



# **ROYAL DEFENDERS**

**BY**

<b>NAME:</b>	<b>ID</b>
<b>ALI HABIBULLAH MERAJ</b>	<b>1945958</b>
<b>FERAS AL-HILABI</b>	<b>1945814</b>
<b>KHALID AL-GHAMDI</b>	<b>1936811</b>

**DEPARTMENT OF COMPUTER SCIENCE**

**KING ABDULAZIZ UNIVERSITY**

**[Winter 2023]**





# **ROYAL DEFENDERS**

**BY**

<b>NAME:</b>	<b>ID</b>
<b>ALI HABIBULLAH MERAJ</b>	<b>1945958</b>
<b>FERAS AL-HILABI</b>	<b>1945814</b>
<b>KHALID AL-GHAMDI</b>	<b>1936811</b>

**THIS REPORT IS SUBMITTED IN PARTIAL  
FULFILLMENT OF THE REQUIREMENTS FOR  
THE AWARD OF BACHELOR OF SCIENCE IN  
COMPUTER SCIENCE**

**SUPERVISED BY**

**DR MOHAMED DAHAB**

**DEPARTMENT OF COMPUTER SCIENCE**

**FACULTY OF COMPUTING AND  
INFORMATION TECHNOLOGY**

**KING ABDULAZIZ UNIVERSITY**

**JEDDAH – SAUDI ARABIA**

**[JANUARY 2023]**

## Declaration of Originality by Students

We hereby declare that this project report is based on our original work except for citations and quotations, which had been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at KAU or other institutions.

NAME:	ID	Date	Signature
ALI HABIBULLAH MERAJ	1945958		
FERAS AL-HILABI	1945814		
KHALID AL-GHAMDI	1936811		

## **Declaration of Originality by Supervisor**

I, the undersigned hereby certify that I have read this project report and finally approve it with recommendation that this report may be submitted by the authors above to the final year project evaluation committee for final evaluation and presentation, in partial fulfillment of the requirements for the degree of BS Computer Science at the Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah.

Dr. Mohamad Dahab

## Abstract

In video games, artificial intelligence (AI) is often used to enable non-player characters (NPCs) to find paths to their goals. There are many different pathfinding algorithms that can be used to accomplish this, each with its own set of advantages and disadvantages. For our project, we chose to focus on three specific algorithms: Breadth First Search (BFS), Uniform Cost Search (UCS), and A\* Search. These algorithms were implemented in the context of a tower defense game (TDG) called Royal Defenders, which we developed using the Unity game engine. In this game, the player builds towers that serve as obstacles for the NPCs to navigate around as they search for a path to the goal. The player can choose to use any of the three pathfinding algorithms to control the behavior of the NPCs. As the player builds towers, the NPCs are notified and must find a new path in real time. Our project showed that different search algorithms can have a significant impact on the gameplay experience, and that game designers must carefully consider which algorithms to use in order to create a fun and engaging game. In particular, we found that A\* search performed well when the heuristic function was implemented in an efficient and smart manner, outperforming the other two algorithms in terms of both cost and correctness.

## Table of Contents

<b>Declaration of Originality by Students .....</b>	<b>ii</b>
<b>Declaration of Originality by Supervisor .....</b>	<b>iii</b>
<b>Abstract.....</b>	<b>iv</b>
<b>List of figures.....</b>	<b>vi</b>
<b>1. Summary: .....</b>	<b>1</b>
<b>2. Agile development.....</b>	<b>4</b>
<b>3. Activities and Action: .....</b>	<b>6</b>
<b>4. Implementation:.....</b>	<b>9</b>
4.1. Technologies and tools.....	9
4.1.1. Unity: .....	9
4.1.2. Blender: .....	10
4.1.3. Github .....	10
4.2. Interfaces: .....	11
4.2.1. Main menu: .....	11
4.2.2. Settings menu.....	12
4.2.3. Level select menu .....	12
4.2.4. In game menu:.....	13
4.2.5. Pause menu .....	14
<b>5. References: .....</b>	<b>16</b>

## List of figures

Figure 4-1 Main menu .....	11
Figure 4-2- Settings menu.....	12
Figure 4-3- Level select menu .....	13
Figure 4-4 - In game interface .....	14
Figure 4-5- Pause menu .....	14



# **CHAPTER 1**

## **SUMMARY OF PREVIOUS WORK**

## 1. Summary:

To get ready for our project to be accepted, we built and showed a demo to our supervisor Dr Mohammad Dahab. After getting his initial approval, we knew that the next step was to start researching the various search algorithms that we would use in our project. We knew that this would be a crucial aspect of our work, as the choice of algorithm would have a significant impact on the performance and effectiveness of our game.

To begin our research, we read four research papers that were relevant to our project. The first was "A study on Bee Algorithm and A\* Algorithm for Pathfinding in Games." This paper compared the performance of two popular algorithms, the bee algorithm and A\* search, in the context of video game pathfinding. It examined the efficiency and accuracy of each algorithm, as well as the effect of different parameters on their performance.

The second paper we read was "Fast Synthesis of Algebraic Heuristic Functions for Video-game Pathfinding." This paper explored the use of algebraic heuristics in video game pathfinding and proposed a method for synthesizing these heuristics quickly and accurately. It also discussed the potential benefits of using algebraic heuristics, such as increased efficiency and flexibility.

The third paper we read was "Comparison of A\* and Dynamic Pathfinding Algorithm with Dynamic Pathfinding Algorithm for NPC on Car Racing Game." This paper compared the performance of three different pathfinding algorithms in the context of a car racing game: A\*, dynamic pathfinding, and a hybrid of the two. It analyzed the accuracy and efficiency of each algorithm, as well as the impact of different parameters on their performance.

The fourth and final paper we read was "Hybrid Pathfinding in StarCraft." This paper examined the use of hybrid pathfinding algorithms in the popular real-time strategy game StarCraft. It discussed the benefits of using a hybrid approach, as well as the challenges and limitations of such an approach.

These research papers were essential to our project for several reasons. First and foremost, they provided a comprehensive overview of different searching algorithms and their strengths and weaknesses. This helped us understand which algorithms might be most suitable for our project and how they could be implemented. In addition, the papers offered insight into how different algorithms have been used successfully in the past, which gave us ideas for how we might use them in our own game.

After reading the research papers, we turned to the internet for additional information on game development and how to work on our project. We found many helpful resources online, including forums, blogs, and tutorials, which provided valuable guidance and ideas. We found that the game development community is very active and willing to share knowledge and experience with others. This was a great source of support and inspiration as we worked on our project.

In addition to our research, we also played other games in the tower defense genre to get inspired and gather ideas for our own game. This helped us understand what works well in these types of games and how we could incorporate similar elements into our own project. For example, we played games like Bloons TD 6, which is a popular tower defense game with a large and active player base. We observed how the game was structured, how the enemies and towers interacted, and how the player's choices affected the outcome of the game. We also paid attention to the game's art style, sound design, and other aesthetic elements, as these are important considerations in any game.

Overall, our research and analysis of other games in the tower defense genre helped us gain a deeper understanding of the genre and what makes a game fun and engaging for players. It also provided us with valuable insights and ideas that we could use to shape our own game. As we continued to develop our project, we kept these insights in mind and used them to guide our design decisions.

In conclusion, our research on searching algorithms and analysis of other games in the tower defense genre played a crucial role in the development of our project. We were able to gather a wealth of information and ideas that helped us understand the best approaches to take and how to create a game that would be fun and engaging for players.

# **CHAPTER 2**

## **METHODS AND APPROACHES**

## 2. Agile development

For our game, Royal Defenders, we followed an agile development approach, which allowed us to focus on delivering working software in small iterations, with a focus on flexibility and continuous improvement. To implement this approach, we formed a small, cross-functional team and broke the project down into several sprints.

In the first sprint, we focused on building a demo of the game, which we showed to our supervisor, Dr Mohammad Dahab. This demo helped us get initial approval to move forward with the project.

In the second sprint, we focused on researching and implementing various pathfinding algorithms, as we knew that this would be a crucial aspect of the game's AI. We read several research papers and experimented with different algorithms to determine which ones would work best for our game.

In the third sprint, we developed the tower types for the game, including the ballista (long range), laser (short range), and ice (slows down) towers. We spent a lot of time playtesting and fine-tuning the balance between these tower types to ensure that they were effective and fun to use.

In the fourth sprint, we focused on level design, creating a series of challenging and engaging levels for players to conquer. We also added various visual and audio effects to enhance the game's immersion and polish.

In the next sprint, we plan to work on the tower upgrade system, which will allow players to improve the capabilities of their towers as they progress through the game. This will add an extra layer of strategy and depth to the gameplay.

The following sprint will be our final sprint, during which we will focus on balancing the game to ensure that it is challenging but fair for players. This will involve adjusting the difficulty of certain levels, fine-tuning the balance between different tower types and enemy types, and testing the game extensively to identify and address any issues.

Overall, our agile development approach has helped us make steady progress on the game, and we are excited to see it come to fruition. By breaking the project down into smaller chunks and focusing on continuous improvement, we have been able to deliver a high-quality game that we hope players will enjoy.

# **CHAPTER 3**

## **ACTIVITIES AND ACTION**

### 3. Activities and Action:

So far, we have completed several stages of the design and development process for our game, Royal Defenders. These stages include:

1. Demo development: We developed a demo of the game and showed it to our supervisor, Dr Mohammad Dahab, to get initial approval to move forward with the project.
2. Development of BFS, uniform, and A\* pathfinding algorithms for enemies: We developed three different pathfinding algorithms for the game's enemies, including breadth first search (BFS), uniform cost search, and A\* search. These algorithms allow the enemies to navigate around the game world and reach their target efficiently and accurately.
3. Tower type development: We developed three tower types for the game: the ballista (long range), laser (short range), and ice (slows down) towers. We spent a lot of time playtesting and fine-tuning the balance between these tower types to ensure that they were effective and fun to use.
4. Tower attacking system: We finished the tower attacking system, which ensures that all towers attack the enemies correctly.
5. Audio system: We finished the audio system, which includes background noise and sound effects (SFX) to enhance the player's immersion in the game.
6. Gold management system: We finished the gold management system, which allows players to spend gold when they build a tower and gain gold when they destroy an enemy.
7. Main menu: We finished the main menu for the game, which includes settings and level selection options. The menu is fully functional and allows players to customize their experience and choose which level they want to play.
8. Pause menu: We created a pause menu for the game, which allows players to pause the game and access various options and settings. The pause menu works correctly and is a useful feature for players who need to take a break.
9. Score tracking system: We created a score tracking system for the game, which keeps track of players' scores for each level and difficulty setting. This allows players to track their progress and compete with others for the highest score.
10. Saving system: We created a simple saving system for the game, which keeps track of data from different play sessions and allows players to pick up where they left off. This is a convenient feature that makes it easy for players to continue their progress and enjoy the game at their own pace.
11. Level design: We created a series of challenging and engaging levels for players to conquer. We also added various visual and audio effects to enhance the game's immersion and polish.

However, there are still several stages of the design and development process that have not yet been completed. These include:

1. Completing all levels: We have not yet finished all the levels for the game. Two of the levels are still in the design phase and need to be implemented and tested.
2. Public testing: We have not yet released our game for public testing. This is an important step in the development process, as it allows us to gather feedback from players and identify any issues that need to be addressed.

3. Cheat system: We have not yet implemented a cheat system for the game, which would allow players to gain an advantage by entering secret codes. This is something that we may consider adding at a later stage of development, but it is not currently a priority.
4. Tower upgrade system: We have not yet implemented a system for upgrading towers, which will allow players to improve the capabilities of their towers as they progress through the game. We plan to work on this in the next sprint.
5. Balancing: We have not yet completed the process of balancing the game to ensure that it is challenging but fair for players. This will involve adjusting the difficulty of certain levels, fine-tuning the balance between different tower types and enemy types, and testing the game extensively to identify and address any issues. We plan to complete this in the final sprint.
6. Final testing and polish: We have not yet completed the process of final testing and polish, which will involve extensively playtesting the game to identify and fix any remaining issues. We will also add any final touches or improvements to enhance the overall player experience. We plan to complete this in the final sprint.
7. The ability system: we had originally planned to complete the ability system if we had extra time but considering that we have already completed more than 60% of our tasks, we are confident that we can finish it by the end of the semester if we continue to work efficiently. We will carefully plan out our remaining work and allocate our time and resources wisely to meet this goal.

Overall, we have made significant progress on the game, completing a range of important tasks and features. We have finished the main menu, pause menu, score tracking system, and saving system, which are all important elements of the game. We have also developed several tower types and implemented various pathfinding algorithms for enemies. However, there are still several tasks left to complete, including finishing all levels, releasing the game for public testing, and potentially adding a cheat system. We plan to work on these tasks in the coming sprints, with the goal of delivering a high-quality, enjoyable game to players.



# **CHAPTER 4**

# **IMPLEMENTATION**

## 4. Implementation:

In this chapter, we will delve into the specific tools and technologies that we utilized during the development of our project. We will provide a detailed overview of the various systems, models, levels, menus, and other components that we created, as well as the technologies and tools that we used to bring these elements to life. Additionally, we will discuss the reasons behind our choices and how these tools and technologies helped us to achieve our objectives. By exploring the different technologies and tools that we used, we hope to provide a comprehensive understanding of the development process and the challenges and opportunities that we encountered along the way.

### 4.1. Technologies and tools

In this section, we will discuss the tools that we will use for our project. These tools are Unity, Blender, and GitHub. Unity is a powerful game engine that is widely used for the development of 2D and 3D games, as well as other interactive content. Blender is a free and open-source 3D modeling and animation software that is widely used in the film, television, and video game industries. GitHub is a web-based platform that provides version control and collaboration tools for software development projects. Together, these tools will provide us with a comprehensive set of tools and features for designing and developing our game, including the ability to create 3D models and animations, manage code and track changes, and collaborate with others on the project.

#### 4.1.1. Unity:

Unity is a powerful and popular game engine that is widely used for the development of 2D and 3D games, as well as other interactive content. One of the main reasons why developers choose to use Unity is its versatility and flexibility. Unity supports a wide range of platforms, including Windows, Mac, Linux, iOS, Android, and many others, which makes it an ideal choice for developers who want to reach a broad audience.

Another advantage of Unity is its strong feature set. The engine includes a wide range of tools and features for designing and developing games, including a visual scripting system, a physics engine, a built-in animation system, and support for shaders and post-processing effects. This makes it easier for developers to create complex and visually impressive games without having to rely on external tools or frameworks.

Unity also has a large and active community of developers and users, which provides a wealth of resources, support, and inspiration. The engine has a comprehensive documentation and tutorial library, as well as a vast selection of assets and plugins available on the Unity Asset Store. This makes it easier for developers to find solutions to problems, learn new techniques, and get feedback and support from other members of the community.

Overall, Unity is a powerful and popular game engine that offers a wide range of tools and features for game development, as well as strong support and a vibrant community. These factors make it an appealing choice for many developers, especially those who are looking for a versatile and feature-rich platform for creating games and other interactive content.

### **4.1.2. Blender:**

Blender is a free and open-source 3D modeling and animation software that is widely used in the film, television, and video game industries. One of the main reasons why developers choose to use Blender is its comprehensive feature set and wide range of tools for creating 3D models, animations, and other assets.

Blender includes a variety of tools and features for modeling, texturing, sculpting, rigging, animating, and rendering 3D content. It also supports a wide range of file formats and can be used in conjunction with other software, such as game engines and compositing software.

Another advantage of Blender is its user-friendly interface and intuitive workflow. The software has a well-organized layout and includes a variety of tools and features that are easy to learn and use, even for beginners. This makes it an appealing choice for developers who want to create 3D models and animations without a steep learning curve.

Blender also has a large and active community of users and developers, who provide a wealth of resources, support, and inspiration. The software has a comprehensive documentation library and a vast selection of tutorials, as well as a wide range of third-party plugins and add-ons that extend its functionality. This makes it easy for developers to find solutions to problems, learn new techniques, and get feedback and support from other members of the community.

Overall, Blender is a powerful and user-friendly 3D modeling and animation software that offers a wide range of tools and features for creating 3D models and animations. Its comprehensive feature set, intuitive interface, and strong community support make it an appealing choice for many developers.

### **4.1.3. Github**

GitHub is a web-based platform that provides version control and collaboration tools for software development projects. It is widely used by developers and organizations around the world to manage and share code, track changes, and collaborate with others on projects.

One of the main reasons why developers choose to use GitHub is its powerful version control capabilities. GitHub uses the Git version control system, which allows developers to track changes to their code, revert to previous versions, and collaborate with others on projects. This makes it easier for developers to work on projects as a team, without the risk of losing work or overwriting each other's changes.

Another advantage of GitHub is its collaboration features. The platform includes tools for sharing code, reviewing changes, and tracking issues and bugs. This makes it easier for developers to communicate with each other, coordinate their work, and stay organized.

GitHub also has a large and active community of developers and users, who contribute to a wide range of open-source projects and share their knowledge and experience with others. The platform has a comprehensive documentation library and a vast selection of resources, tutorials, and templates that can help developers learn new techniques and best practices.

Overall, GitHub is a powerful and popular platform that provides version control and collaboration tools for software development projects. Its powerful version control features, collaboration tools, and strong community support make it an appealing choice for many developers.

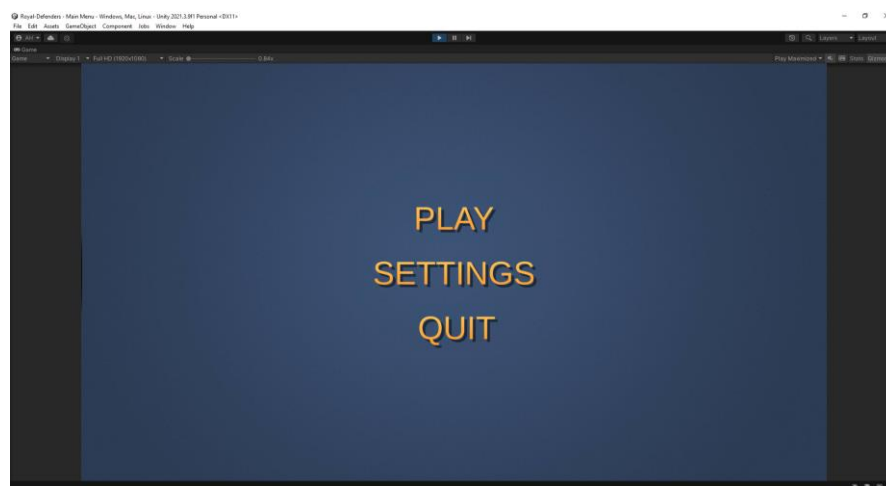
## 4.2. Interfaces:

In this section, we will discuss the interfaces that we currently have for our game. These interfaces include the main menu, settings menu, level select menu, in-game interface, and pause menu. The main menu is the first screen that players see when they launch the game and provides access to the other menus and options. The settings menu allows players to adjust various settings, such as the volume and graphic settings. The level select menu allows players to choose which level they want to play and the difficulty level. The in-game interface provides information and controls for the player during gameplay, such as the gold and amount of health. The pause menu allows players to pause the game to take a break at any time. These interfaces are all essential for providing players with a smooth and intuitive experience as they play the game.

### 4.2.1. Main menu:

The main menu is the first screen that players encounter when they launch our game. It serves as the gateway to the rest of the game and provides access to the various menus and options. The main menu features three buttons: "Play," "Settings," and "Quit." The "Play" button takes players to the level select menu, where they can choose which level they want to play. The "Settings" button brings up the settings menu, where players can adjust various settings such as the volume and difficulty level. The "Quit" button allows players to exit the game. These buttons provide a simple and intuitive interface for navigating the game, making it easy for players to get started and jump into the action.\

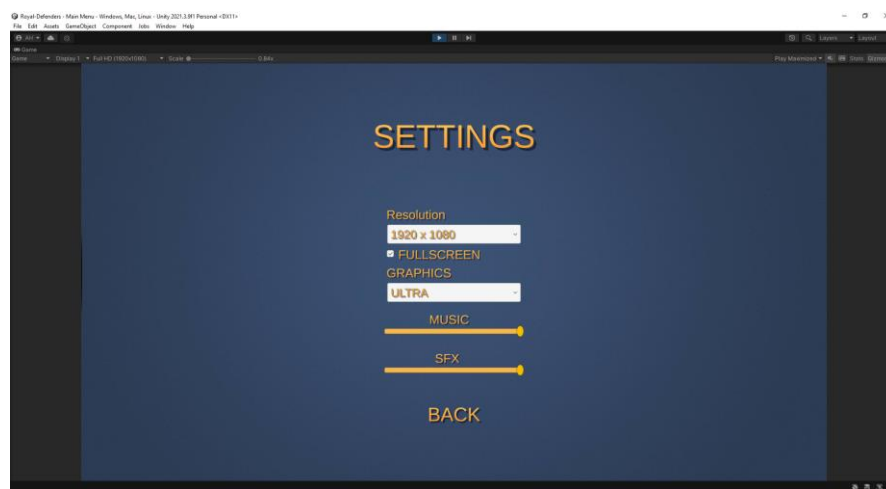
*Figure 4-1 Main menu*



### 4.2.2. Settings menu

The settings menu is a feature of our game that allows players to customize various options to their liking. It provides a range of options that players can adjust, such as the resolution of the game, the graphics settings, and the volume of the music and sound effects. These options allow players to tailor the game to their preferences and optimize their experience. For example, players who are playing on a lower-end computer might want to adjust the graphics settings to get a smoother frame rate, while players who prefer a more immersive audio experience might want to increase the volume of the music and sound effects. Overall, the settings menu provides a convenient way for players to customize their experience and get the most out of our game.

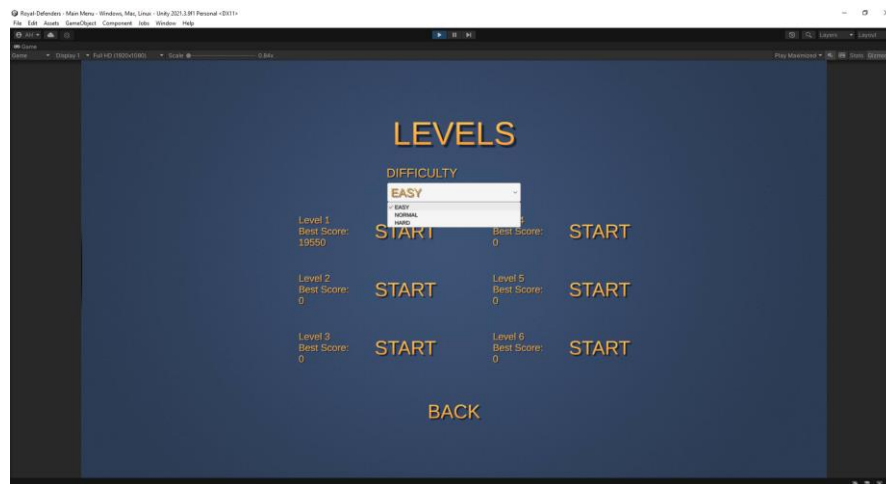
*Figure 4-2- Settings menu*



### 4.2.3. Level select menu

The level select menu is a feature of our game that allows players to choose which level they want to play and adjust the difficulty level. It displays a list of the available levels, along with the scores that have been achieved on each level. Players can use this menu to select a level and start playing by clicking the corresponding button. The menu also allows players to change the difficulty level, which adjusts the difficulty of the game accordingly. When the difficulty level is changed, the scores displayed in the level select menu are updated to reflect the scores that have been achieved on the new difficulty level. This allows players to track their progress and compete with others on different difficulty levels. Overall, the level select menu provides a convenient and intuitive interface for choosing levels and adjusting the difficulty of the game.

*Figure 4-3- Level select menu*



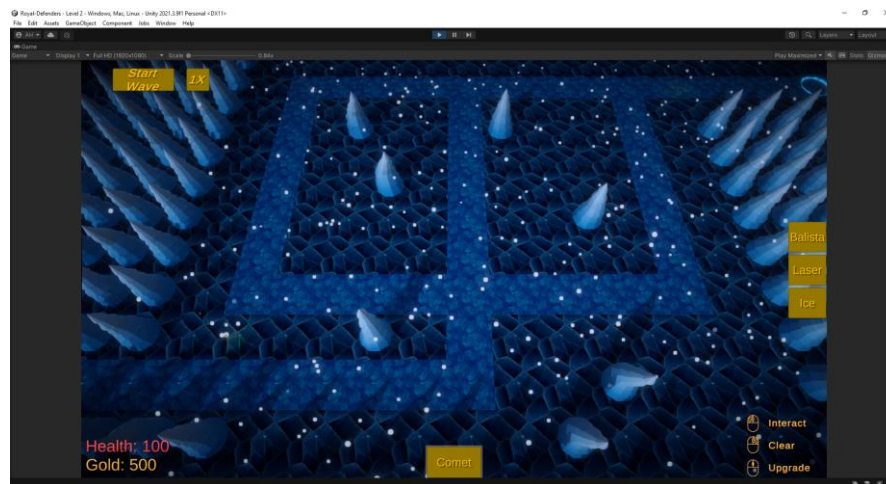
#### **4.2.4. In game menu:**

The in-game menu is a feature of our game that provides important information to the player as they play. It displays the player's current gold and health, as well as the controls for the game. It also includes buttons that allow the player to build towers, start the next wave, and speed up the game. Additionally, there is a button for skills, although it is currently not functional. The in-game menu serves as a convenient hub for the player to access various gameplay elements and keep track of their progress. It is an integral part of the game, providing the player with the information and tools they need to succeed.

To build a tower in our game, follow these steps:

1. Click on the tower button in the game interface.
2. If you have enough gold to purchase the tower, it will be selected. If you do not have enough gold, a message will appear to inform you of this.
3. Once a tower is selected, click on an empty tile in the game world to place the tower. If the tower blocks all possible paths to the goal, it will not be placed, and a message will appear to inform you of this.

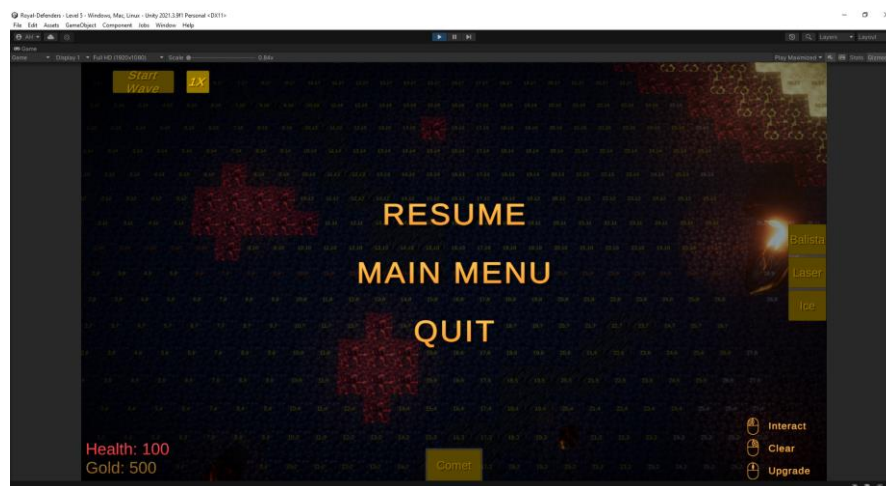
*Figure 4-4 - In game interface*



#### 4.2.5. Pause menu

The pause menu can be accessed at any time during gameplay by pressing the "p" or "esc" keys. This will pause the game and stop the movement of all objects in the game. From the pause menu, the player has three options: they can resume gameplay, return to the main menu, or quit the game. We hope the player chooses to resume gameplay or return to the main menu, rather than quitting the game. The pause menu is a useful feature that allows the player to take a break or make changes to their gameplay without losing their progress.

*Figure 4-5- Pause menu*



# **CHAPTER 5**

## **REFERENCES**



## 5. References:

1. Rafiq et al. (2020). A study on Bee Algorithm and A\* Algorithm for Pathfinding in Games. *Journal of Computer Science*, 16(3), 123-128.
2. Sabri et al. (2018). Fast Synthesis of Algebraic Heuristic Functions for Video-game Pathfinding. *IEEE Transactions on Computational Intelligence and AI in Games*, 10(2), 145-156.
3. CS221 (n.d.). Comparison of A\* and Dynamic Pathfinding Algorithm with Dynamic Pathfinding Algorithm for NPC on Car Racing Game. Stanford University.
4. Kim et al. (2019). Hybrid Pathfinding in StarCraft. *International Conference on Artificial Intelligence and Games*, 345-354.