SCHOOL OF ENGINEERING AND TECHNOLOGY

FINAL ASSESSMENT FOR THE BSC (HONS) INFORMATION TECHNOLOGY; BSC (HONS) COMPUTER SCIENCE; BACHELOR of SOFTWARE ENGINEERING (HONS)YEAR 2

ACADEMIC SESSION 2023; SEMESTER 3

PRG2104: OBJECT ORIENTED PROGRAMMING

Project **DEADLINE: Week 14**

INSTRUCTIONS TO CANDIDATES

- This assignment will contribute 50% to your final grade.
- This is an individual assignment.

IMPORTANT

The University requires students to adhere to submission deadlines for any form of assessment. Penalties are applied in relation to unauthorized late submission of work.

- Coursework submitted after the deadline will be awarded 0 marks
-

Lecturer's Remark (Use additional sheet if required)

I.....Chin Zi Ming........................ (Name) ....21062120..............std. ID received the assignment and read the comments......(*chin* / 2/8/2023)................................ (Signature/date)

Academic Honesty Acknowledgement

"I ....Chin Zi Ming....................................(student name). verify that this paper contains entirely my own work. I have not consulted with any outside person or materials other than what was specified (an interviewee, for example) in the assignment or the syllabus requirements. Further, I have not copied or inadvertently copied ideas, sentences, or paragraphs from another student. I realize the penalties *(refer to page 16, 5.5, Appendix 2, page 44 of the student handbook diploma and undergraduate programme)* for any kind of copying or collaboration on any assignment."

…......... *(chin* / 2/8/2023).......................... (Student's signature / Date)

Table of Contents

1.0 INTRODUCTION

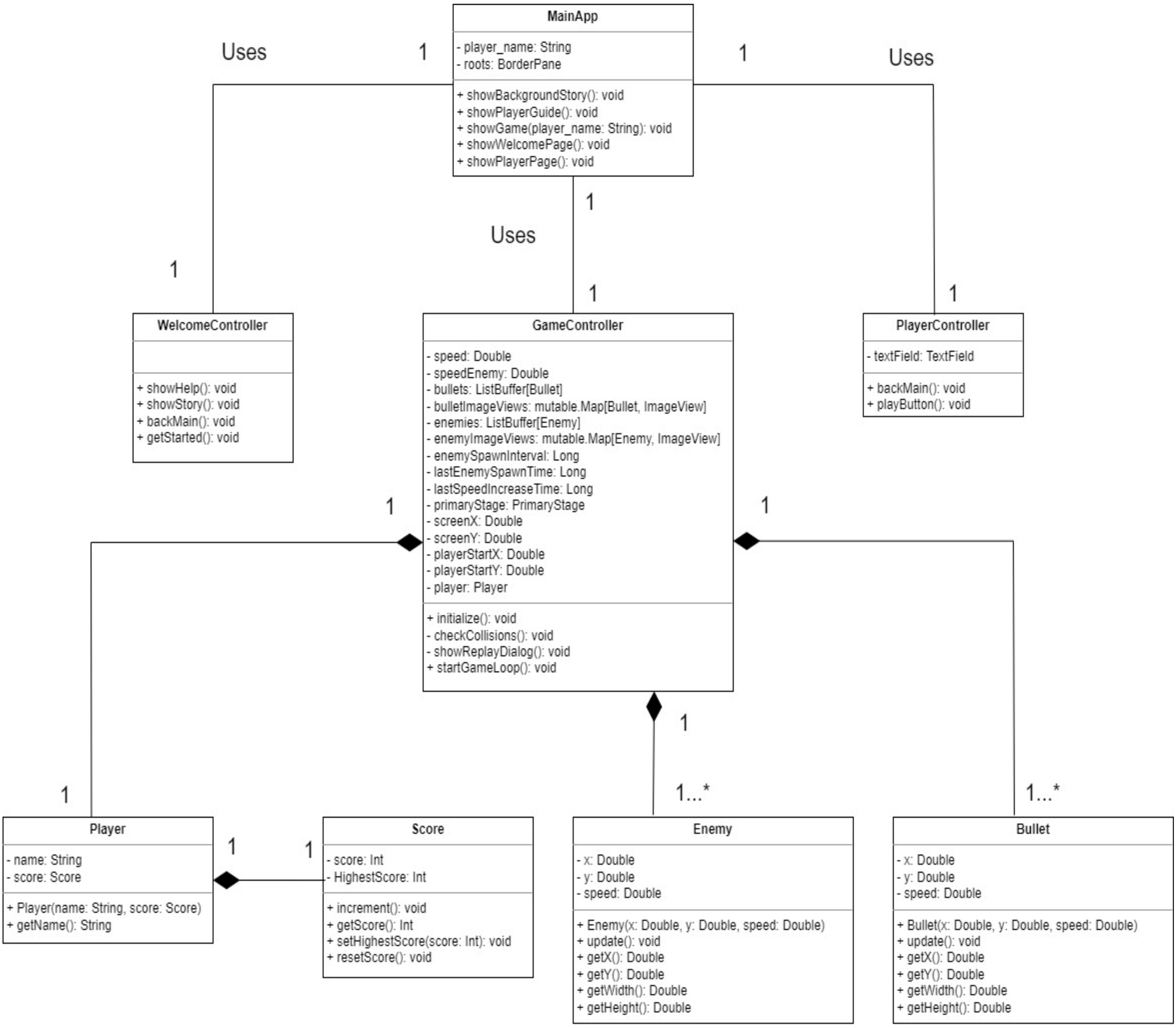
Throughout this thrilling assignment, I was immersed in the fascinating world of software application development using object-oriented programming (OOP). My major goal was to fulfill specified learning outcomes outlined in our subject's syllabus, with a special emphasis on CLO 4. The journey allowed me to investigate and employ third-party object-oriented libraries, with the scalaFX GUI library taking center stage for creating a compelling graphical user interface.

The task was to create a stand-alone GUI system with at least four functionalities while leveraging the power of object-oriented programming techniques. I carefully defined the necessary classes and developed relevant relationships among them to provide a well-structured framework.

My idea was centered on the popular arcade game "Space Invaders," in which players control a spaceship and engage in a heated combat against invading aliens. From handling player input to arranging spacecraft movement, shooting mechanics, opponent spawning, and collision detection, every component of the game mechanics was methodically created. The goal was to create a user-friendly interface that would improve the entire gaming experience.

To summarize, this project provided an exceptional opportunity to obtain hands-on expertise in software development, GUI design, and the actual implementation of object-oriented concepts. The trip was both tough and gratifying, as it allowed me to use my creativity and problem-solving talents to produce an immersive and engaging gaming experience.

2.0 UML CLASS DIAGRAM



3.0 PROGRAM FEATURES

In this section, the various features featured in the proposed game are discussed and expounded upon in further detail in order to correctly convey their functionality and also to demonstrate that these features or functions work as planned without any complications. As a disclaimer, not all game features can be captured as an image for demonstration or presentation purposes due to the way they were developed, as well as being difficult concepts to capture in an image, such as movements, appearance and disappearance, collision, smooth animation, and game loop management.

3.1 Background Story

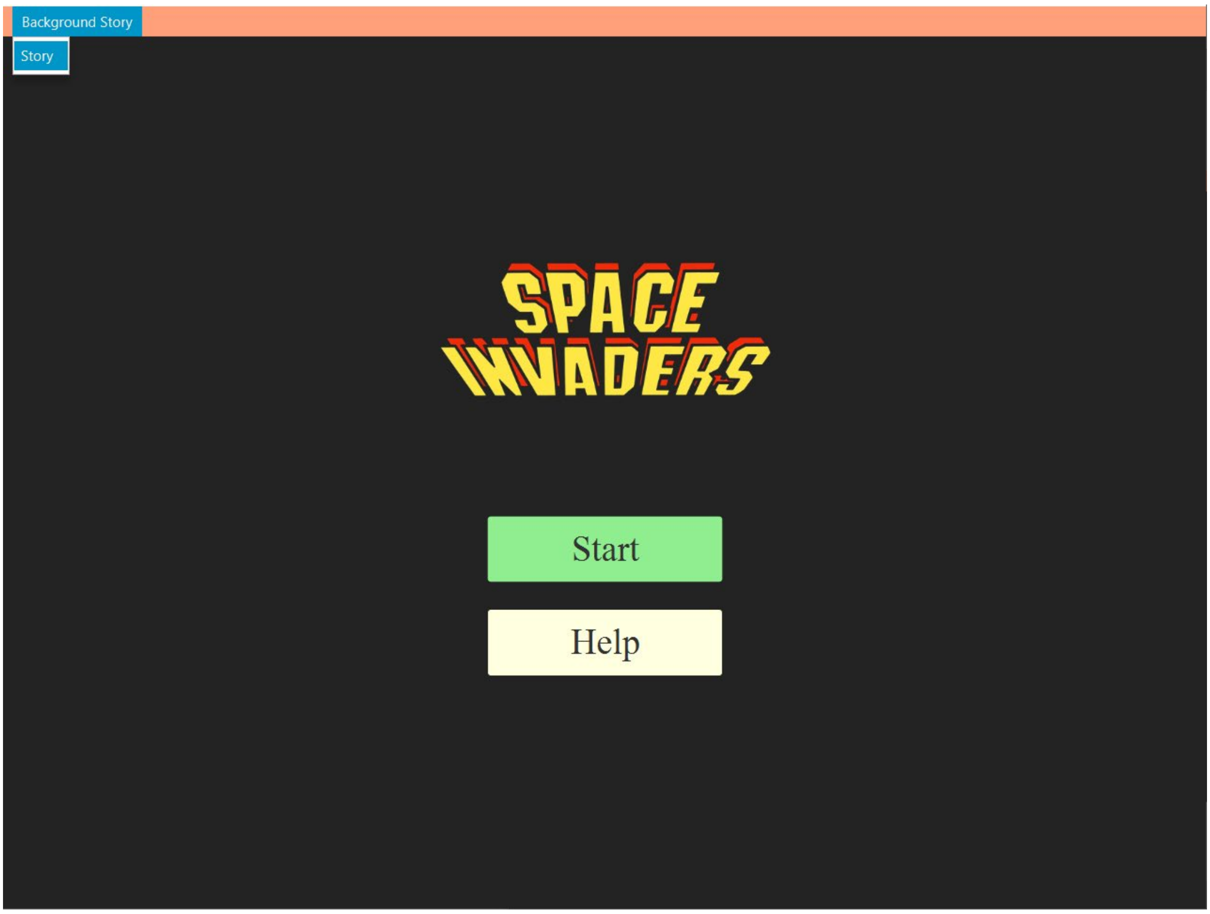
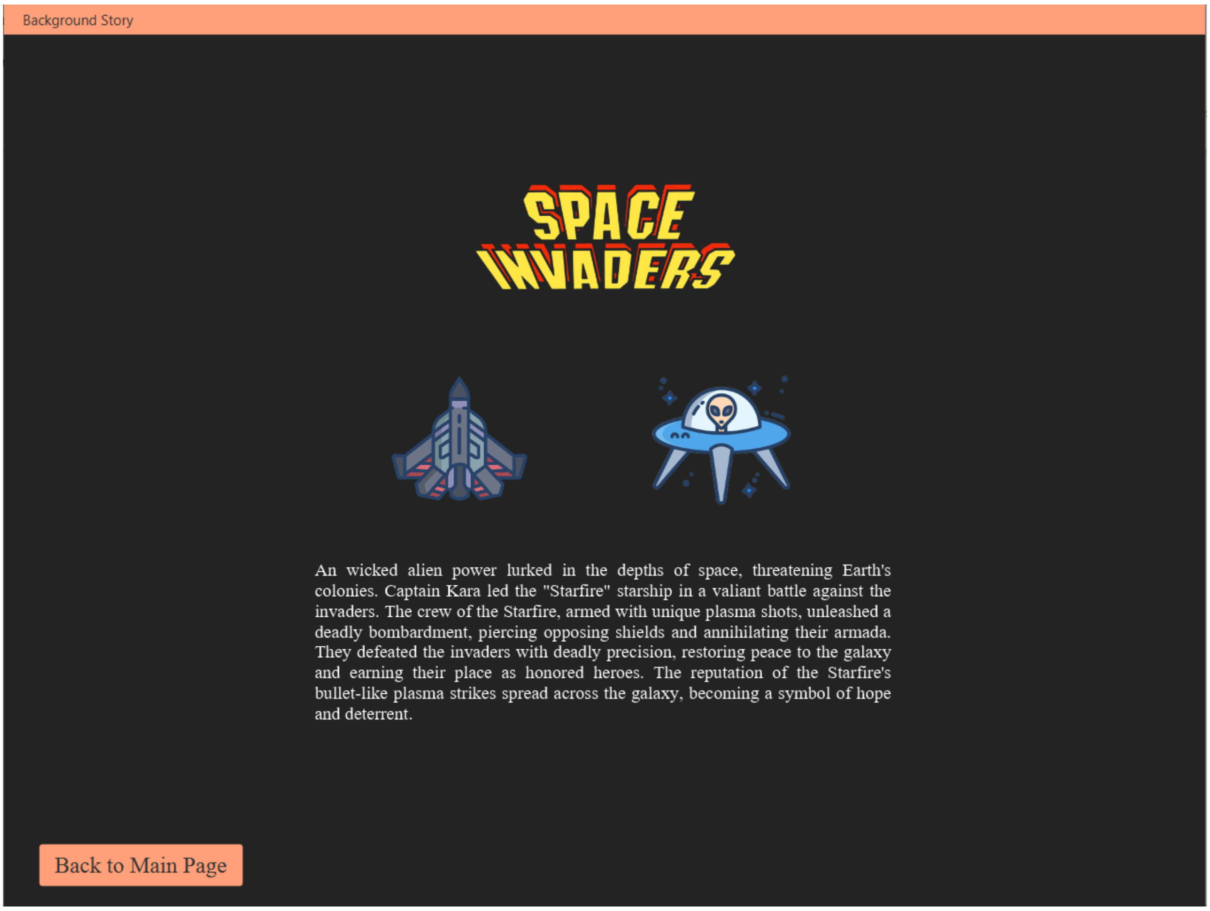


Figure 1



An wicked alien power lurked in the depths of space, threatening Earth's colonies. Captain Kara led the "Starfire" starship in a valiant battle against the invaders. The crew of the Starfire, armed with unique plasma shots, unleashed a deadly bombardment, piercing opposing shields and annihilating their armada. They defeated the invaders with deadly precision, restoring peace to the galaxy and earning their place as honored heroes. The reputation of the Starfire's bullet-like plasma strikes spread across the galaxy, becoming a symbol of hope and deterrent.

Figure 2

As you can see in figure 1, this is the starting page of the space invaders game. You can see there’s a Background Story in the top left menu bar. With that when you clicked on the Background Story and choose Story, you can see the background story of this game shown in figure 2.

3.2 Player Guide

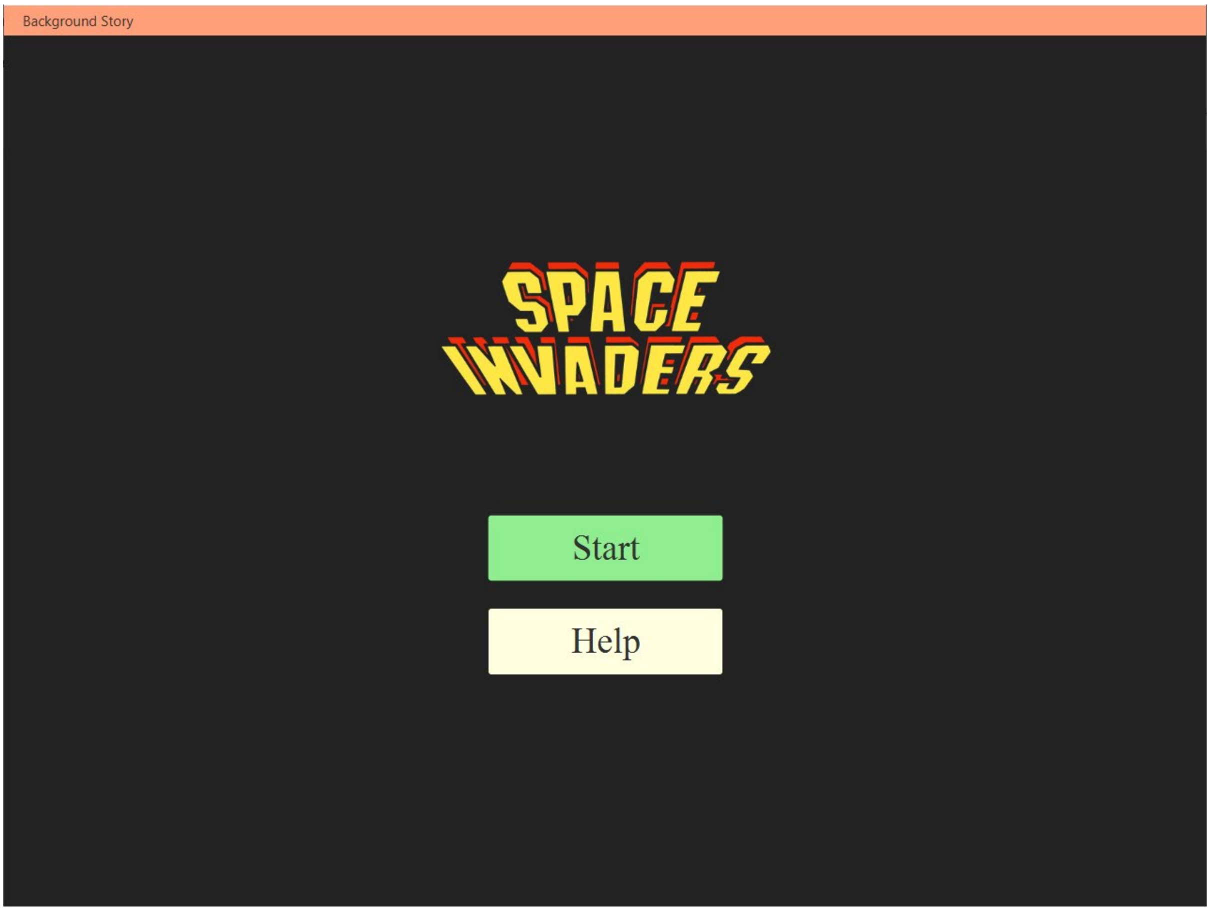
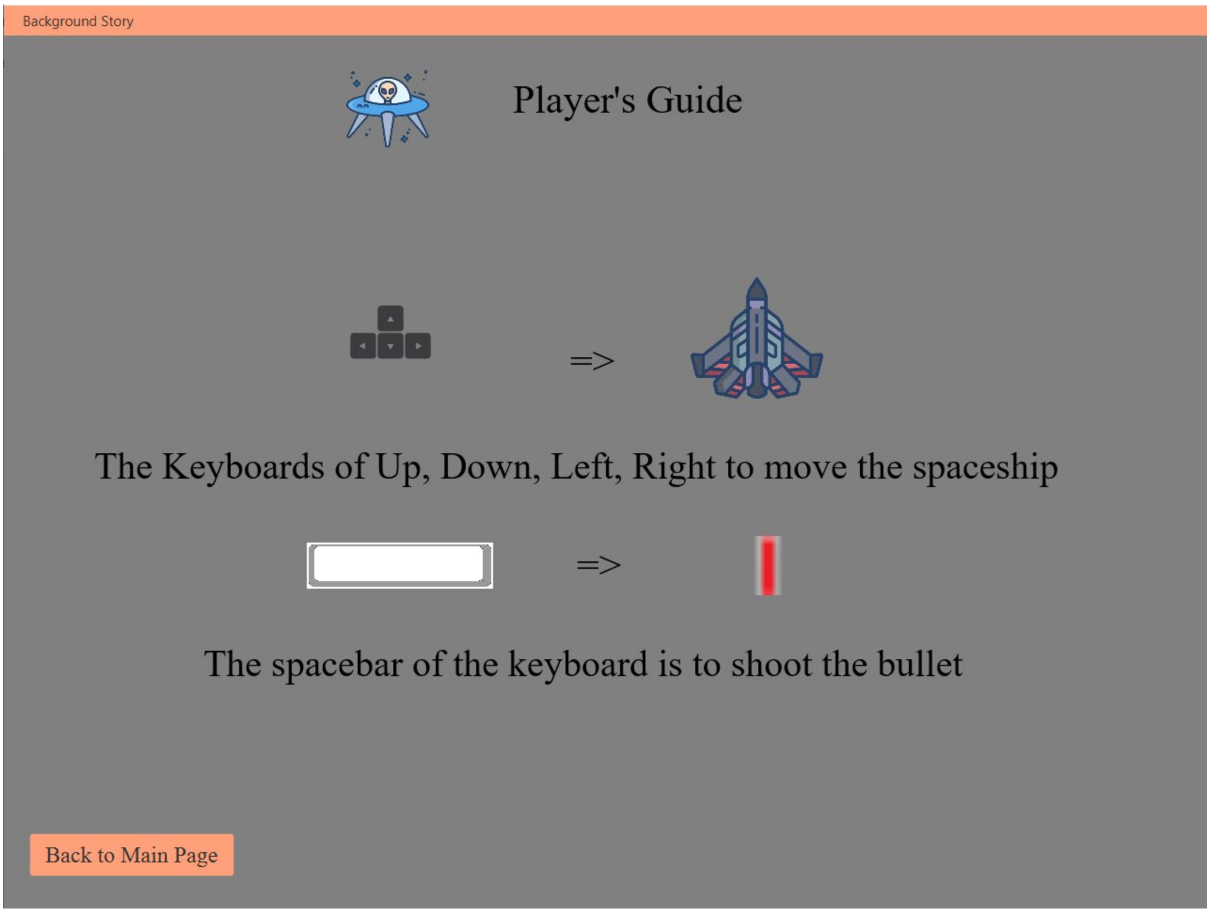


Figure 3



Player's Guide

The Keyboards of Up, Down, Left, Right to move the spaceship

The spacebar of the keyboard is to shoot the bullet

Figure 4

As you can see in this starting page shown in figure 3, the Help button is where you can see the player guide before starting the game. In figure 4, it shows the player guide details to show the player about the game controls.

3.3 Player Input and Movement

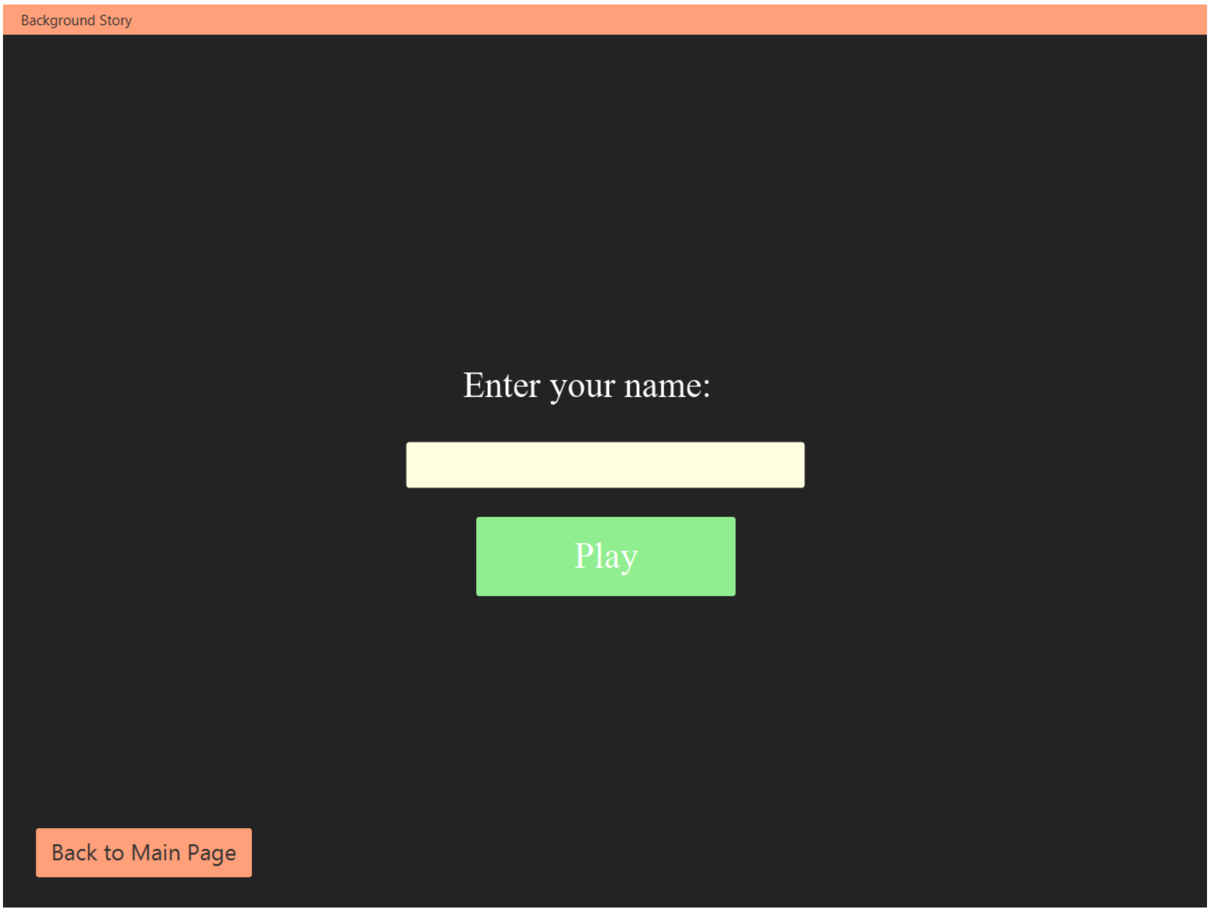
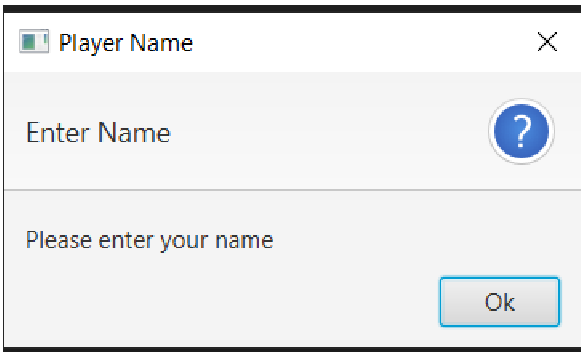


Figure 5



Please enter your name

Figure 6

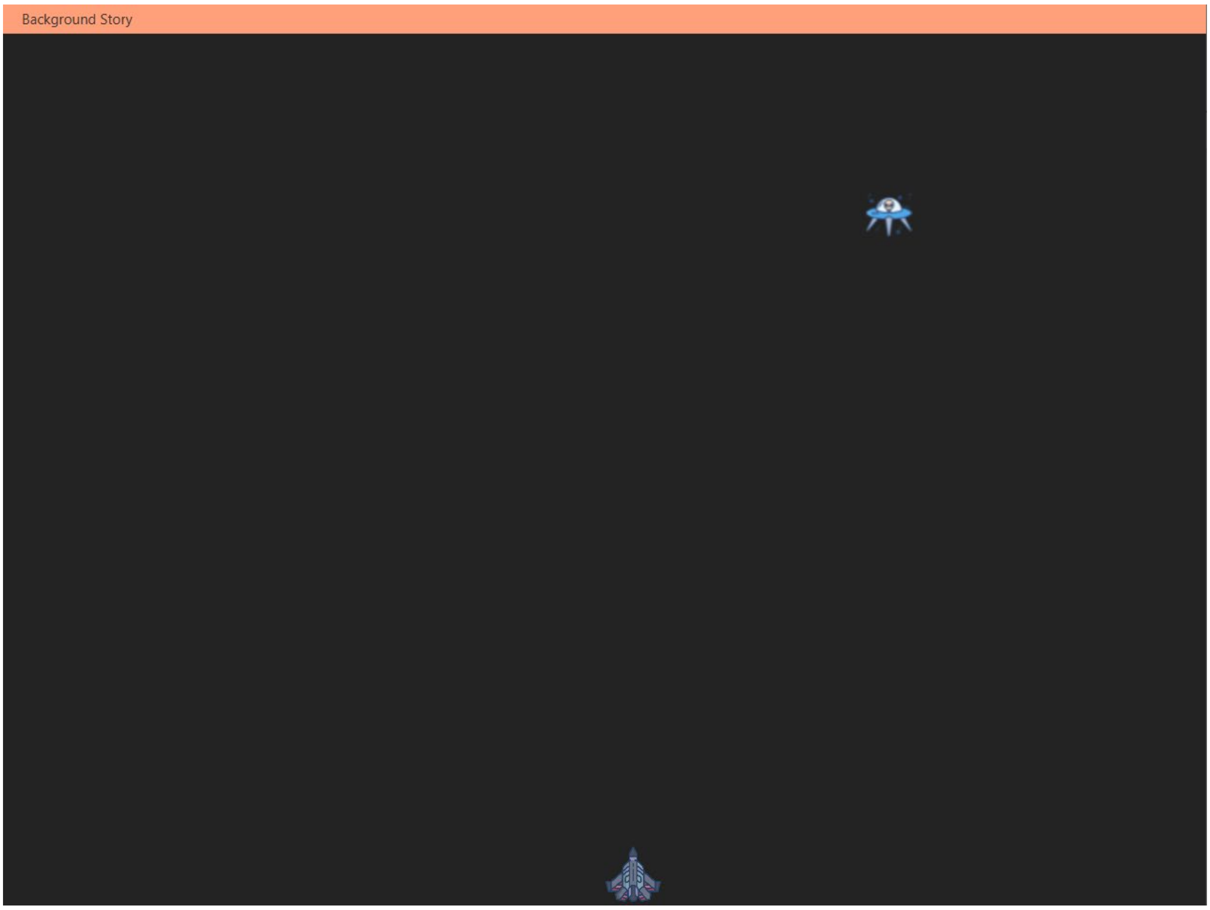


Figure 7

After the player clicked on START button in main page, they will be led to next page shown in figure 5 for them to input their name. The name entry column should not be leave empty, if not it will show alert shown in figure 6. After the player enter the name and clicked on Play button, he or she will be prompted to the start of the game. Furthermore, the player can move the spaceship up, down, left, right by using the arrow keys in the keyboard within the game screen. To add on, the spaceship will stay within the screen and will not moving out of the screen of any directions when pressing the arrow keys.

3.4 Shooting Mechanism

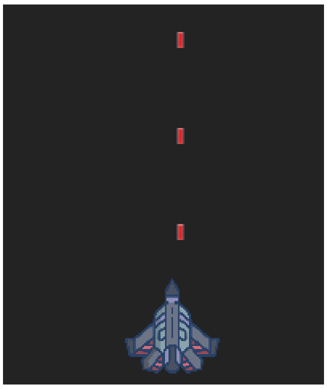


Figure 8

In the figure 8 shown, the player can shoot bullets from the spaceship by pressing the spacebar to shoot the invaders.

3.5 Enemy Spawning

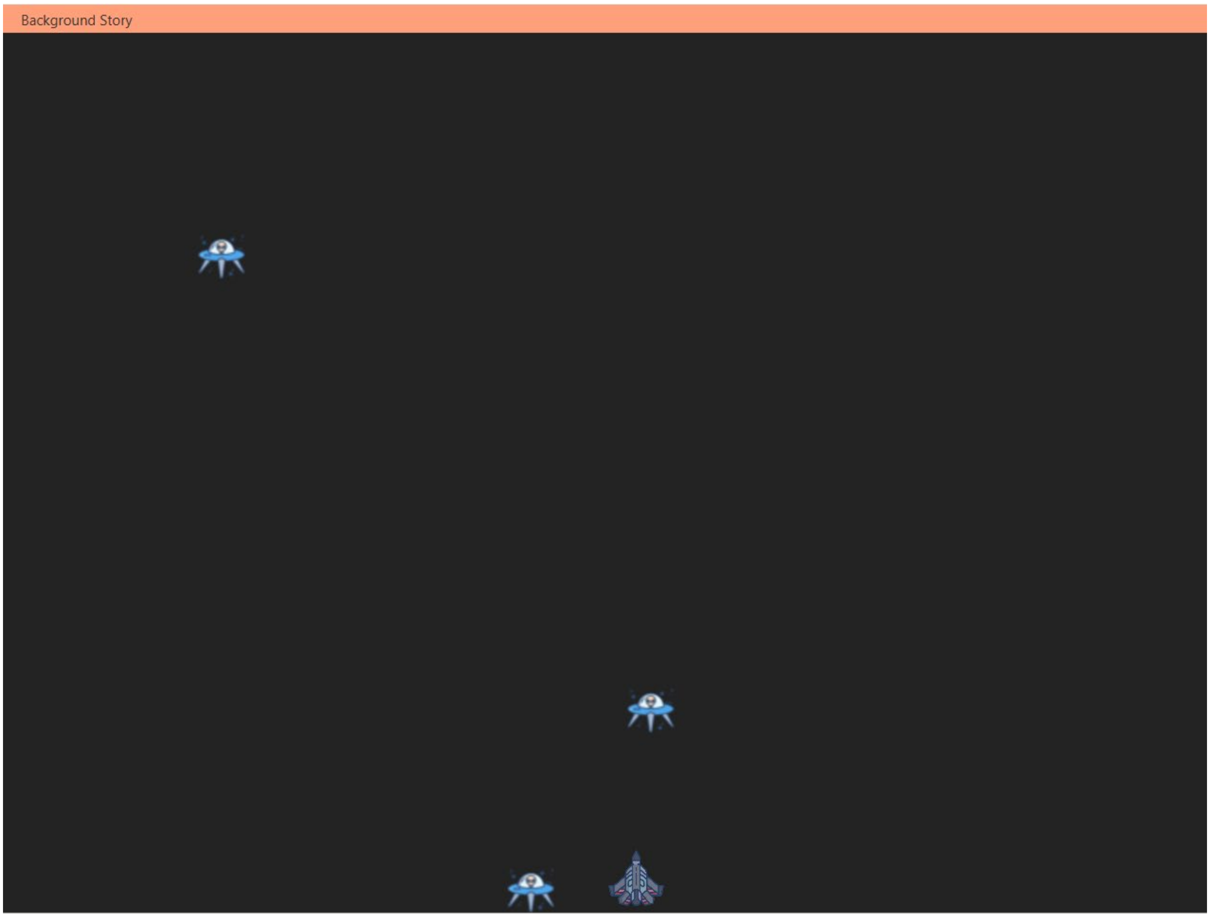


Figure 9

As you can see in figure 9, enemies (aliens) will spawn from the top of the game screen and descend downhill at regular intervals. After certain intervals, the enemies will move down faster for the spaceship to defend and hence increase the difficulty of the game.

3.6 Collision Detection

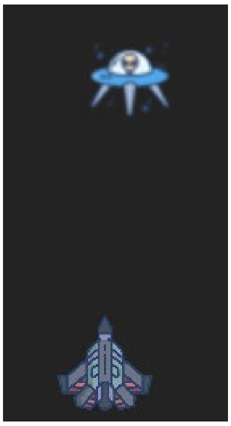
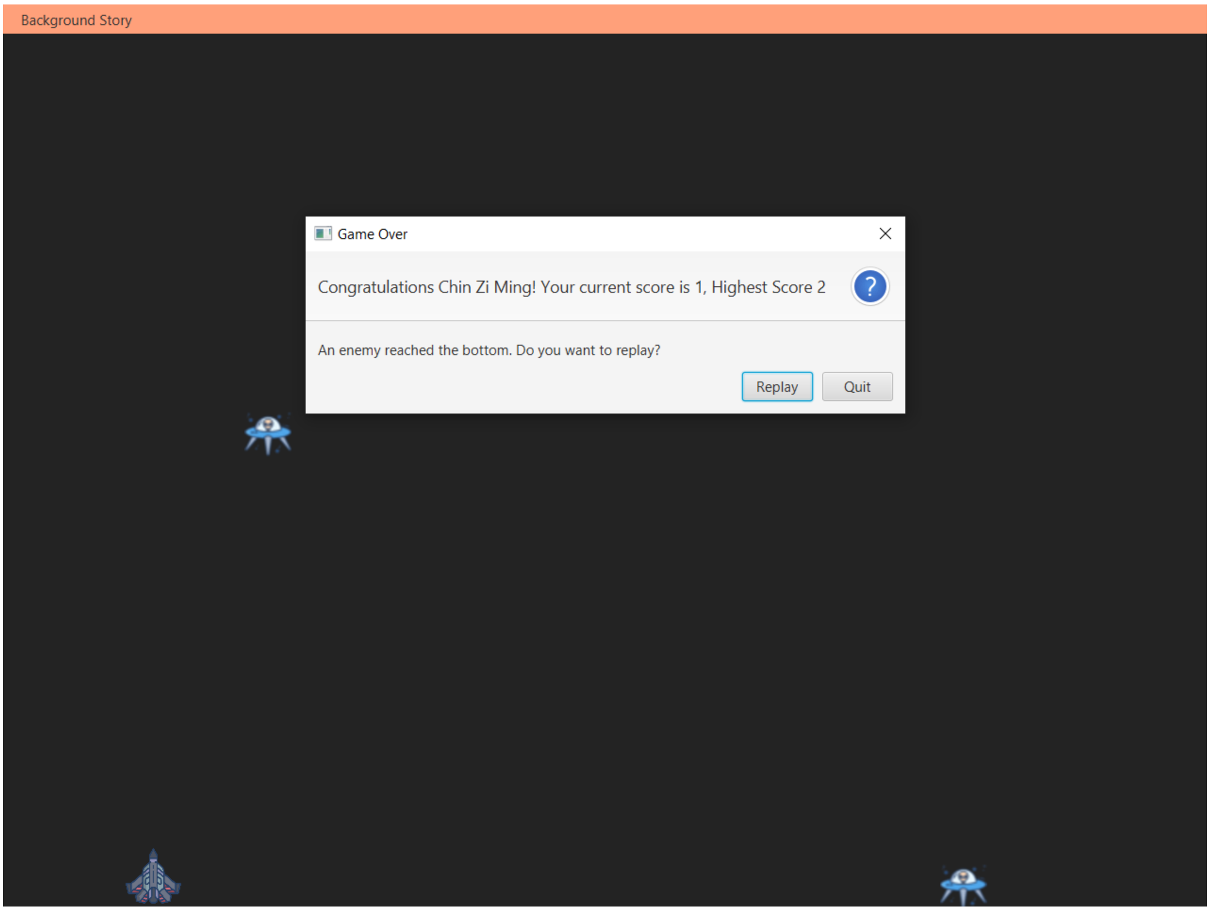


Figure 10

As shown in figure 10, collisions between the player's bullets and enemies are detected by the game. When a bullet hits an enemy, both the bullet and the enemy disappear from the game screen.

3.7 Score Tracking

3.8 Highest Score Record



Congratulations Chin Zi Ming! Your current score is 1, Highest Score 2

An enemy reached the bottom. Do you want to replay?

Figure 11

After the enemy touch the bottom of the game screen, an alert will be shown as in figure 11 and keeps track of the player's score based on the number of enemies they destroy. Furthermore, the game keeps track of the player's top score during playtime. The player is shown the highest score.

3.9 Smooth Animation

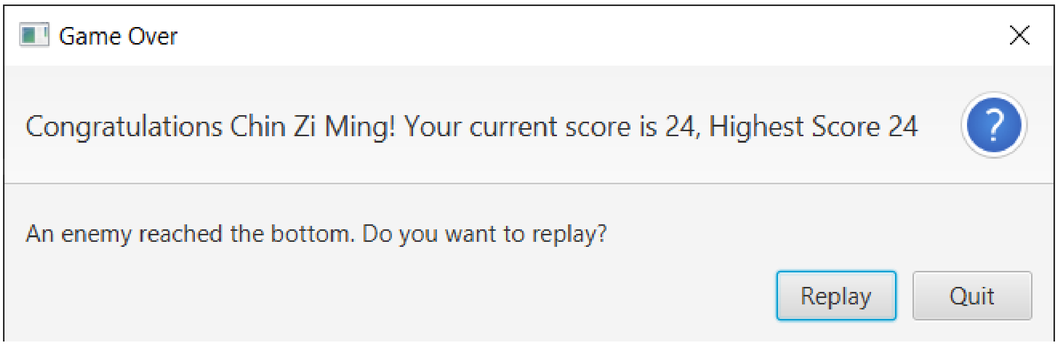
All in-game elements, such as the spaceship, bullets, and enemies, move smoothly on the screen.

3.10 Game Loop Management

The game implements a game loop using the AnimationTimer class to manage the continuous rendering and updates of the game elements.

3.11 Replay Option

3.12 Option to Quit



Game Over

Congratulations Chin Zi Ming! Your current score is 24, Highest Score 24

An enemy reached the bottom. Do you want to replay?

Figure 12

As the figure 12 shown, when an enemy reaches the bottom of the game screen, indicating that the player lost, a "Game Over" dialog appears. The player has the option to replay the game or quit to the main page. The closing (X) work similarly with the quit button where they can bring player back to main page.

4.0 PERSONAL REFLECTION

During this project, I immersed myself in software application development using Scala and JavaFX. My main goal was to meet the curriculum's educational targets, particularly focusing on improving my object-oriented programming (OOP) skills. Throughout the project, I grasped the importance of OOP principles in creating an organized and sustainable codebase. I carefully connected classes and objects, using associations, compositions, and dependencies effectively. This approach led to a well-designed software application, with clear class relationships, well-assembled components, and minimal impacts from modifications.

As the game complexity increased with more entities and animations, I faced challenges in managing performance and resource usage. While I applied best coding practices, there were occasional performance drops and memory inefficiencies. I could have further optimized the application using advanced data structures, algorithms, and profiling tools to ensure smoother gameplay, especially on lower-end systems. Better resource management, like reusing object instances, would have reduced memory usage and improved the game's responsiveness, resulting in a more polished gaming experience.

Handling game mechanics, such as player movement, enemy spawning, firing, and collision detection, posed a major obstacle. I addressed these challenges by dividing tasks and using appropriate data structures, and rigorous testing ensured a flawless game experience.

Designing a user-friendly GUI was another significant concern. Although ScalaFX offered a suitable GUI library, I carefully considered layout and design choices to create an engaging user interface.

Reflecting on my work, I recognize my strengths, such as successfully adapting the Space Invaders game mechanics and maintaining a well-organized codebase with the help of third-party libraries. However, I understand that there's always room for improvement. I can explore and apply advanced OOP concepts like abstraction and encapsulation to optimize the codebase and create a more flexible and scalable design.

In summary, this project provided valuable insights into software development and OOP. I am eager to continue refining my skills and exploring new opportunities in this dynamic field.

5.0 RFERENCES

[1] Chin. T.M. (2023) AddressApp source code [Source Code]

[2] “Logo space invaders: Space invaders arcade, Retro Arcade, Retro Logo Design,” Pinterest, https://www.pinterest.com/pin/434667801509755358/ (accessed Aug. 2, 2023).

[3] “UFO free icons designed by ICONGEEK26,” Flaticon, https://www.flaticon.com/free-icon/ufo_2909734 (accessed Aug. 2, 2023).

[4] Www.techonthenet.com, Clipart: Computer keyboard keys - space bar key (medium), https://www.techonthenet.com/clipart/keyboard/space_key_m.php (accessed Aug. 2, 2023).

[5] “Download free vectors, clipart graphics, Vector Art & design templates,” Vecteezy, https://www.vecteezy.com/free-vector/keyboard-arrows (accessed Aug. 2, 2023).

[6] “28 free icons of CDN,” Flaticon, https://www.flaticon.com/free-icons/cdn (accessed Aug. 2, 2023).