development bachelor's degree at University Jaume I. These objectives can be classified as follows:

- **Making an original video game using own models and animations:** The goal was to create a unique video game by designing and developing custom models and animations, showcasing our creativity and artistic skills in the process.

- **Implementing mechanics similar to other idle video games:** The project aimed to incorporate mechanics commonly found in idle video games, taking inspiration from successful titles in the genre. By implementing familiar mechanics, we sought to provide an engaging and enjoyable gameplay experience for players.

- **Having a video game that provides differential value to our portfolio:** Developing a high-quality idle video game would enhance our portfolio and differentiate it from others. By creating a standout game, we aimed to demonstrate our proficiency in game development and increase our chances of securing future opportunities in the industry.

- **Developing a video game with the knowledge obtained throughout the degree:** The project aimed to apply and expand upon the knowledge and skills acquired during this degree.

- **Realizing a memory that properly explains the project:** In addition to the game development itself, the objective was to create a comprehensive and well-documented report that effectively explains the project's concept, design, development process, and overall implementation. This report would serve as a testament to our ability to communicate and present our work professionally.

By accomplishing these objectives, we aimed to create an original and well-crafted video game, demonstrate our knowledge and skills, and provide a valuable addition to our portfolio while fulfilling the requirements for obtaining the Video Game Design and Development bachelor's degree.

In order to achieve these objectives, two assets have been used from the Unity Asset Store [17]: DOTween [7], a Unity tween engine, and Safe Area Helper [13], a script that adapts the UI for notched devices. Additionally, a music and sound effects bundle from Dark Fantasy Studio [14] has been incorporated into the game. Furthermore, the C# implementation of BreakInfinity [1] has been utilized to handle extremely large numbers within the game's mechanics.

The inclusion of these assets allowed us to leverage existing tools, scripts, and resources to enhance the development process and improve the overall quality of the game. By utilizing these assets, we were able to save time, optimize functionality, and enhance the audiovisual experience within the game.

1.3 Environment and Initial State

The project was undertaken individually during an internship, with no team involved in the development process. The work environment consisted of a personal workspace equipped with the necessary software and hardware tools for game development.

In terms of the initial state, the project utilized specific assets obtained from the Unity Asset Store to enhance the development process. The assets included DOTween [7], a Unity tween engine, and Safe Area Helper [13], a script designed to adapt the game's user interface for notched devices. Additionally, a music and sound effects bundle from Dark Fantasy Studio [14] was incorporated into the game to enrich the audiovisual experience. The C# implementation of BreakInfinity [1] was also employed to handle extremely large numbers within the game's mechanics.

By working individually during the internship and utilizing these assets, the project was able to leverage existing resources and frameworks to create a polished and engaging game.

1.4 Preview of the results

Presented here is a preliminary glimpse of the noteworthy results achieved in this project:

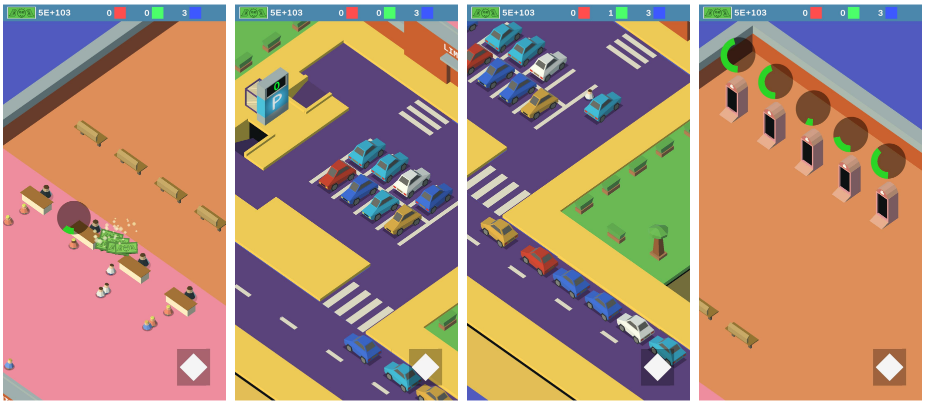


Figure 1.2: A preview of the results

In summary, the presented preview of the results provides a concise yet enticing overview of Gem Maker Tycoon. Through captivating visuals and intriguing glimpses into the gameplay mechanics, it offers a glimpse of the immersive and enjoyable experience that awaits players. This preview serves as a tantalizing introduction to the in-depth exploration of Gem Maker Tycoon in the subsequent chapters. With its promising features and engaging gameplay, Gem Maker Tycoon holds great potential to captivate and entertain players.

Chapter 2

Planning and resources evaluation

Contents

This chapter focuses on the technical aspects of the work, providing an explanation of the followed planning as well as the software and resources utilized.

2.1 Planning

In this section, an overview of the time allocation for the general tasks of the project is provided, as presented in Table 2.1. For a more detailed breakdown of each programming and art subtask, Table 2.2 and Table 2.3 respectively can be referred to. Additionally, the key milestones of the project can be found in Table 2.4.

2.2 Resource Evaluation

In this section, a comprehensive evaluation of the theoretical economic cost and resources required for the development of the project is conducted, as per the established guidelines. This evaluation takes into account various aspects to provide a thorough understanding of the project's financial and resource requirements.

Assuming that a single person undertakes the development of this project, the economic cost can be kept at zero euros. However, it is essential to consider that investing in certain assets can significantly enhance the project's quality and expedite its completion. Acquiring high-quality pre-made assets, such as 3D models, sound effects, or music compositions, can save substantial development time and effort. Consequently, the economic cost could increase when purchasing such assets from reliable sources.

Moving on to hardware resources, a decent computer system is necessary to facilitate a smooth and efficient development process. A capable computer with sufficient processing power, RAM, and storage capacity allows the developer to work fluently and handle the project's demands effectively. Considering the cost of a reliable computer system for game development purposes, the estimated investment would be around 800 euros. This cost accounts for acquiring a computer system that meets the recommended specifications for game development, ensuring optimal performance and productivity.

When considering the software resources, an additional cost of 25 euros would be incurred due to the acquisition of the music and sound effects bundle from Dark Fantasy Studio [14]. Nevertheless, there are several powerful tools and platforms available free of charge, that significantly reduce the overall cost of the project.

- **Unity (Free) [18]:** A comprehensive 2D and 3D game engine that provides a wide range of functionalities, including physics simulations, scripting support, and a robust editor interface.

- **Visual Studio (Free) [12]:** An integrated development environment (IDE) that seamlessly integrates with Unity, offering powerful code editing and debugging capabilities.

- **GitHub (Free) [8]:** A free GIT repository hosting service that facilitates efficient collaboration and version control management.

- **Google Drive (Free) [9]:** A cloud storage platform for storing and accessing project-related documents, ensuring easy data backup and sharing.

- **Unity Asset Store (Free) [17]:** A marketplace abundant in assets tailored for Unity, offering a wide variety of 3D models, animations, sound effects, and more.

- **Blender (Free) [2]:** A versatile software used for creating and animating 3D models, providing advanced features for modeling, sculpting, rigging, and rendering.

- **Krita (Free) [10]:** A powerful software utilized for crafting textures for the models, offering a range of painting and texture creation tools.

Consequently, the total cost of the project would amount to **825 euros**, considering the additional expenses for hardware and software resources.

Table 2.1: General tasks

Task	Type	Hours
Modeling and animation	Art	77h
Music & Sound	Music	10h
Programming the game's concept in Unity3D	Programming	150h
Testing	Programming	30h
Technical proposal	Documents	7h
Game Design Document	Documents	25h
Memory	Documents	30h
Presentation	Documents	10h
Total hours		343h

Table 2.2: Programming subtasks breakdown

Subtask	Hours
Client & Worker behaviour	26h
Parking & Road behaviour	36h
CounterZone behaviour	19h
WorkZone behaviour	16h
RestZone behaviour	13h
Top row UI	10h
Upgradable parts UI	19h
Options UI	1h
Camera system	10h
Total programming hours	150h

Table 2.3: Artistic subtasks breakdown

Subtask	Hours
Workers assets	2h
Clients assets	10h
Parking assets	15h
CounterZone assets	13h
WorkZone assets	12h
RestZone assets	16h
Surroundings assets	4h
Limitless Gems assets	5h
Total artistic hours	77h

Table 2.4: Milestones

Milestone
Client & Worker behaviour
Zones behaviours
Assets modeled and animated
User interface

Chapter 3

Game Design Document

Contents

3.1 Introduction

Gem Maker Tycoon is an isometric 3D low-poly video game designed for Android devices. The player assumes the role of the boss of "Limitless Gems", a company that specializes in producing and selling gems to its clients.

3.2 Description

In Gem Maker Tycoon, the player takes on the role of guiding the company to success. The player is responsible for managing various aspects of "Limitless Gems" and ensuring client satisfaction.

3.3 Genre

The genre of Gem Maker Tycoon is idle, which means that the game's action progresses automatically without any input from the player [11]. As a result, "Limitless Gems" runs on its own.

3.4 Platform

The game is targeted for the Android platform, with the possibility of an iOS port depending on its success. However, the primary platform for the game will be Android.

3.5 References

- **Idle Burger Empire Tycoon Game** [4]: It is a game of the same genre that adds complexity to the typical idle game by requiring players to acquire materials to earn money, much like Gem Maker Tycoon.

- **SuperMarket Tycoon - Shop** [5]: It is also a similar game, but simpler. Nonetheless, its art style will serve as a significant source of inspiration for Gem Maker Tycoon.

3.6 Key Features

- **Age range:** Gem Maker Tycoon appeals to a wide age range, as it has features that cater to both younger and adult players. The low-poly and simplistic style of the game is appealing to younger players, while the challenge of managing resources and upgrading the company is attractive to adult players.

- **Difficulty:** The difficulty level of Gem Maker Tycoon will be appropriate for its intended age range, as it features straightforward player controls that only require tapping on the screen.

3.7 Core Gameplay

The core gameplay of Gem Maker Tycoon centers around the money-earning cycle typical of idle games. This cycle involves increasing the revenue generated by "Limitless Gems".

The cycle is comprised of six essential parts, namely the workers, clients, and different parts of the company, such as the parking lot, Rest Zone, Work Zone, and Counter Zone.

The player will need to make decisions on how to spend their money to upgrade the different parts, weighing options such as increasing revenue per sale or attracting more clients to "Limitless Gems", among other choices.

3.8 Game Flow

Gem Maker Tycoon's game flow revolves around a circular, money-earning cycle with the ultimate objective of boosting the revenue generated by "Limitless Gems". The main focus of the game is to constantly earn more money, upgrade the company, and play endlessly, which is a characteristic feature of the idle game genre.

3.9 Gameplay Elements

In 3.8 section, six essential parts of Gem Maker Tycoon's cycle were mentioned. In these next subsections, each of these parts will be explained in detail.

3.9.1 Clients

Clients arrive at the parking lot in their cars and enter "Limitless Gems" to place an order for gems. They wait for an indefinite period to receive their order, and once they receive it, the company earns revenue. After that, they return to their cars.

3.9.2 Workers

Workers also follow a route similar to that of clients. However, they create a gem and then go to the Rest Zone before returning to their table to create another gem. This cycle of gem creation and rest continues endlessly.

3.9.3 Work Zone

The Work Zone consists of up to 20 tables, each assigned to one worker. Additionally, it controls the amount of money received per gem, starting with 100 for red gems, 300 for green gems, and 600 for blue gems. The game begins with two tables, and players can unlock up to 100 levels per table. Each table has two attributes:

- **Money earned:** This refers to the amount of money earned per gem. Each level increases the money earned by 10%.

- **Production time:** This is the time it takes to create a gem. The base time is 12 seconds. Each level reduces the actual production time by 2%.

Moreover, there are three major upgrades available for purchase every five new tables, starting from major upgrade level 0. To perform a major upgrade, the player must ensure that each table has reached the maximum level for the current major upgrade, which are 25, 50 and 100. Once this condition is met, the player can execute a major upgrade that increases the earned money by 10. After the major upgrade the player can unlock more tables.

3.9.4 Rest Zone

Gem Maker Tycoon features a Rest Zone with up to 10 rest cubicles. Starting with two cubicles, each cubicle has a maximum level of 100. Each cubicle has two attributes:

- **Resting Time:** This refers to the time a worker spends inside the rest cubicle. It starts at 12 seconds and reduces by 2% with each level increase.
- **Probability of reducing the rest time:** This is the likelihood of a worker resting for half the time. It starts at 5% and only increases by 5% every 25 levels.

The Rest Zone also offers two major upgrades, which become available every five new cubicles, starting from major upgrade level 0. To purchase a major upgrade, players must meet certain conditions, including having enough rest cubicles (which is 5 multiplied by the current major upgrade level plus 1 cubicles) and ensuring that each cubicle has reached the required level for that particular major upgrade (which is the current major upgrade level plus 1 multiplied by 50). Once these conditions are met, players can execute a major upgrade that multiplies the current probability of increasing the production speed for the next gem creation by 1.75. Additionally, if all the rest cubicles are occupied, any workers who need to rest will have to wait in line until a rest cubicle becomes available. After executing the major upgrade, the player can unlock more tables to continue earning major upgrades.

3.9.5 Counter Zone

In the Counter Zone, customers place their orders at the corresponding counter and wait in line. The player can increase the number of counters up to 8, starting with two counters, to attend to more clients simultaneously. Each counter can be upgraded up to 100 levels, and has two attributes:

- **Order time:** This is the time a client takes to place their order for a new gem. It starts at 12 seconds and is reduced by 2% for each level.
- **Probability of earning the double amount of money:** This is the likelihood of receiving the double amount of money when selling a gem. It starts at 5% and increases by 5% every 25 levels. next gem.

Furthermore, it also has two major upgrades that function in the same way as those in the Rest Zone (refer to subsection 3.9.4). However, with each upgrade, the probability of earning double the amount of money is increased by 1.75.

3.9.6 Parking

The parking area limits the number of clients who can be present in the scene simultaneously. It has two attributes:

- **Capacity:** This refers to the number of available parking spaces. It starts at 2 and increases by 1 per level up to a maximum of 51, with 50 levels in total.
- **Publicity:** This factor determines the time interval between the arrival of two clients. Initially set at 4.65 seconds, it decreases by 0.35 seconds with each level upgrade, up to a maximum of 10 levels. Increasing the publicity level results in clients arriving more frequently.

3.10 Player Controls

The player controls in this game are simple and easy to use, like most games of this genre. They consist of tapping and dragging on the screen.

3.11 Design of Limitless Gems

In the previous section (3.8), a detailed explanation of each element of the game was provided. Now, in this section, the placement of these elements on the game scene map will be presented. The design of the Limitless Gem layout, characteristic of an isometric game, has been completed. Figure 3.1 showcases the visual representation of this layout.

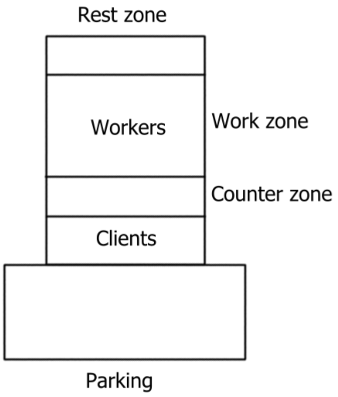


Figure 3.1: Design of Limitless Gems

3.12 Design of the surroundings

After completing the design of Limitless Gems, the only remaining aspect to design is the surrounding environment (as shown in Figure 3.2).

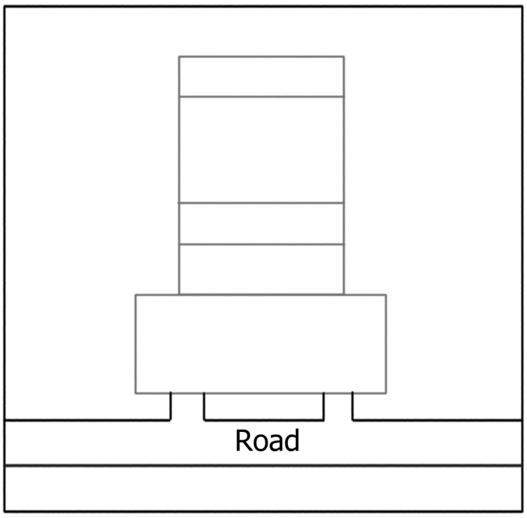


Figure 3.2: Design of the surroundings

3.13 3D Art & Animation

In the two previous sections (3.11 and 3.12) are stated the parts that need to be modelled and some of them also need to be animated. Furthermore, the general aspect of the game will be with flat colors and static shadows, except the circular shadow below the clients and workers.

3.13.1 Clients and Cars

The client models will resemble those in SuperMarket Tycoon - Shop [5], but without arms and legs, and with a simple movement animation. There will be four versions of the client model differentiated by the color of their clothes. The same will apply to the car models, but there will be five different versions. See in Figures 3.3 and 3.4.

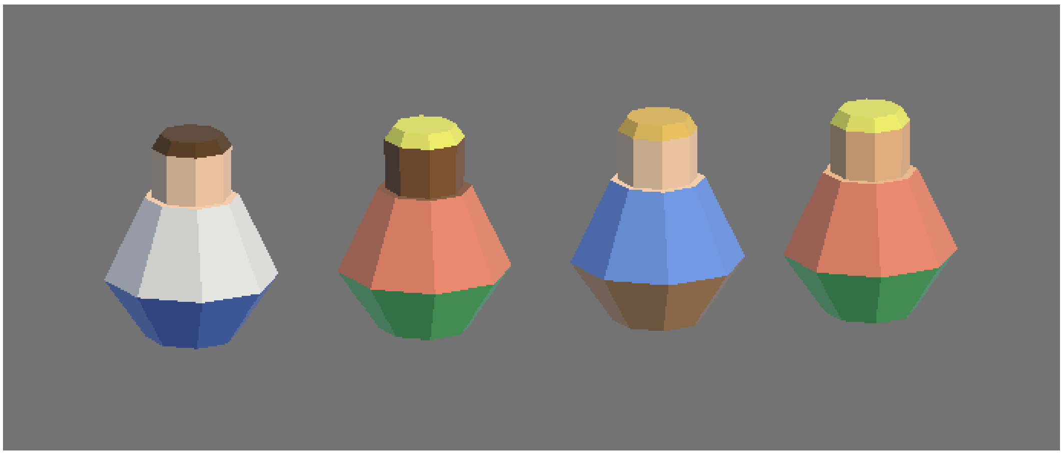


Figure 3.3: The clients' models

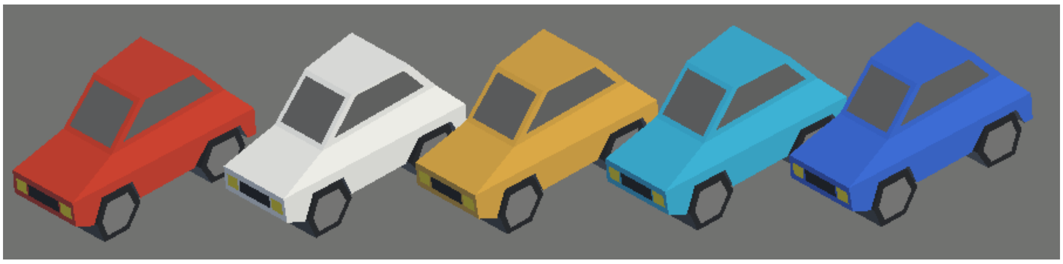


Figure 3.4: The cars' models

3.13.2 Workers

The worker models will have a similar design to the client models, featuring a lack of arms and legs, but they will be dressed in work uniforms. All workers will share the same appearance, and their animation will be similar the clients' animation (see Figure 3.5).

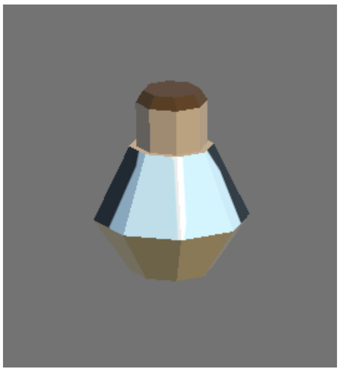


Figure 3.5: The workers' model

3.13.3 Work Zone

In the Work Zone, there will be tables with four different skins that change after each major upgrade (Figure 3.6). The table skins will be as follows:

1. A basic table design.
2. A table with added side walls.
3. A table with a combination of different colors and square-shaped table legs.
4. A table with a unique artistic table leg, while maintaining the same color scheme.

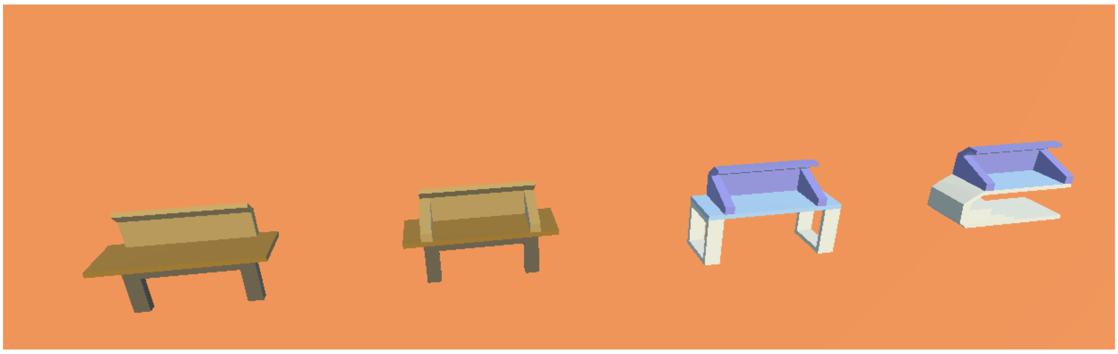


Figure 3.6: Work table skins

3.13.4 Counter Zone

The Counter Zone will feature rectangular counters, and will have three different skins that will change after each major upgrade (Figure 3.7). The counter skins will be as follows:

1. A basic rectangular counter design.
2. A counter with added side walls and a variation in colors.
3. A counter with side walls, an inclined table, and a unique color scheme.

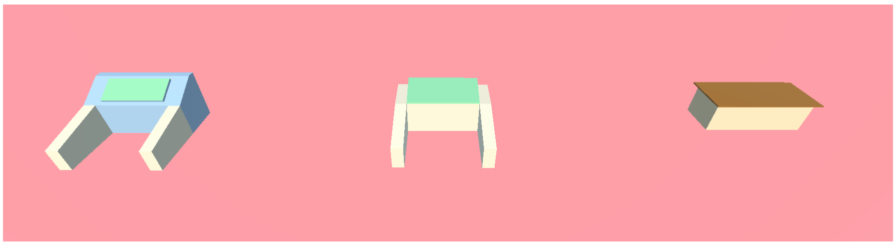


Figure 3.7: Counter skins

3.13.5 Rest Zone

In the Rest Zone, there will be rest cubicles with three different skins that will change after each major upgrade (Figure 3.8). The skins of the rest cubicles will be as follows:

1. The first skin will resemble a room, similar to a portable toilet, and will include a logo.
2. The second skin will add a door and change the logo from the first skin.
3. The third skin will feature different colors, a new logo, and a larger size compared to the previous skins.

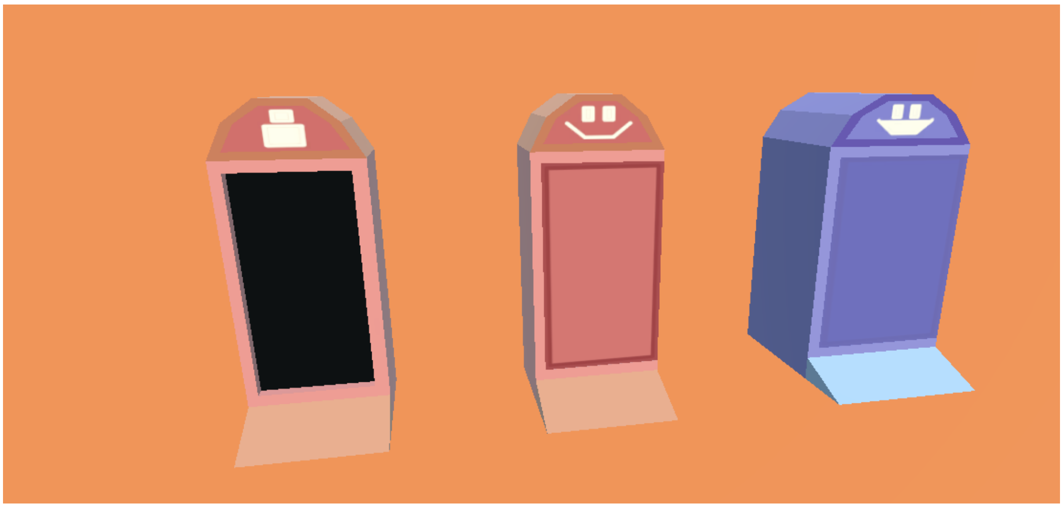


Figure 3.8: Rest cubicle skins

3.13.6 Parking

The parking lot will be designed to accommodate vehicles and will consist of two sections: surface parking and underground parking. It will have a total of eight surface parking spots and forty-three underground spaces. Access to the underground parking area will be provided by a ramp, and there will be a separate ramp for exiting. Between these ramps, there will be an elevator that displays the number of available parking spots, ensuring efficient parking management (Figure 3.9).

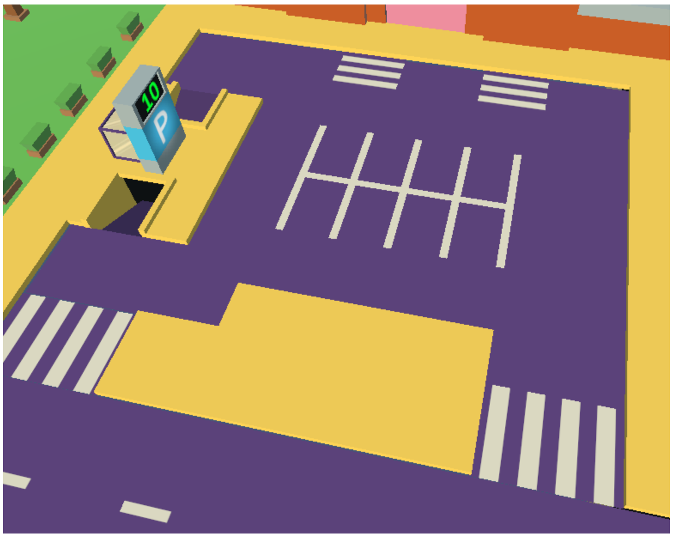


Figure 3.9: Parking of Limitless Gems

3.13.7 Limitless Gems

The interior details of the company have been described, but now let's discuss the appearance of the walls, both inside and outside. The walls of the company will be adorned with windows featuring a non-transparent, flat color that covers a portion of the wall's surface. These windows will add a modern and stylish touch to the overall design of the building (Figure 3.10).

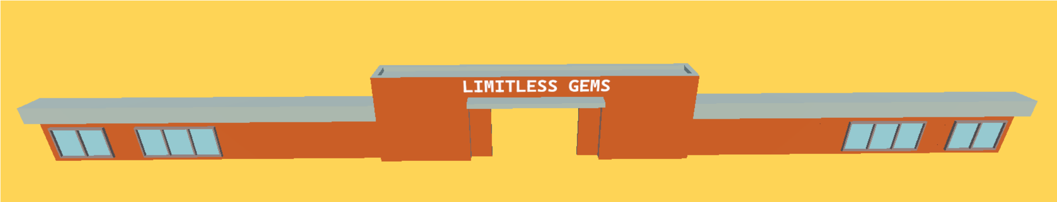


Figure 3.10: Entrance of Limitless Gems

3.13.8 Surroundings

In front of the company, there will be a road that crosses the map. Surrounding the company, there will be a flat green area with grass and palm trees. Additionally, there will be sidewalks adjacent to the road and parking area (Figure 3.11).

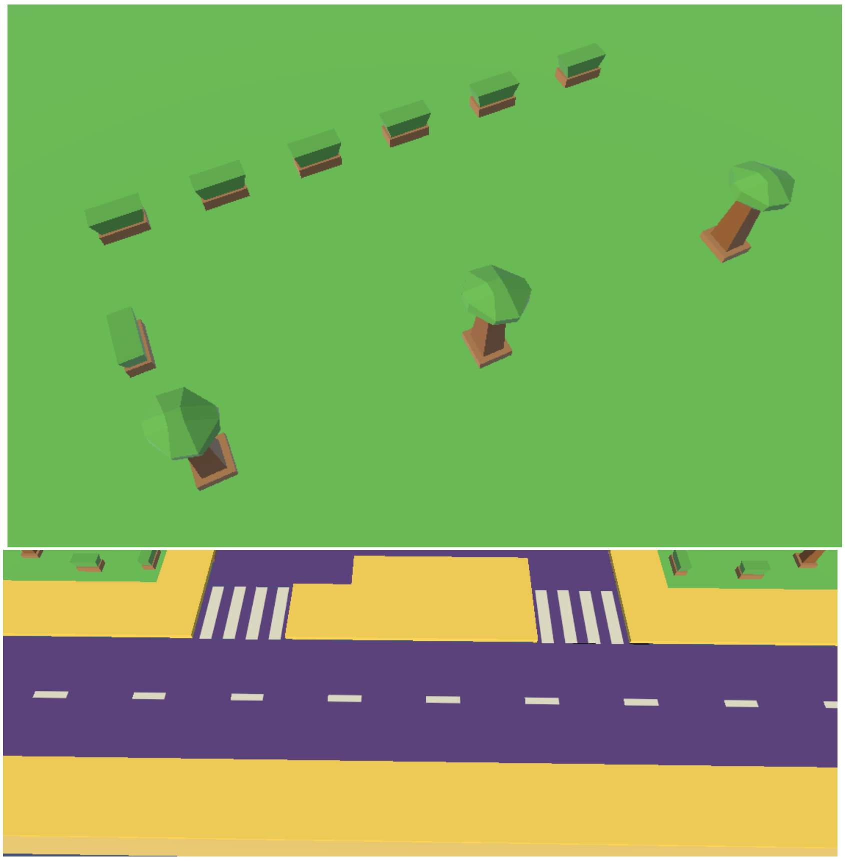


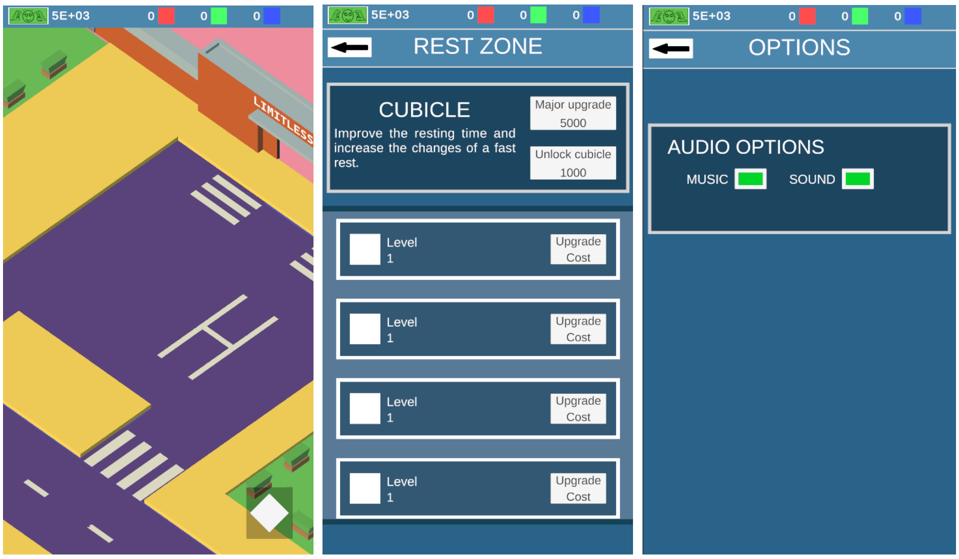
Figure 3.11: Surroundings of Limitless Gems

3.14 User Interface

The user interface (UI) for Limitless Gems will be similar to that of Supermarket Tycoon—Shop [5], featuring opaque buttons and a top row. Please refer to Figure 3.13 for a visual reference. However, the Limitless Gems UI will have fewer buttons, yet maintain the same overall appearance. The UI will consist of three canvases:

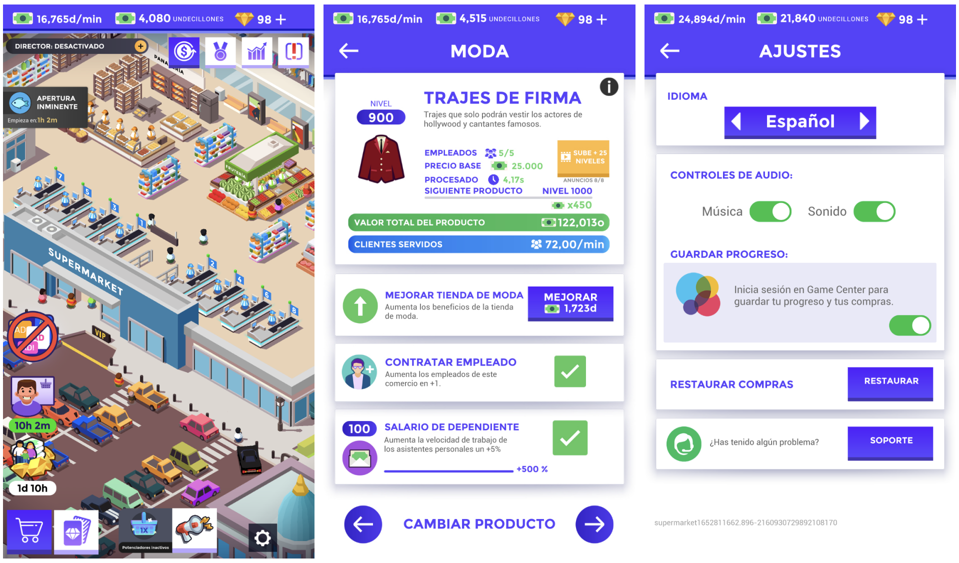
- **Main canvas:** This canvas will feature an options button and a top row displaying the total money and the available amount of each gem 3.12.
- **Upgrades canvas:** This canvas will be similar to the reference, showing information about the upgradables zone and providing buttons for upgrades 3.12.

- **Options canvas:** This canvas will allow players to adjust the audio settings, which will be divided into two groups: music and sound effects 3.12.



Improve the resting time and increase the changes of a fast rest.

Figure 3.12: User Interface from Gem Maker Tycoon



Trajes que solo podrán vestir los actores de hollywood y cantantes famosos.

Aumenta los beneficios de la tienda de moda.

Aumenta los empleados de este comercio en +1.

Aumenta la velocidad de trabajo de los asistentes personales un +5%

Inicia sesión en Game Center para guardar tu progreso y tus compras.

¿Has tenido algún problema?

supermarket1652811662.896-2160930729892108170

Figure 3.13: User Interface from Idle Supermarket Tycoon — shop [5]

3.15 Sound effects & Music

Gem Maker Tycoon will feature background music and sound effects designed to enhance the player's experience. The arcade-style music will set the tone for the game, while the sound effects will signal different actions or events. For example, the sound of gems being collected, an upgrade being purchased, or a button being pressed. These sound effects will be carefully crafted to provide a satisfying feedback to the player, reinforcing their sense of progress and achievement.

In addition, Gem Maker Tycoon will offer players the ability to adjust the volume of both the background music and sound effects in the options menu, allowing them to customize their audio experience to their liking.

Chapter 4

Work Development and Results

Contents

4.1 Work development

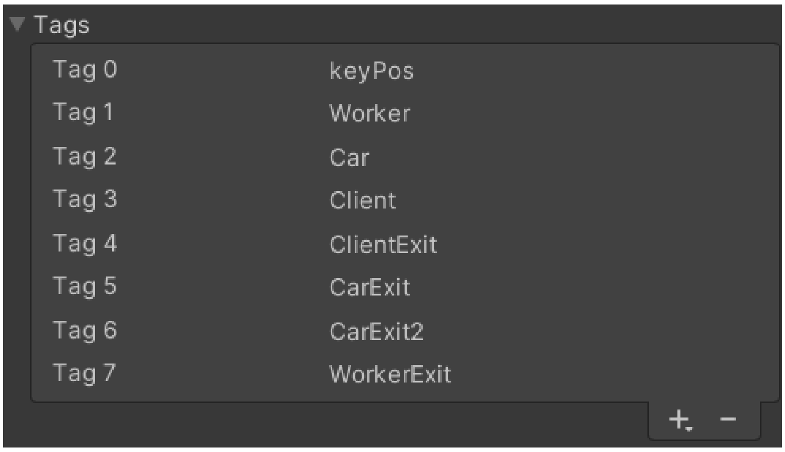
In this section, a comprehensive and detailed explanation will be provided, delving into the most crucial aspect of the development process.

4.1.1 Clients and Workers behaviour

The development process encompasses the integration of clients and their cars as integral components. Consequently, when referring to clients, it implicitly includes their associated cars. These clients follow a predetermined path that can be easily tailored to meet diverse requirements. For instance, incorporating additional zones or modifying the path layout can be seamlessly achieved by the developer. Triggers were employed to facilitate this functionality. Initially, the system performed adequately during testing with a limited number of clients. However, challenges arose when testing with the maximum number of clients. To overcome this, a solution was implemented wherein the client's tag (see Figure 4.1) was dynamically modified based on their position along the path.

Likewise, the worker system operates based on the aforementioned path mechanism. As mentioned earlier, this adaptable path system can cater to various requirements. However, similar challenges were encountered within this context. To overcome these challenges, each worker was successfully associated with their respective work table using appropriate referencing techniques. Furthermore, the tags of the workers were dynamically modified to ensure smooth functionality.

By implementing these strategies, both the clients and workers are effectively managed within the game environment, allowing for seamless navigation and interaction.


▼ Tags

Tag 0	keyPos
Tag 1	Worker
Tag 2	Car
Tag 3	Client
Tag 4	ClientExit
Tag 5	CarExit
Tag 6	CarExit2
Tag 7	WorkerExit

Figure 4.1: Tags used to avoid undesired triggers

In the following sections, two key components related to the behavior of clients and workers will be discussed: the NavMesh design and the Spawn System.

NavMesh design

In addition to utilizing different tags and references to prevent undesired triggers, an important element in achieving a robotic-like path was the implementation of a NavMesh design inspired by Codigames' games [3]. The NavMesh design employed in this project involved using multiple planes that are not rendered by the camera, as they are placed in a layer not utilized by the camera. This approach allows for more efficient movement, although it can result in somewhat robotic motion as the clients and workers follow a relatively straight path.

The NavMesh design is particularly crucial in the parking area, where cars need to adhere to specific paths resembling those commonly found in parking lots. By utilizing the NavMesh approach, the cars can accurately navigate along the designated routes. Refer to the figure shown in Figure 4.2 for a visual representation of the NavMesh design.

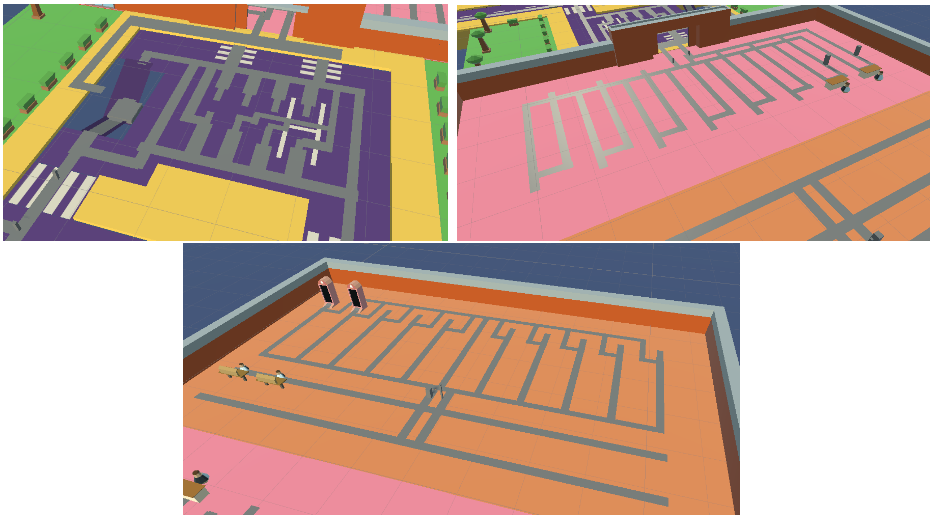


Figure 4.2: NavMesh design for achieving robotic-like path

Spawn System

In an idle game where clients continuously arrive and depart, a robust spawn system is vital to ensure a dynamic and engaging gameplay experience. The spawn system is responsible for generating new clients at regular intervals and updating their appearance, along with their associated cars. This seamless flow of clients is crucial for maintaining the game's momentum and keeping players engaged.

To efficiently manage the spawning process, an object pool is employed. At the start of the game, the object pool creates a predetermined number of client instances and deactivates them. When it's time for a new client to appear on the screen, the system simply activates an available client from the object pool. This approach eliminates the need for creating and destroying client objects dynamically, which can lead to performance overhead and memory inefficiency.

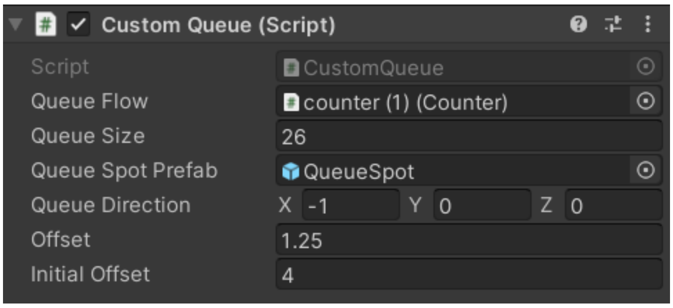
By utilizing an object pool, the game benefits from efficient memory management and improved performance. The object pool optimizes resource utilization by reusing existing client instances, minimizing the overhead of object instantiation. This ensures a smooth and optimized spawning process, contributing to a seamless gaming experience.

4.1.2 Queues behaviour

In this project, there are two types of queues: the Normal Queue, consisting of clients, workers, or cars, and the Better Queue, which organizes the active queues of a zone based on the number of people in each queue, prioritizing the one with the fewest people.

Normal Queue

Let's begin by discussing the Normal Queue and its various components. The Normal Queue allows for adjustments in size, direction, and distance between elements, serving as the foundation for the specific queues that will be explained in detail.



Custom Queue (Script)

Script	CustomQueue
Queue Flow	counter (1) (Counter)
Queue Size	26
Queue Spot Prefab	QueueSpot
Queue Direction	X -1 Y 0 Z 0
Offset	1.25
Initial Offset	4

Figure 4.3: Normal Queue

- **Counter Queue:** When a client reaches the end of the counter queue, a sequence is triggered using DoTween [7] to ensure the correct order of events:

  1. The client places an order for a gem, which takes a few seconds.
  2. The system checks if the requested gem already exists. If it does, the client can take the gem immediately. Otherwise, a gem request is created.
  3. If the gem is already available, the client takes it. Otherwise, the client waits for a worker to create the gem.
  4. After taking the gem, the client moves to the exit spot of the counter.
  5. Finally, the client moves to the next position on their path.

- **Rest Cubicle Queue:** When a worker reaches the end of the rest cubicle queue, a simpler sequence is executed:

  1. The worker takes a break for a few seconds.
  2. The worker moves to the next position on their path.

- **Road Queue:** Unlike the previous queues, the parking queue does not follow a predefined sequence. When a client reaches the end of the road queue, they will attempt to park their car if there are available parking spots. If there are no free spots, the client will wait until a spot becomes available.

Better Queue

The Better Queue comes into play when a client or worker enters a zone with multiple Normal Queues, such as the Rest Zone or Counter Zone. In these cases, the individual must choose the queue with the fewest number of people. To facilitate this, the Better Queue was implemented, utilizing a priority queue structure represented as a list. This system maintains references to every active queue in the zone and orders them based on their level of occupancy, prioritizing the queue with the least number of people.

These queues play a crucial role in controlling the movement and interaction of both clients and workers within the game environment. By understanding the intricacies of each queue and its associated behaviors, developers can ensure smooth gameplay flow and provide an immersive gaming experience.

4.1.3 Parking and Road behaviour

The Road Zone and the Parking Zone are interconnected to facilitate communication between the car queue and the parking area. This enables the car queue to stay informed about the availability of parking spots.

However, a challenge arises when a car is about to enter the parking area. Until it reaches the next Key Trigger on its path, the status of the parking area does not update. This can potentially lead to issues if the car behind it reaches the end of the queue and attempts to enter the parking area, only to get stuck at the Key Trigger indefinitely. To address this challenge, a key solution was implemented: reducing the distance between the end of the queue and the Key Trigger.

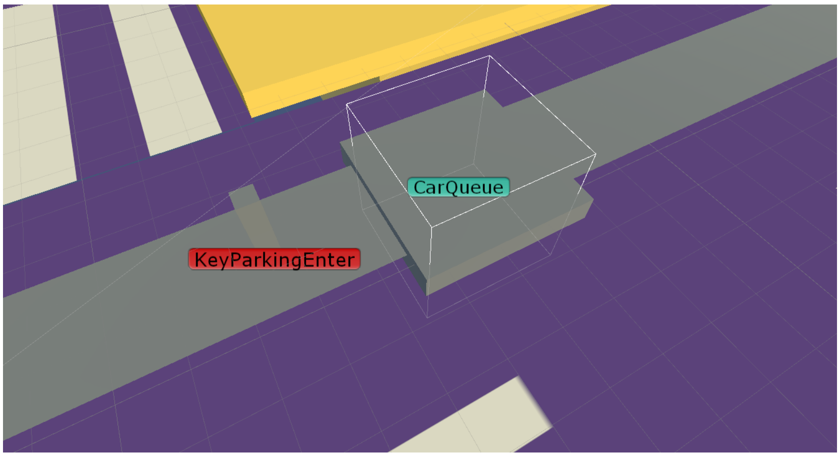


Figure 4.4: Car Queue and Key Trigger

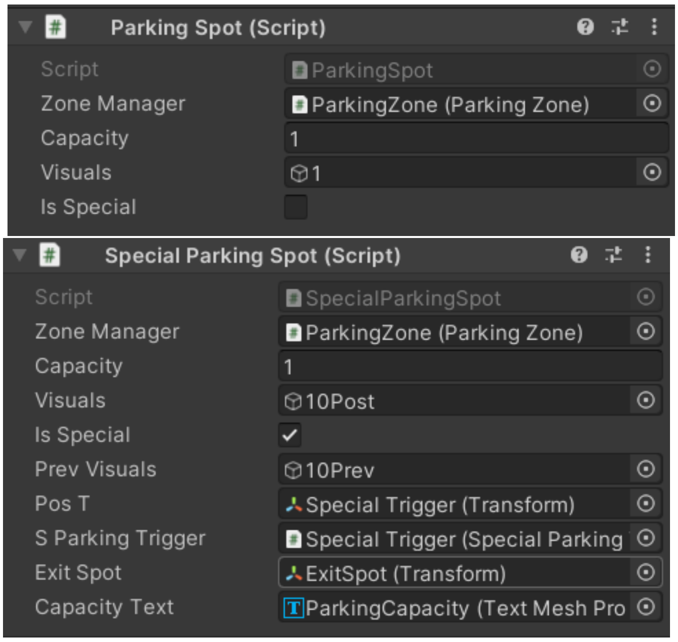
By minimizing the distance between the end of the Car Queue and the Key Trigger, the system ensures that when the second car reaches the end of the queue, the first car has already reached the Key Trigger and the parking status is promptly updated. This adjustment ensures a smoother flow of cars and prevents cars from becoming stuck at the Key Trigger, enhancing the overall gameplay experience.

Parking Spots

Within the parking zone, there are two types of parking spots: Regular Parking Spots and Special Parking Spots. The Regular Parking Spots are designed to accommodate a single car at a time, providing a straightforward parking experience. On the other hand, the Special Parking Spots represent an underground parking area that offers additional features and functionalities.

In the game, both types of parking spots can be easily customized to support any desired number of cars, allowing for a flexible and tailored gameplay experience. However, the Special Parking Spots do have certain unique characteristics due to their underground nature. When clients park their cars in the Special Parking Spots, they undergo a simulated teleportation process to the next location, emulating the concept of an elevator. This adds an extra layer of immersion and realism to the game.

Additionally, the Special Parking Spots require managing their visuals to enhance the underground parking ambiance. This may involve incorporating appropriate lighting effects, textures, or other visual elements to create a distinct and engaging environment for players.



Parking Spot (Script)

Script	ParkingSpot
Zone Manager	ParkingZone (Parking Zone)
Capacity	1
Visuals	1
Is Special	□

Special Parking Spot (Script)

Script	SpecialParkingSpot
Zone Manager	ParkingZone (Parking Zone)
Capacity	1
Visuals	10Post
Is Special	✓
Prev Visuals	10Prev
Pos T	Special Trigger (Transform)
S Parking Trigger	Special Trigger (Special Parking
Exit Spot	ExitSpot (Transform)
Capacity Text	ParkingCapacity (Text Mesh Pro

Figure 4.5: Regular and Special parking spot

4.1.4 Counter Zone and Work Zone behaviour

The Counter Zone and the Work Zone are interconnected components in the game, working in tandem to ensure the availability of gems and facilitate smooth gameplay interactions. Understanding the behavior of both zones is crucial for creating a cohesive and immersive gaming experience. In this section, the interconnected behavior of the Counter Zone and the Work Zone, as well as their respective functionalities, will be explored.

The Counter Zone relies on the Work Zone to determine the current quantity of gems available. When a client requests a gem at the Counter, the Counter Zone communicates with the Work Zone to check its availability. This allows the client to be informed promptly if the desired gem is currently available or if they need to wait for it to be created. If the requested gem is not immediately available, the client can place an order, indicating their preference to the Work Zone.

Upon a worker's arrival at their assigned table in the Work Zone, they perform several checks to ensure efficient gem creation. Firstly, they examine if there are any pending gem requests from clients. If there are pending requests, the worker randomly selects one of the most recent requests to fulfill. On the other hand, if there are no pending requests, the worker chooses to create a random gem.

This interconnected behavior between the Counter Zone and the Work Zone ensures a dynamic and responsive gameplay experience. The Work Zone prioritizes gem creation based on pending requests, ensuring timely delivery to clients. This system not only enhances gameplay flow but also adds an element of unpredictability and variety as workers create a mix of random and requested gems.

Work Table and Counter

The Work Table and the Counter are two essential components in the game, each with its unique sequence of actions. However, they share some common elements in terms of visual feedback and management. In this section, the similarities between the Work Table and the Counter and their visual management will be explored.

Both the Work Table and the Counter incorporate a slider with a circular form that fills up gradually. This visual element serves as an indicator to the player, signaling that an action is in progress or being completed. As the slider fills, it provides a visual representation of the progress made towards achieving a specific task or action. Once the slider is fully filled, it signifies that the corresponding action has been successfully completed.

This visual feedback mechanism adds an important layer of interactivity and engagement to the gameplay experience. It informs the player of ongoing processes, such as gem creation or order fulfillment, and allows them to track the progress visually. The circular slider provides a clear visual cue that captures the player's attention.

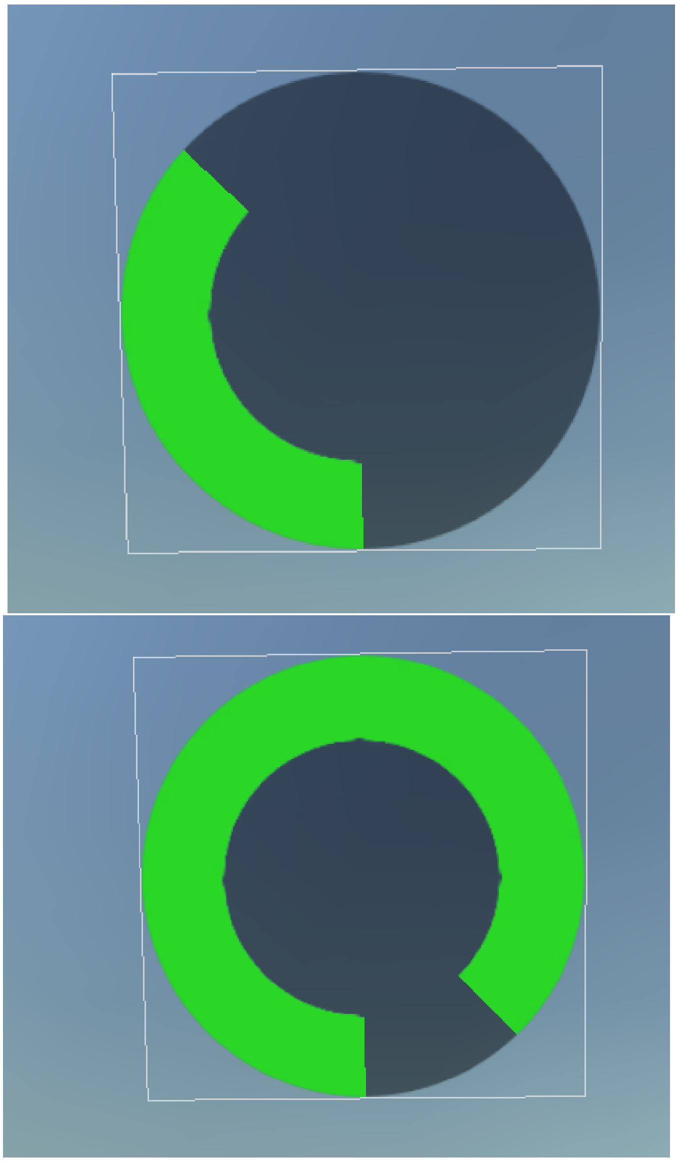


Figure 4.6: Circular slider

4.1.5 Rest Zone behaviour

The Rest Zone operates differently from the previous zones in terms of its interactions and communication with other zones. Unlike the other zones, the Rest Zone functions independently and does not have any direct communication or interconnectedness with other zones.

Similar to the Counter Zone, the Rest Zone also incorporates the concept of Normal Queues (4.1.2) and the Better Queue (4.1.2). These queue systems play a crucial role in managing the movement and interaction of individuals within the Rest Zone, ensuring an organized and efficient flow of workers.

The Normal Queues in the Rest Zone allow workers to queue up for their designated rest cubicles, ensuring that each worker has a dedicated space for relaxation. The Better Queue system assists workers in selecting the queue with the fewest number of people, facilitating a fair and balanced distribution of available rest cubicles.

Although the Rest Zone operates independently, it shares certain similarities with the Counter Zone in terms of its queue management. These similarities provide consistency and familiarity for players, making it easier to understand and navigate the game mechanics.

Rest Cubicle

The Rest Cubicle is an essential component of the Rest Zone, providing a dedicated space for workers to take a break and recharge. In this section, the functionality of the Rest Cubicle and its role within the game will be explored.

Each Rest Cubicle is designed to accommodate a single worker at a time. Workers can determine the most optimal Normal Queue (refer to Section 4.1.2) to join, thanks to the Better Queue (refer to Section 4.1.2) implemented in the Rest Zone. By utilizing the Better Queue, workers can choose the queue with the least number of people, minimizing their waiting time and ensuring a fair distribution of resting opportunities.

Once a worker reaches the front of the queue and enters a Rest Cubicle, a specific sequence of actions is triggered. These actions may include various visual and auditory cues, such as animations, sound effects, or visual indicators like the circular slider shown in Figure 4.6. These cues serve to inform players that the worker is currently taking a break and is not available for other tasks.

The duration of the break can be predetermined or dynamically adjusted based on various game-specific factors or mechanics. This flexibility allows for fine-tuning the balance between worker rest periods and overall gameplay dynamics.

Once the designated break duration is complete, the worker will conclude their rest in the Rest Cubicle and proceed to the next position along their predefined path.

4.1.6 UI System

The UI System is an integral component of the game, serving as a crucial connection between players and the gameplay experience, particularly in the context of an idle game [11]. It provides players with the means to interact and engage with the game mechanics, allowing them to perform various actions and make decisions. The UI System comprises three key components: the Top Row, Zone Layer, and Options Layer (as illustrated in Figure 3.12).

To enhance compatibility and optimize the user experience across different mobile devices, the UI System incorporates the use of the Safe Area Helper [13] asset. This asset assists in adapting the UI elements to accommodate different screen sizes and resolutions, ensuring that the UI is effectively utilized and displayed within the available screen space.

One important aspect of the UI System is its ability to reflect the current state of the game through dynamic updates. For instance, when the amount of money changes, the corresponding text element displaying the money is automatically updated to reflect the new value. Similarly, the quantity of gems is dynamically updated in the UI as it changes. This ensures that players always have real-time information about their financial resources and gem quantities.

The UI System also manages the enabling and disabling of buttons based on specific conditions. When a player has sufficient funds to perform a particular action, the corresponding button is enabled, allowing them to interact with it. Conversely, if the player lacks the necessary funds, the button is disabled to prevent unintended actions. It's worth noting that due to the large number of buttons present in the game, their updates are triggered when their respective zone layer is opened or if the zone layer is already open. This optimization helps reduce unnecessary updates and improve overall performance.

Top Row

The Top Row plays a crucial role in conveying important information to the player, providing a quick overview of the current situation within the game. It primarily focuses on displaying the updated amount of in-game currency and the quantity of each different gem available to the player.

The top row's concise and accessible presentation of this key information ensures that players are well-informed about their resources and gem quantities, enabling them to strategize and make optimal choices throughout the game. This feature enhances the player's sense of control and progress, as they can clearly track their financial status and gem inventory in real time.

Zones' User Interface

There are four Zone's User Interface in the game, each representing one of the available zones. These zones include the Counter, Rest, Work, and Parking zones. These interfaces provide players with the ability to upgrade and customize various elements within their respective zones, such as counters, rest cubicles, work tables, and parking capacity.

The Counter, Rest, and Work layers are categorized as Special Zone Layers, where players have the ability to upgrade specific objects within each respective zone. These upgrades not only enhance the visual appearance of the objects but also improve the overall statistics and efficiency of the corresponding zones. Furthermore, players are presented with opportunities to perform major upgrades, which remove level limits, unlock additional spots, and introduce visual changes to the objects in these zones.

On the other hand, the Parking Layer is considered a Zone Layer. Within the Parking Layer, players can increase the capacity of the parking area, and increase the number of clients per minute. While major upgrades may not be available for the parking zone, the visuals of the parking area dynamically change based on its current capacity. This provides players with a visual representation of the parking area's utilization and overall functionality.

Now, let's provide a more detailed explanation of the Zone's User Interface:

- **ObjectInLine**

  The ObjectInLine is a crucial element within a Zone's User Interface. It displays the current level of an object within the corresponding zone and provides information on the cost to upgrade that object. If the object has reached its maximum level, the ObjectInLine indicates this as well.

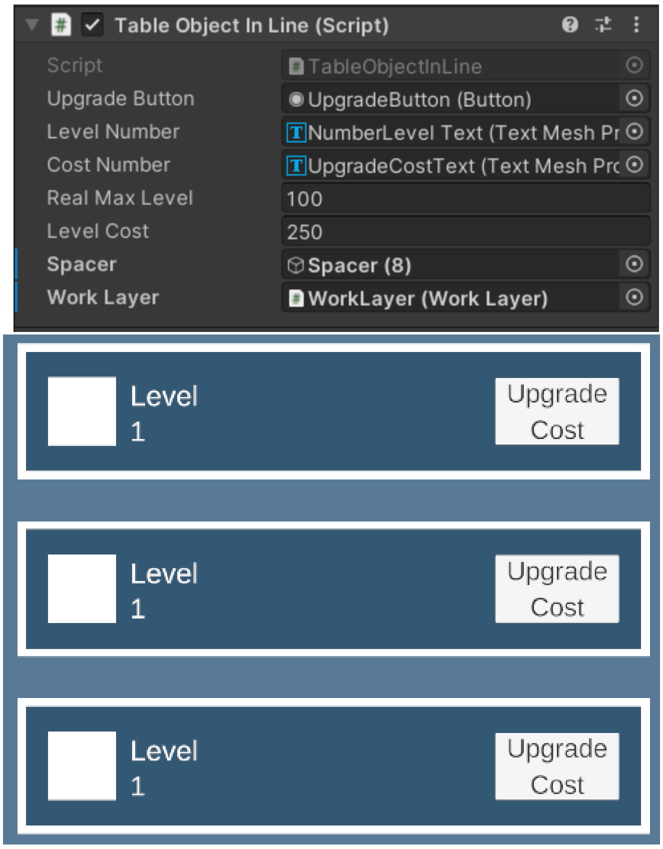


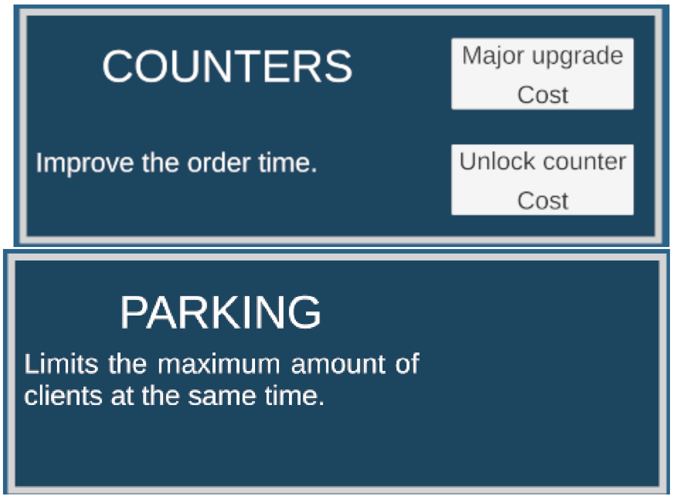
Table Object In Line (Script)

Script: TableObjectInLine
Upgrade Button: UpgradeButton (Button)
Level Number: NumberLevel Text (Text Mesh Pr
Cost Number: UpgradeCostText (Text Mesh Prc
Real Max Level: 100
Level Cost: 250
Spacer: Spacer (8)
Work Layer: WorkLayer (Work Layer)

Figure 4.7: ObjectInLine display

- **Special Zone Layer vs Zone Layer**

  The main difference between a Zone Layer and a Special Zone Layer lies in the additional functionality offered by the Special Zone Layer. In addition to the core features found in a Zone Layer, the Special Zone Layer introduces additional buttons and options that unlock new spots within the corresponding zone or enable major upgrades. These buttons provide players with the ability to expand their zones, and unlock new opportunities.



Improve the order time.

Limits the maximum amount of clients at the same time.

Figure 4.8: Difference between Special Zone Layer and Zone Layer

Options layer

The Options Layer serves a simple yet essential purpose in enhancing players' enjoyment of the game. It provides players with the ability to customize their audio experience by toggling the music and sound effects on or off. To facilitate this, the Options Layer incorporates two distinct audio groups: one for the background music and another for the in-game sounds. This separation allows players to independently control the audio elements according to their preferences.

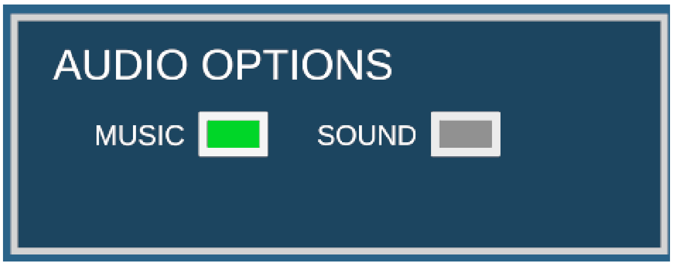


Figure 4.9: Options layer toggles

4.1.7 Camera system

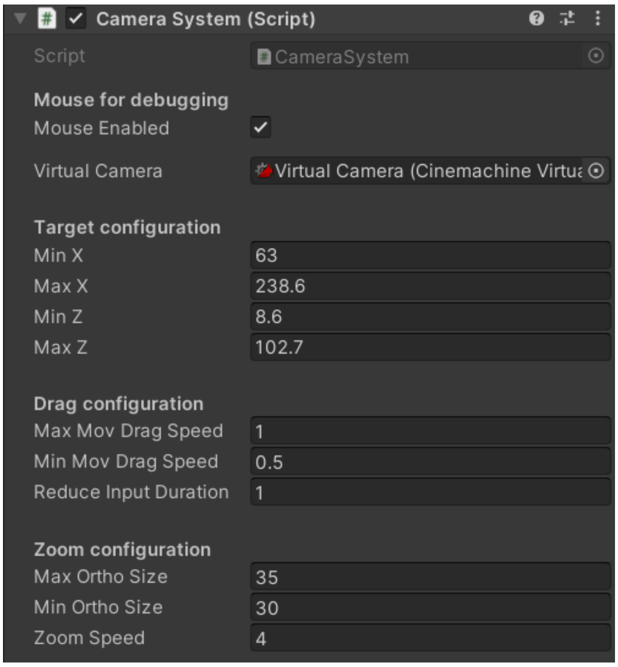
The camera behavior is a crucial aspect of this project, providing essential functionality for controlling the in-game camera. It enables players to manipulate the camera view by dragging their finger on the screen, creating a smooth and responsive movement. Unity offers the Cinemachine asset [15] to simplify the implementation of camera-related tasks, making it easier for developers to achieve the desired camera behavior.

In addition to camera movement, the system supports zooming, which is accomplished through the familiar pinch gesture [6] using two fingers. This intuitive interaction allows players to adjust the zoom level and explore the game environment with convenience. The parameters for both dragging and zooming can be customized according to the developer's preferences, offering flexibility in adapting the camera behavior to meet specific game requirements.

An important consideration in the camera behavior is the relationship between dragging speed and zoom level. When the player zooms in, the dragging speed is automatically reduced. This adjustment enhances control and precision when navigating the game world at closer zoom levels, allowing for more accurate interactions and exploration.

To ensure that the camera stays within the designated game area, the use of a Cinemachine Confiner is employed [16]. This powerful feature restricts the movement of the camera within a defined boundary, effectively keeping it within the intended game space. It is worth noting that the movement limitations imposed by the Cinemachine Confiner apply solely to the camera itself. In contrast, the camera's target retains its freedom of movement, allowing it to move without restrictions. However, it is important to highlight that the camera's target also adheres to certain movement limitations set by the game design. This ensures that both the camera and its target operate within the defined boundaries.

Furthermore, the intention was for the camera to maintain momentum after the player releases their finger, providing a smooth and natural movement. However, it should be noted that this feature currently functions only when using a mouse input. During testing with touch inputs, the momentum preservation feature does not currently operate as intended. This discrepancy between mouse and touch input is an area for improvement and further development in order to provide a consistent and seamless camera experience across all input methods.



Camera System (Script)

Script	CameraSystem
Mouse for debugging	
Mouse Enabled	✓
Virtual Camera	Virtual Camera (Cinemachine Virtua
Target configuration	
Min X	63
Max X	238.6
Min Z	8.6
Max Z	102.7
Drag configuration	
Max Mov Drag Speed	1
Min Mov Drag Speed	0.5
Reduce Input Duration	1
Zoom configuration	
Max Ortho Size	35
Min Ortho Size	30
Zoom Speed	4

Figure 4.10: Camera configuration

4.1.8 Assets development

This section presents the UV maps and textures of the models created for the video game. The UV maps and textures play a crucial role in enhancing the visual appeal of the assets.

Models

Now, the UV maps and textures of every model made during the development of the video game will be shown.

- **Clients:**

  The UV maps and textures for the client models are displayed in Figure 4.11.

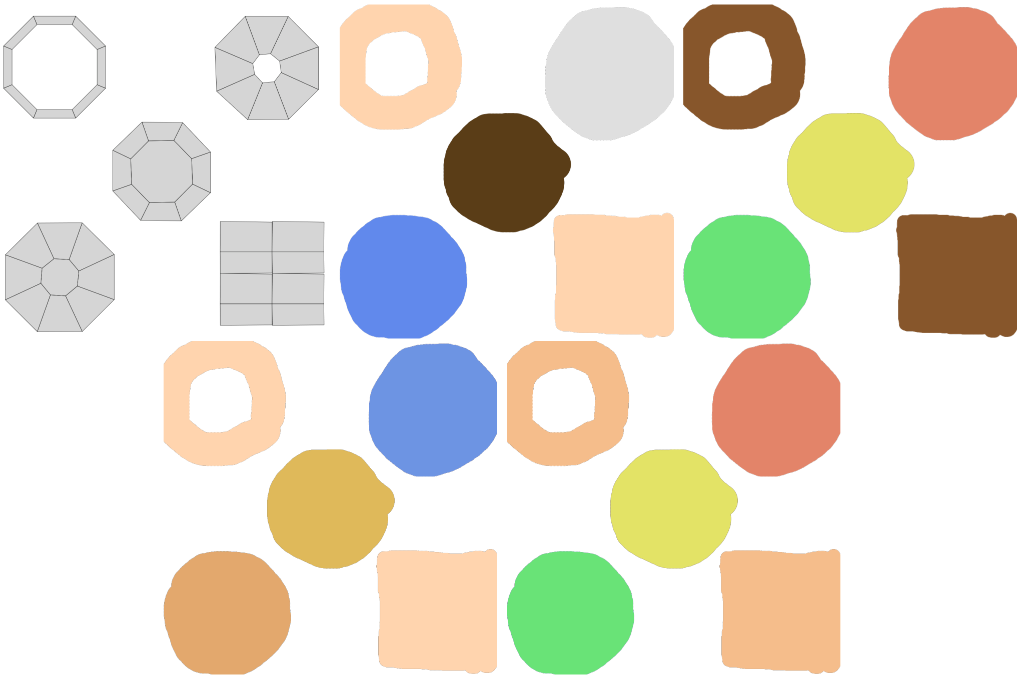


Figure 4.11: UV maps and textures for clients

- **Cars:**

The UV maps and textures for the car models are shown in Figure 4.12.

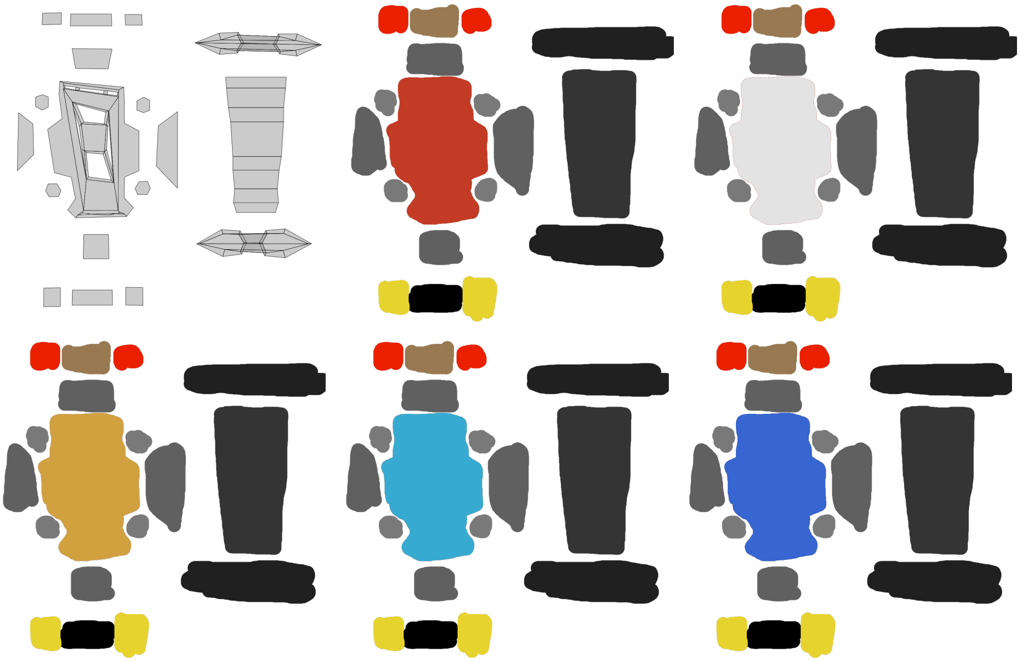


Figure 4.12: UV maps and textures for cars

- **Workers:**

  The workers share the same UV maps as the clients shown in Figure 4.11. The texture for the workers is displayed in Figure 4.13.

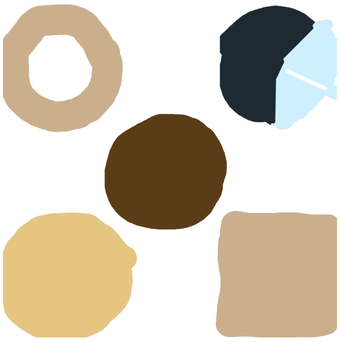


Figure 4.13: Texture for workers

- **Work Tables:**

  The UV maps and textures for the work tables are presented in Figures 4.14, 4.15, 4.16, and 4.17.

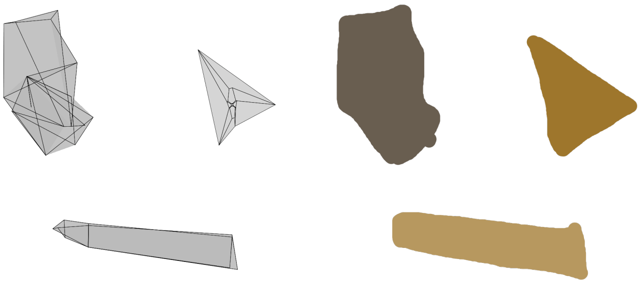


Figure 4.14: UV map and texture for Work Table 1

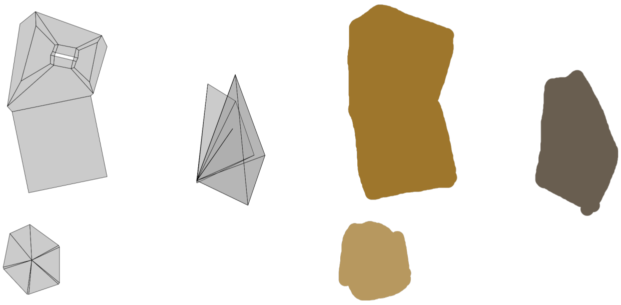


Figure 4.15: UV map and texture for Work Table 2

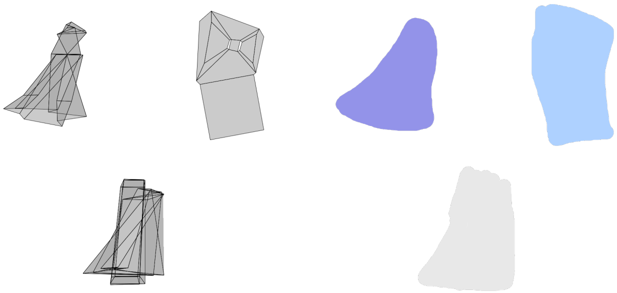


Figure 4.16: UV map and texture for Work Table 3

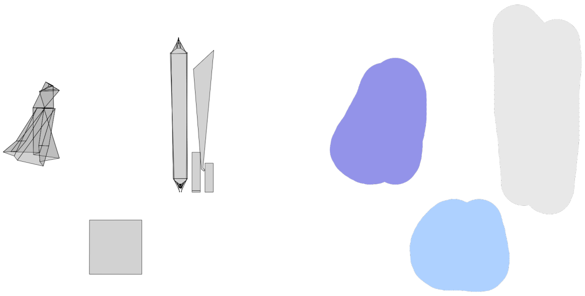


Figure 4.17: UV map and texture for Work Table 4

- **Counters:**

  The UV maps and textures for the counters are showcased in Figures 4.18, 4.19, and 4.20.

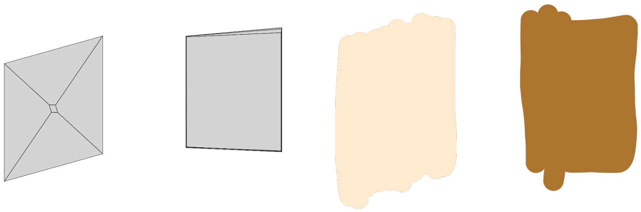


Figure 4.18: UV map and texture for Counter 1

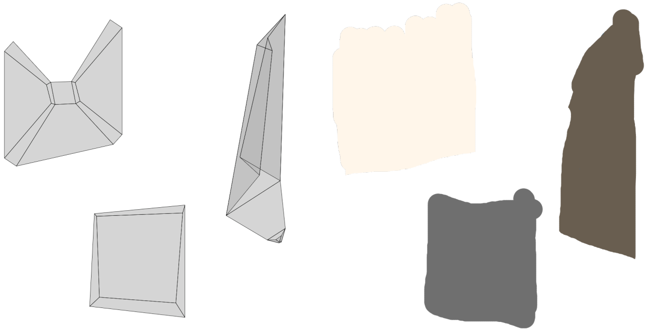


Figure 4.19: UV map and texture for Counter 2

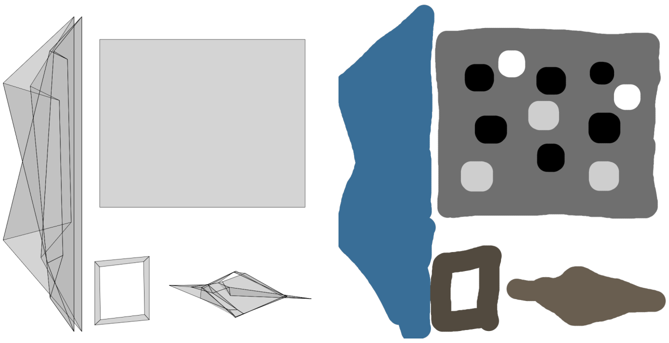


Figure 4.20: UV map and texture for Counter 3

- **Rest Cubicles:**

The UV maps and textures for the rest cubicles and doors are depicted in Figures 4.21, 4.22, and 4.23.

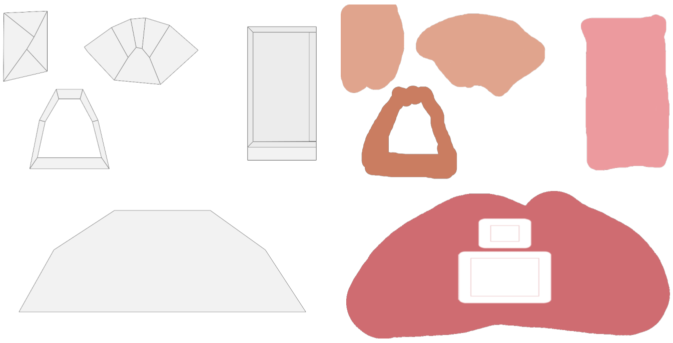


Figure 4.21: UV map and texture for Rest Cubicle 1

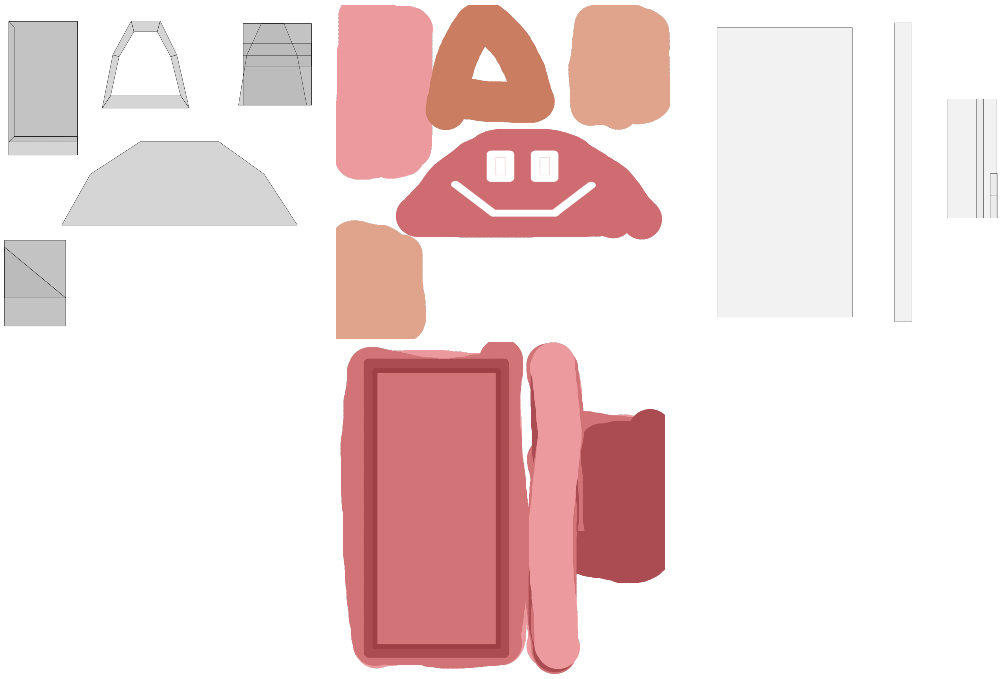


Figure 4.22: UV maps and textures for Rest Cubicle 2 and door

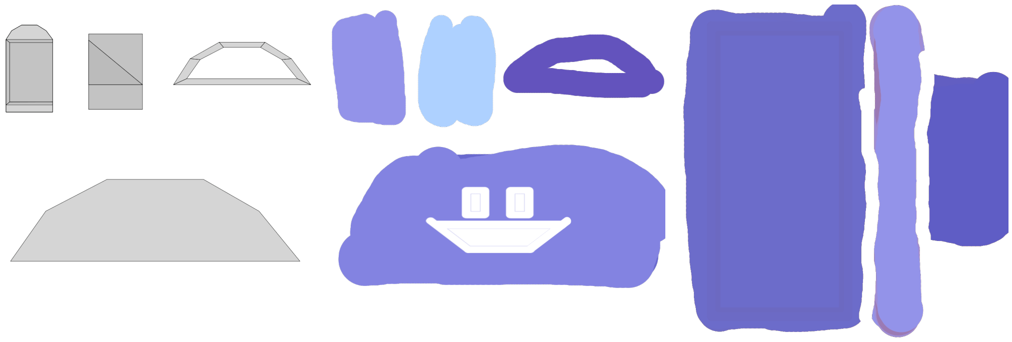


Figure 4.23: UV map and texture for Rest Cubicle 3 and texture for door

- **Parking:**

  The UV map and texture for the underground parking are presented in Figure 4.24.

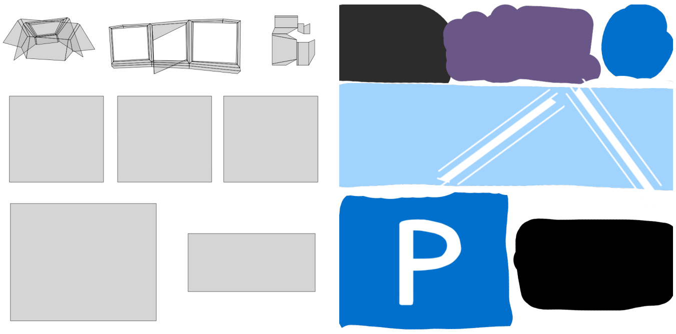


Figure 4.24: UV map and texture for Underground Parking

- **Limitless Gems:**

  The UV maps and textures for the entrance, wall, and window of the Limitless Gems asset are shown in Figures 4.25, 4.26, and 4.27.

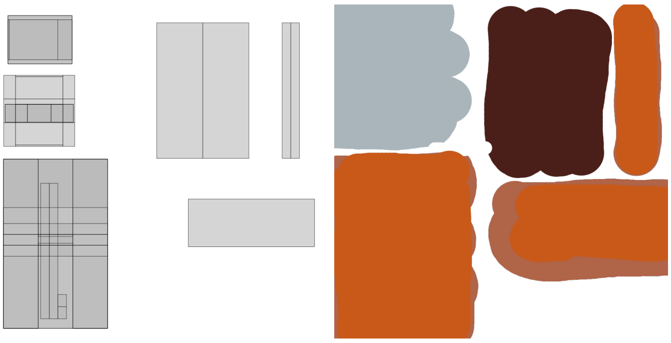


Figure 4.25: UV map and texture for Entrance

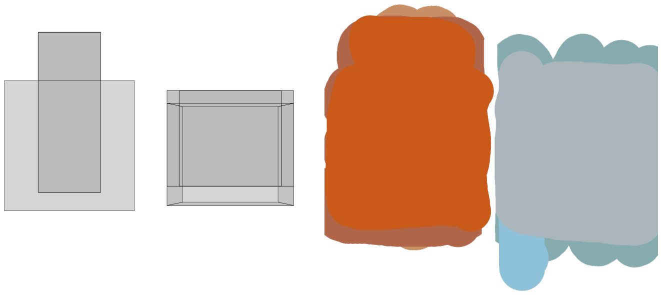


Figure 4.26: UV map and texture for Wall

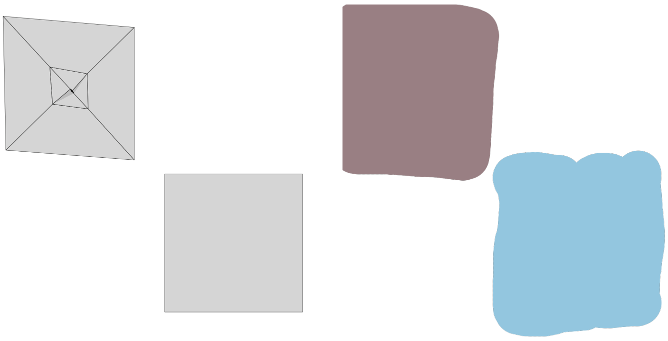


Figure 4.27: UV map and texture for Window

- **Surroundings:**

  The UV maps and textures for the road, bush, and tree in the surroundings are displayed in Figures 4.28, 4.29, and 4.30.

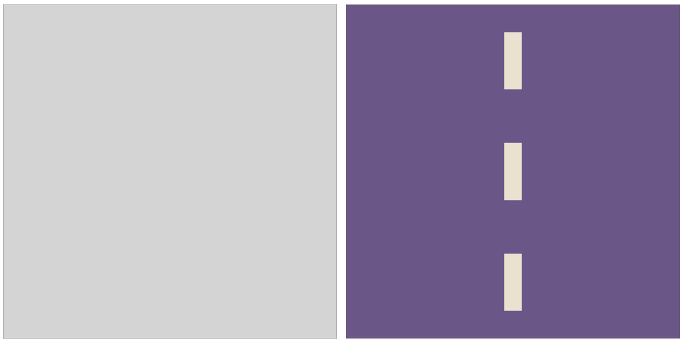


Figure 4.28: UV map and texture for Road

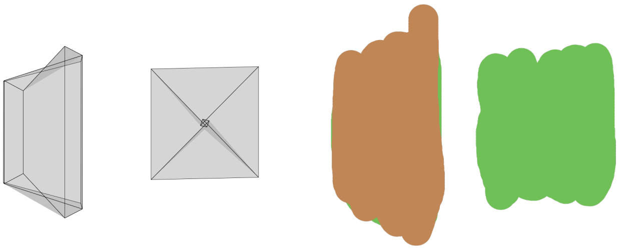


Figure 4.29: UV map and texture for Bush

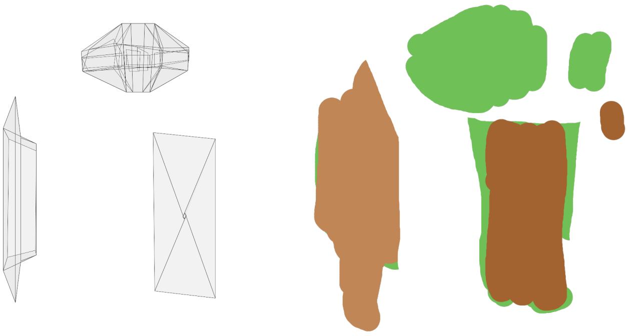


Figure 4.30: UV map and texture for Tree

- **Money particles:**

  These particles were added later during the development in order to give visual satisfaction to the player whenever they earned money.

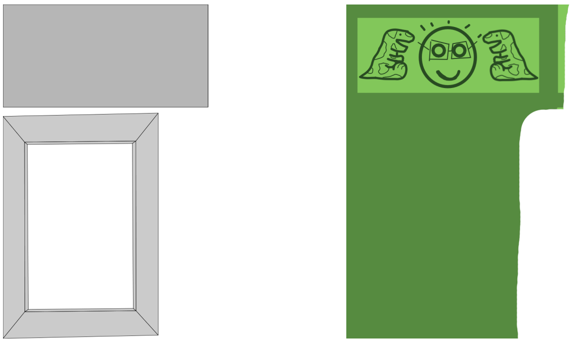


Figure 4.31: UV map and texture for Money particles

Please refer to Section 4.2.4 for a comprehensive visual representation of the developed assets.

Animations

Different animations were implemented for various models in the game:

- **Clients and Workers:** The client and workers models incorporate rotation changes using a sequence from DoTween [7].
- **Static Workers:** The static worker models also utilize an animator component for animation.
- **Rest Cubicle door:** The door of the Rest Cubicle 2 and 3 is animated to open and close.

To see the animations in action, please visit the following links:

- Agent animations: https://youtu.be/cENx3yPpkiY
- Door animation: https://youtube.com/shorts/Vp8Ipko27Hk.

4.1.9 Music & Sounds

The main music featured in the game is composed by Dark Fantasy Studio [14]. You can listen to the main theme here: https://www.youtube.com/watch?v=vX2Wp5Wk0-s.

Regarding the game's sound effects, during a thorough exploration of the Dark Fantasy Studios sound bundle, there were limited suitable sounds that aligned with the game's theme. As a result, there is only one sound effect in the game, which plays when a worker has rested. You can listen to the sound effect here: https://youtube.com/shorts/tzftoiKhgRg.

4.2 Results

This section presents the achieved results in comparison to the initial objectives of the project. It evaluates the actual implementation in relation to the planned execution and provides appropriate justifications.

4.2.1 Achieved results

1. **Creation of original models and animations:** All the models and animations used in the game were developed from scratch, contributing to a unique and distinctive gaming experience.

2. **Incorporation of mechanics inspired by popular idle games:** The game incorporates mechanics similar to those found in popular idle games like "Idle Supermarket Tycoon - Shop" [5] and "Idle Burger Empire Tycoon - Game." [4]. These mechanics include features such as queues, parking lots, upgradable components, and the primary objective of earning more money and upgrading various elements.

3. **Enhancement of portfolio through a unique video game:** Having a strong portfolio is crucial for securing employment in the game industry. This project serves as a valuable addition to my portfolio, showcasing my ability to develop a solo game from inception to completion.

4. **Application of knowledge acquired during the degree:** The project allowed for the practical application of the knowledge and skills acquired throughout the degree program.

5. **Creation of a comprehensive and well-structured project report:** A thorough and well-structured project report was produced, effectively explaining the various aspects of the project.

4.2.2 Expected Planning vs Actual Planning

There were some differences between the expected and actual amount of hours spent on certain tasks.

- **Programming:** The expected time for incorporating the mechanics inspired by popular idle games was 120 hours. However, due to unexpected problems and the nature of programming, the actual time spent was 150 hours.

- **Document Realization:** The Game Document Design (GDD) is a crucial aspect of game development, so it is reasonable to expect it to take more time than initially planned. Similarly, this memory report requires comprehensive coverage of various project aspects, resulting in additional hours spent.

- **Modeling:** There was a slight difference in the modeling time, with the expected duration being 70 hours, but the actual time spent was 77 hours. The difference is not significant.

These variations in time allocation highlight the challenges and unpredictability that can arise during a project, particularly in the field of game development. Adaptability and flexibility in adjusting the schedule and resources are essential to successfully navigate such situations.

4.2.3 Milestone Fulfillment

The four milestones mentioned in Table 2.4 have been successfully accomplished, resulting in a successful and satisfying outcome.

The milestones and their corresponding achievements are as follows:

1. **Client & Worker Behavior:** The behavior of clients and workers in the game has been implemented and refined, ensuring realistic and engaging interactions between the game entities and the player.

2. **Zone Behaviors:** The behaviors of different zones within the game have been implemented, providing distinct functionalities and enhancing the gameplay experience for the players.

3. **Assets Modeled and Animated:** All the required assets, including models and animations, have been created and incorporated into the game. These assets contribute to the visual appeal and overall immersion of the gaming environment.

4. **User Interface:** The user interface (UI) of the game has been designed and developed, offering intuitive controls, informative displays, and an enjoyable user experience.

The successful accomplishment of these milestones is a testament to the dedication, effort, and expertise invested in the project. Each milestone represents a significant step towards the completion of the overall game development process, and their fulfillment signifies the successful realization of key objectives.

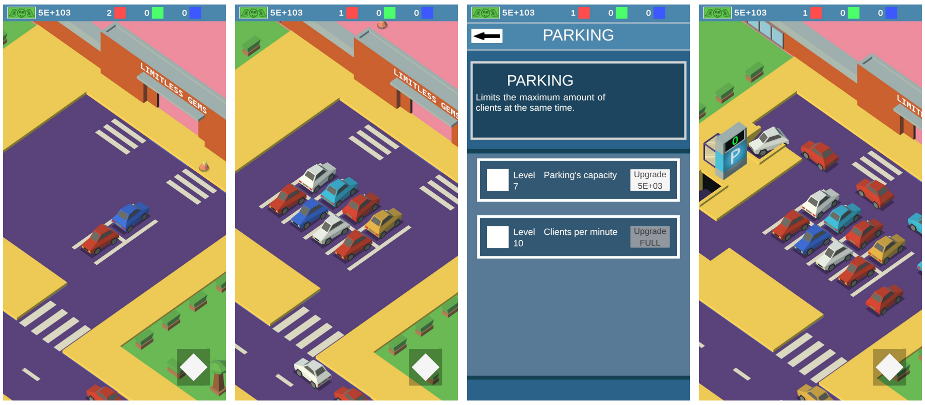
Overall, the achievement of these milestones contributes to a polished and functional game, delivering an engaging and enjoyable experience for the players.

4.2.4 Pictures of the video game

In this section, a series of pictures will be presented that showcase the visual aspects of the final video game. These images offer a glimpse into the game's aesthetics, design, and gameplay elements, providing an overview of the overall experience.

Each picture captures a specific aspect or moment in the game, offering insights into the immersive experience and the attention to detail that has been invested in its creation.

Let's now explore the visual showcase and appreciate the visual journey that awaits players in the game.



PARKING

Limits the maximum amount of clients at the same time.

Level 7 Parking's capacity Upgrade 5E+03

Level 10 Clients per minute Upgrade FULL

Figure 4.32: Parking of Gem Maker Tycoon

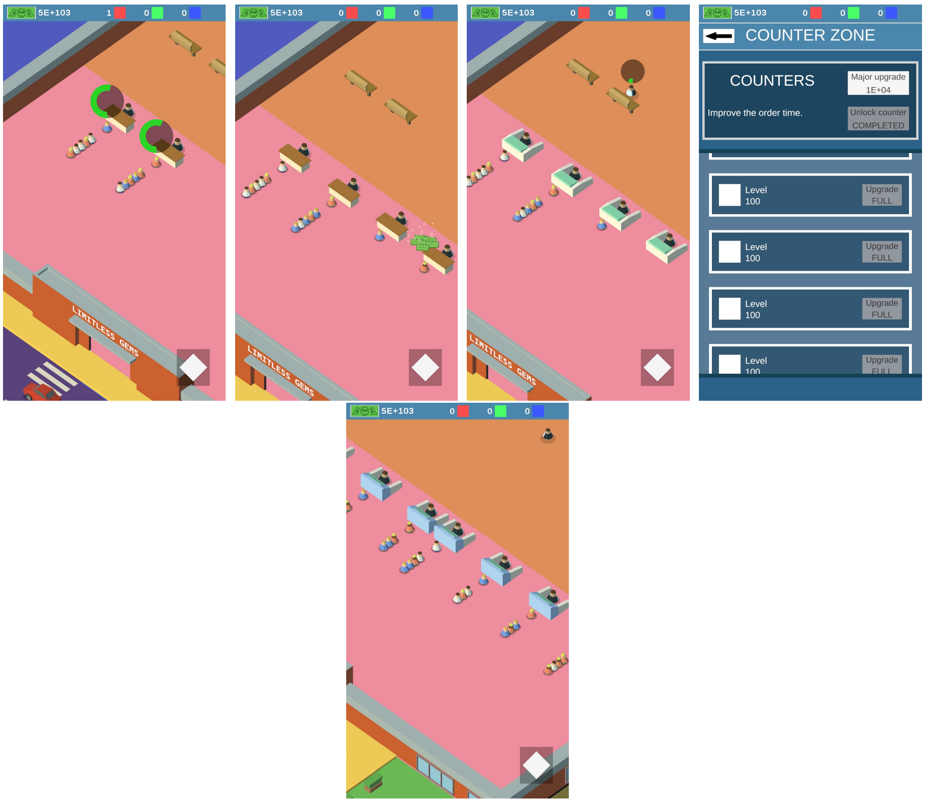


Figure 4.33: Counter Zone of Gem Maker Tycoon

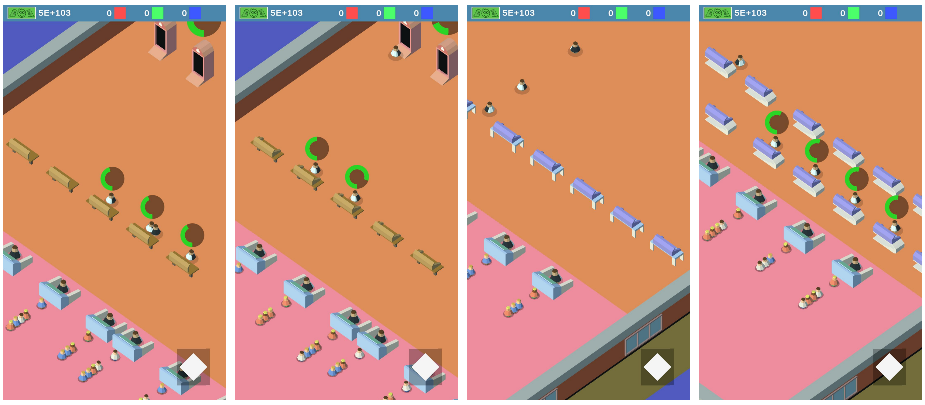


Figure 4.34: Work Zone of Gem Maker Tycoon

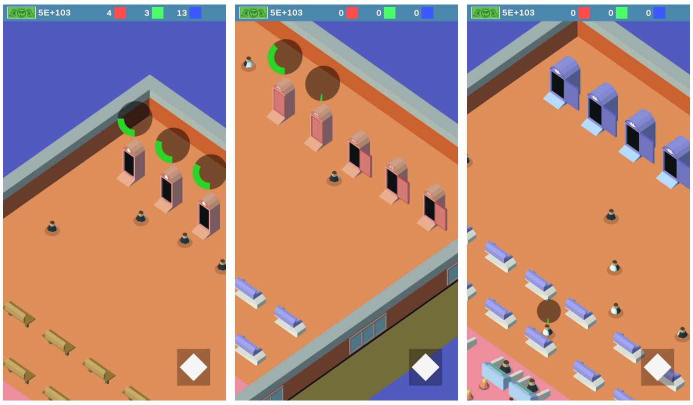


Figure 4.35: Rest Zone of Gem Maker Tycoon

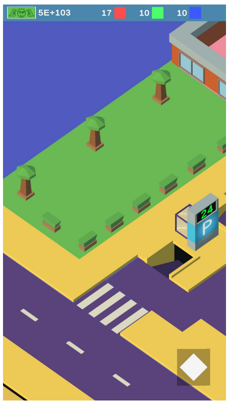
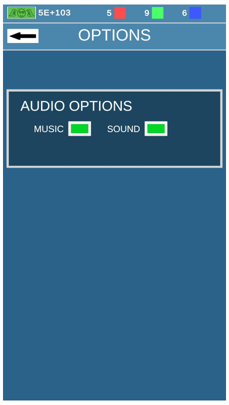


Figure 4.36: Vegetation and Road of Gem Maker Tycoon



OPTIONS

AUDIO OPTIONS

MUSIC 🟩 SOUND 🟩

Figure 4.37: Options of Gem Maker Tycoon

4.2.5 Final Result

The video showcasing the gameplay without the user interface can be found at the following link:

- https://youtu.be/FMh6EINY1pY.

Additionally, you can access a build of the game, with unlimited money, at this location:

- https://drive.google.com/drive/u/1/folders/1rr2N-4dbmpGrCSKGtvGxyy0nNnZH2t1h.

Chapter 5

Conclusions and Future Work

Contents

This section presents the conclusions drawn from the completed work and outlines the aspects that require further consideration for the future development of the project.

5.1 Conclusions

After successfully completing the project and ensuring the fulfillment of all objectives, the final result can be described as excellent. The project has met its intended goals, including the creation of original models and animations, the incorporation of mechanics inspired by popular idle games, and the development of a comprehensive and well-structured project report. The game demonstrates a unique and engaging gaming experience, showcasing the ability to create a solo game from start to finish. Moreover, the practical application of knowledge and skills acquired throughout the degree program is evident in the project's execution.

5.2 Future work

Looking ahead, there are still areas that require attention for the future development of Gem Maker Tycoon. To ensure a successful release, I plan to conduct thorough playtesting and gather feedback from potential players. This will help me refine the gameplay, address any performance issues, and ensure a polished and enjoyable experience. I also aim to expand the game's content by introducing new features, challenges, and levels to keep players engaged and interested.

In conclusion, Gem Maker Tycoon is a significant achievement for me, both personally and professionally. It has strengthened my skills as a game developer and has provided me with a valuable addition to my portfolio. I am excited about the future potential of this project and the opportunities it may bring. With continued dedication and improvement, I believe Gem Maker Tycoon has the potential to be quite a decent game. I look forward to taking the necessary steps to prepare it for publication and sharing it with players around the world.

Bibliography

[1] Andrei Andreev and Timothy Stiles. Breakinfinity's github repositoy. https://github.com/Razenpok/BreakInfinity.cs.

[2] Blender. Blender web page. https://www.blender.org/.

[3] Codigames. Codigames web page. https://codigames.com/.

[4] Codigames. Idle burger empire tycoon—game. https://play.google.com/store/apps/details?id=com.codigames.idle.burger.empire.tycoon.

[5] Codigames. Idle supermarket tycoon — shop. https://play.google.com/store/apps/details?id=com.codigames.market.idle.tycoon.

[6] ComputerHope. What is pinch? https://www.computerhope.com/jargon/p/pinch.htm.

[7] Demigiant. Unity tween engine. http://dotween.demigiant.com/index.php.

[8] Github. Github web page. https://github.com/.

[9] Google. Google drive web page. https://www.google.com/intl/es_es/drive/.

[10] Krita. Krita web page. https://krita.org/es/.

[11] Marcos Martínez. 'idle games': videojuegos en los que se mira la pantalla, y ya. https://www.nobbot.com/pantallas/idle-games-videojuegos-mira-pantalla/. Accessed: 2023-02-25.

[12] Microsoft. Visual studio web page. https://visualstudio.microsoft.com/es/vs/community/.

[13] Crystal Pug. Asset for unity. https://assetstore.unity.com/packages/tools/gui/safe-area-helper-130488.

[14] Dark Fantasy Studio. Dark fantasy studio youtube channel. https://www.youtube.com/@DarkFantasyStudio.

[15] Unity. Cinemachine. https://unity.com/unity/features/editor/art-and-design/cinemachine.

[16] Unity. Cinemachine confiner. https://docs.unity3d.com/Packages/com.unity.cinemachine@2.3/manual/CinemachineConfiner.html.

[17] Unity. Unity asset store. https://assetstore.unity.com/.

[18] Unity. Unity engine. https://unity.com/es.