



**TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
THAPATHALI CAMPUS**

**A Final Project Report
On
Space Invaders: A Multiplayer Space Game**

Submitted By:

Atul Tiwari (THA079BEI006)

Prabal Koirala (THA079BEI019)

Prasan Gautam (THA079BEI022)

Prashant Bhusal (THA079BEI024)

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

February 2024

COPYRIGHT

The author has granted the Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering, permission to make this report available for inspection by the general public. Additionally, the author has given permission for extensive copying of this project report for scholarly purposes, subject to approval by the supervisor(s) who oversaw the project work recorded in this report, or in the absence of such supervisors, by the Head of the Department in which the project was completed. It is understood that proper attribution will be given to the author and to the Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering in any use of the material contained within this project report. The unauthorized copying or publication of this report for financial gain without the prior written consent of the Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering, and the author is strictly prohibited. Requests for permission to copy or make any other use of the material in this report, in whole or in part, should be directed to:

The Head of

Department of Electronics and Computer Engineering

Thapathali Campus, Institute of Engineering

Kathmandu, Nepal

ACKNOWLEDGEMENT

We are grateful to the Department of Electronics and Computer Engineering, Thapathali Campus for the opportunity to work on "Space Invaders," a fun multiplayer Space shooting game. Our supervisor, Er. Saroj Shakya provided invaluable guidance and support throughout the project. We also appreciate the input and feedback we received from our seniors, which helped us finalize our idea and plan of action. Additionally, we would like to thank the open-source code contributors, various programming libraries, and open-source communities that have indirectly supported the development of "Space Invaders." These resources played a crucial role in the creation of our project and we are grateful for their contributions. We also acknowledge the support and collaboration of our partners, who provided valuable resources and expertise to develop "Space Invaders." Finally, we express our appreciation to everyone who helped in the wider research and development for their ongoing contributions to the space game. We hope that "Space Invaders" will make a fun way to realize your shooting game skills.

Atul Tiwari (THA079BEI006)

Prabal Koirala (THA079BEI019)

Prasan Gautam (THA079BEI022)

Prashant Bhusal (THA079BEI024)

ABSTRACT

The project report explores the development and implementation of "Space Invaders," a dynamic and entertaining multiplayer space game. Inspired by the classic arcade game of the same name, our rendition leverages modern technologies and design principles to deliver an immersive gaming experience. Throughout the development process, the project focuses on using efficient coding techniques and optimizing the game's performance. By making the most of C++ and SFML, the game runs smoothly and uses system resources effectively. The combination of C++ and SFML allows for seamless integration of game logic, graphics, and user input, creating a responsive and enjoyable gaming experience.

Keywords: C++, Multiplayer, Space Invaders, SFML

Table of Contents

COPYRIGHT.....	i
ACKNOWLEDGEMENT.....	ii
ABSTRACT.....	iii
LIST OF FIGURES.....	1
LIST OF TABLE.....	2
LIST OF ABERRATIONS.....	2
1 INTRODUCTION:.....	4
1.1 Background:.....	4
1.2 Motivation:.....	4
1.3 Objectives:.....	5
1.4 Scope and Applications:.....	5
2 LITERATURE REVIEW:.....	6
2.1 Similar Projects:.....	6
2.1.1 Multiplayer Asteroids by Asteroids Zone.....	6
2.1.2 Galactic Combat by Stellar Games.....	7
2.2 Limitations of Similar Projects.....	7
2.2.1 Limited Player Interaction.....	7
2.2.2 Scalability Issues.....	8
2.2.3 Lack of Advanced AI.....	8
2.3 Limitations of our Projects.....	9
2.3.1 Educational Focus.....	9
2.3.2 Learning and Development.....	9
2.3.3 Limited Resources for Graphics and Design:.....	9
2.3.4 Time Constraint.....	10
2.4 Importance of our Projects.....	10
3 METHODOLOGY.....	11
3.1 System Architecture.....	11
3.1.1 Main Game Loop:.....	11
3.1.2 Classes:.....	11
3.1.2.1 Player Class:.....	11
3.1.2.2 Enemy Class:.....	11
3.1.2.3 Bullet Class:.....	12
3.1.2.4 Texture manager Class:.....	12
3.1.2.5 Sound and font manager Class:.....	12
3.1.2.6 Splash Class:.....	12

3.1.2.7 Menu Class:	12
3.1.2.8 Gameover Class:	12
3.1.2 Graphics and Display:	13
3.1.4 User Input Handling:	13
3.1.5 Collision Detection:	13
3.1.6 Game State Management:	13
3.2 Operation:	13
4 PROPOSED SYSTEM DESCRIPTION:	15
4.1 Block Diagram of System Architecture:	15
4.2 Parts of program:	16
4.2.1 Splash Screen:	16
4.2.2 Menu Menu:	17
4.2.3 Playing State:	17
4.2.4 Game Over:	18
4.3 Tools and Environment:	19
4.3.1 Libraries:	19
4.3.2 Tools:	19
5 RESULTS AND ANALYSIS:	20
5.1 Results:	20
5.1.1 Gameplay Functionality:	20
5.1.2 User Interface:	20
5.1.3 Graphics:	21
5.1.4 Game Dynamics:	21
5.2 Analysis:	22
5.2.1 Modular Code Structure:	22
5.2.2 SFML Integration:	22
5.2.3 Resource Management:	23
5.2.4 Error Handling and Debugging:	23
5.2.5 Documentation:	23
6 CONCLUSION AND FUTURE ENHANCEMENT:	25
6.1 Conclusion:	25
6.2 Future Enhancement:	25
7 TIME ESTIMATION:	27
8 REFERENCE:	28

LIST OF FIGURES

Figure 4.1 Block Diagram of project.....	15
Figure 4.2.2.1 Splash Screen.....	16
Figure 4.2.2.2 Main Menu.....	17
Figure 4.2.2.3 Playing State.....	18
Figure 4.2.2.4 Game Over.....	18

LIST OF TABLE

Table 7.1 Time estimation chart	27
---------------------------------------	----

LIST OF ABERRATIONS

API	Application Programming Interface
CPP	C Plus Plus
GUI	Graphical User Interface
IDE	Integrated Development Environment
SFML	Simple and Fast Multimedia Library
UI/UX	User Interface/ User eXperience

1 INTRODUCTION:

The world of gaming has evolved significantly since the advent of classic arcade games, offering an immersive and engaging experience to players across diverse platforms. In line with this evolution, our project introduces "*Space Invaders*," a multiplayer space game that pays homage to the timeless appeal of the original arcade classic. Developed using C++ and the SFML, *Space Invaders* aims to provide a contemporary and exhilarating gaming experience that captures the essence of intergalactic warfare.

1.1 Background:

The inspiration for *Space Invaders* stems from the iconic arcade game released in the late 1970s, which has been reimagined in this project to deliver an immersive and competitive gaming experience. Originally conceived in the era of arcade gaming, our project, *Space Invaders*, aims to be adapted with modern design principles and features, creating a dynamic and engaging gameplay environment. We aim for the use of C++ and SFML as the primary development tools to ensure a robust and efficient framework for crafting an engaging and visually appealing gaming environment. Our project builds upon this foundation, incorporating modern technologies and design principles to create a multiplayer space game that appeals to both nostalgic gamers and new generations seeking thrilling gameplay experiences.

1.2 Motivation:

The motivation behind developing the *Space Invaders* game lies in bringing a nostalgic yet innovative gaming experience to a contemporary audience. By combining the beloved elements of the original game with new features such as player-vs-player interactions, dynamic asteroid movements, this project aims to capture the essence of classic gaming while introducing exciting twists to keep players enthralled. Furthermore, the project serves as a practical exploration of game development using C++ and SFML, offering valuable insights into the challenges and rewards of creating a multiplayer gaming experience.

1.3 Objectives:

The main objectives of the Space Invaders project is:

- To develop a two-player game that offers a blend of competition, strategy, and engaging gameplay.
- To reinvent a classic arcade game

1.4 Scope and Applications:

The scope of the Space Invaders game encompasses various elements, including player dynamics, asteroid behaviors, bullet interactions. The scope also allows for potential future expansions, such as additional player abilities, diverse asteroid types, and multiplayer modes, enhancing the game's versatility and replay value. The project aims to provide not only an enjoyable gaming experience but also a platform for potential growth and innovation within the gaming community.

Through this project, we aim to contribute to the gaming community while honing our skills in game development, programming, and project management. The combination of classic gameplay and modern technology positions Space Invaders as a distinctive addition to the realm of multiplayer space gaming.

2 LITERATURE REVIEW:

In the realm of game development, frameworks serve as the backbone for crafting captivating and interactive experiences. Among these, the SFML stands out, celebrated for its user-friendly interface, adaptability, and cross-platform capabilities. This literature review explores the intricate nexus between SFML and the development of "*Space Invaders*," a seminal multiplayer space shooter game.

Beginning with a retrospective on the evolution and enduring popularity of space shooter games, the review situates "*Space Invaders*" within this esteemed lineage, highlighting its unique contributions and gameplay dynamics. A deep dive into SFML's functionalities reveals its indispensable role in facilitating key "*Space Invaders*," aspects, such as graphics rendering, input management, and audio integration.

Through a curated selection of tutorials and case studies, the review elucidates the practical applications of SFML in developing multiplayer features, shedding light on the technical nuances and design considerations specific to "*Space Invaders*." Furthermore, the review addresses inherent challenges in multiplayer game development, emphasizing the pivotal role of user experience design and iterative player feedback in refining the gameplay mechanics and overall immersive experience.

2.1 Similar Projects:

2.1.1 Multiplayer Asteroids by Asteroids Zone

- This project features a multiplayer space-themed game with static and moving asteroids.
- Players control spaceships and navigate through the asteroid field to reach a designated endpoint.
- The game allows players to shoot bullets to destroy asteroids and each other.

2.1.2 Galactic Combat by Stellar Games

- Galactic Combat is a multiplayer space shooter that incorporates elements of *Space Invaders*.
- Players control customizable spaceships and engage in dogfights while navigating through obstacles.
- The game includes both static and dynamic obstacles, creating an engaging and challenging environment.
- Respawn mechanisms are implemented for defeated players.

2.1.3 Space Race Warriors by Nebula Studios

- Space Race Warriors is a multiplayer racing game set in space with elements reminiscent of *Space Invaders*.
- Players race through a course filled with obstacles, including asteroids that can be destroyed.
- The game introduces a competitive twist to reach the finish line before opponents.
- Features include player-versus-player interactions and respawn mechanics.

2.2 Limitations of Similar Projects

Here are some potential limitations and negative aspects of similar projects that have explored the use of game development and area networking.

2.2.1 Limited Player Interaction

While existing projects incorporate multiplayer elements, the depth of player interaction is often limited. Most games focus on individual player experiences rather than fostering direct player-versus-player (PvP) engagements. Implementing robust

PvP mechanics, such as enhanced combat systems and collaborative strategies, could elevate the multiplayer experience.

2.2.2 Scalability Issues

Some multiplayer games encounter scalability challenges when accommodating a larger number of players. This issue may result in reduced performance, lag, or even server instability. Addressing scalability concerns is vital for ensuring a seamless and enjoyable gaming experience, particularly in scenarios where multiple players are concurrently navigating through the game environment.

2.2.3 Lack of Advanced AI

Existing projects commonly feature basic artificial intelligence (AI) that primarily governs the movement of asteroids or non-player entities. To enhance gameplay dynamics, the development of more sophisticated AI behaviors for opponents and allies is crucial. This could involve implementing adaptive strategies, dynamic decision-making, and a more challenging AI that adapts to players' skills.

2.2.4 Graphics and User Interface (UI) Design

Some projects may fall short in terms of visually appealing graphics and intuitive UI design. To captivate the players, it's essential to invest in high-quality graphics, special effects, and an immersive visual experience. Additionally, a user-friendly interface contributes significantly to player engagement. Ensuring that the UI is intuitive, easily navigable, and aesthetically pleasing will enhance the overall gaming experience.

2.2.5 Network Latency and Synchronization

Multiplayer games, especially those involving real-time interactions, can be sensitive to network latency. Ensuring smooth synchronization between players' actions in both local and online multiplayer modes is challenging. Addressing issues related to latency, lag compensation, and synchronization discrepancies is vital for creating a seamless and fair gaming environment.

2.3 Limitations of our Projects

Here are some potential limitations of the “*Space Invaders*” project:

2.3.1 Educational Focus

This project is fundamentally an educational endeavor aimed at teaching the team about game development, particularly in the realm of multiplayer games. This educational context inherently imposes limitations on our project. Specifically, we are constrained by the finite time and resources available for this initiative. As a result, while we have planned to create a prototype that captures the essence of a multiplayer experience inspired by "*Space Invaders*," it is essential to recognize that this version may not encompass all the features and complexities that could be present in a fully realized commercial game.

2.3.2 Learning and Development

One of the primary objectives of this project is to facilitate the learning and skill development of our team members. As such, the project serves as a platform for us to gain hands-on experience and insight into the intricacies of creating a multiplayer shooting game. Given our evolving understanding and ongoing learning curve, the scope of the project is deliberately limited to the implementation of a foundational prototype. This prototype serves as a testament to the potential of "*Space Invaders*" while also serving as a practical learning tool for our team.

2.3.3 Limited Resources for Graphics and Design:

The visual appeal of the game is essential for player engagement. If there are constraints on resources, such as limited graphics assets or design expertise, it could impact the overall aesthetics of the game. Striking a balance between visual appeal and available resources will be a consideration.

2.3.4 Time Constraint

Time is another critical factor that influences the development trajectory of "*Space Invaders*." The project operates within a predefined timeline, which introduces a sense of urgency and necessitates efficient time management. This temporal constraint limits our capacity to engage in extensive research, experimentation, and iteration. However, it also serves as a catalyst for focused effort and disciplined execution.

2.3.5 Balancing Gameplay Mechanics

Achieving a balance in gameplay, including factors like player abilities, weapons, and respawn mechanics, can be challenging. Fine-tuning these aspects to create a challenging yet fair experience may require thorough testing and iteration.

2.4 Importance of our Projects

Here are some important of this project:

- Provides hands-on learning experiences in game development, fostering skills and knowledge in a practical context.
- Promotes social interaction and collaboration through multiplayer gameplay, fostering a sense of community among players.
- Enables the integration of modern technologies in game development, pushing the boundaries of what's possible in the gaming landscape.
- Developing a multiplayer game using C++ and SFML provides an excellent opportunity for skill development among the project's developers.
- The primary importance of the project lies in delivering an entertaining and enjoyable gaming experience for players.

3 METHODOLOGY

3.1 System Architecture

The architecture of the "Space Invaders" game is designed with a modular and organized structure, incorporating various components to achieve the desired gameplay experience. Below is a description of the key architectural elements:

3.1.1 Main Game Loop:

- The game is centered around a main game loop that continuously iterates, updating the game state, handling user input, and refreshing the display.
- The loop ensures that the game remains interactive and responsive, processing events such as player movements, bullet trajectories, and asteroid animations.

3.1.2 Classes:

The game architecture employs Object-Oriented Programming (OOP) principles, utilizing classes to encapsulate data and behavior for different game entities.

3.1.2.1 Player Class:

- Manages each player's position, movement and shooting capabilities.
- Includes methods for player movement, shooting bullets, and handling collisions.
- Incorporates features like acceleration and deceleration to provide smoother player control.

3.1.2.2 Enemy Class:

- This class represents the asteroids(enemy) moving from right side of the screen to left.
- Defines properties like size, position, and movement behavior.
- Incorporates methods for updating asteroid positions and checking collisions.

3.1.2.3 Bullet Class:

- Handles bullets shot by players.
- Includes attributes such as speed, trajectory, and collision detection.

3.1.2.4 Texture manager Class:

- Manage all the textures that are used in this game

3.1.2.5 Sound and font manager Class:

- This class manages all the sounds and fonts that are used in the game.

3.1.2.6 Splash Class:

- This class represents the splash screen which is displayed at the start of the game.
- It defines properties like size, position, and behavior of the splash screen.

3.1.2.7 Menu Class:

- The Menu class represents the main menu of the game. It serves as the interface through which players can navigate and access various game options and functionalities.
- It initializes its properties and objects during construction, including setting up the menu text, and creating player objects.

3.1.2.8 Gameover Class:

- The GameOver class represents the state of the game when it ends.
- It is responsible for displaying the game over screen, including player scores and options for starting a new game.
- The GameOver class is activated when the game ends.
- It displays the final scores of the players and provides options for starting a new game or quitting.

3.1.2 Graphics and Display:

- Utilizes graphics libraries, such as SFML, for rendering the game environment.
- Incorporates functions for drawing players, asteroids, bullets, and other graphical elements on the screen.
- Updates the display within the main game loop to reflect changes in the game state.

3.1.4 User Input Handling:

- Implements functions for handling user input to control player movements and shooting.
- Listens for keyboard or other input device events within the main game loop.

3.1.5 Collision Detection:

- Integrates collision detection mechanisms to identify interactions between game entities.
- Checks for collisions between bullets and asteroids, player collisions with asteroids, and other relevant interactions.

3.1.6 Game State Management:

- Manages the overall state of the game, including player positions, asteroid positions, and bullet trajectories.
- Keeps track of player scores, time and other game-related data.

3.2 Operation

In "Space Invaders," players take control of two separate spaceships on the same computer, each using distinct sets of controls. Player 1 maneuvers their spaceship using the W, A, S, and D keys, while Player 2 navigates theirs with the arrow keys.

The primary objective is to accumulate as many points as possible within a strict time limit of 3 minutes by destroying incoming asteroids with the spaceship's bullets.

The gameplay unfolds as asteroids gradually descend from a predetermined direction, each carrying a score value of 5 points when successfully eliminated. Players must strategically position their spaceships to evade collisions with the asteroids while simultaneously aligning their firing trajectory to shoot them down. Colliding with an asteroid has dire consequences: the unfortunate player's health gets reduced, adding an element of risk to the gameplay.

Throughout the game, players' scores are prominently displayed on the screen, providing constant feedback on their progress. The competition intensifies as the clock ticks down, with each successful asteroid destruction contributing to the players' overall scores. At the end of the 3-minute time limit, the game concludes, and the player with the highest score emerges victorious.

"Space Invaders" offers a dynamic multiplayer experience that combines precise controls, strategic decision-making, and fast-paced action. With the added challenge of managing health while pursuing high scores, it provides an immersive and engaging gaming experience for players of all levels.

4 PROPOSED SYSTEM DESCRIPTION

The different blocks of the system architecture are explained below:

4.1 Block Diagram of System Architecture

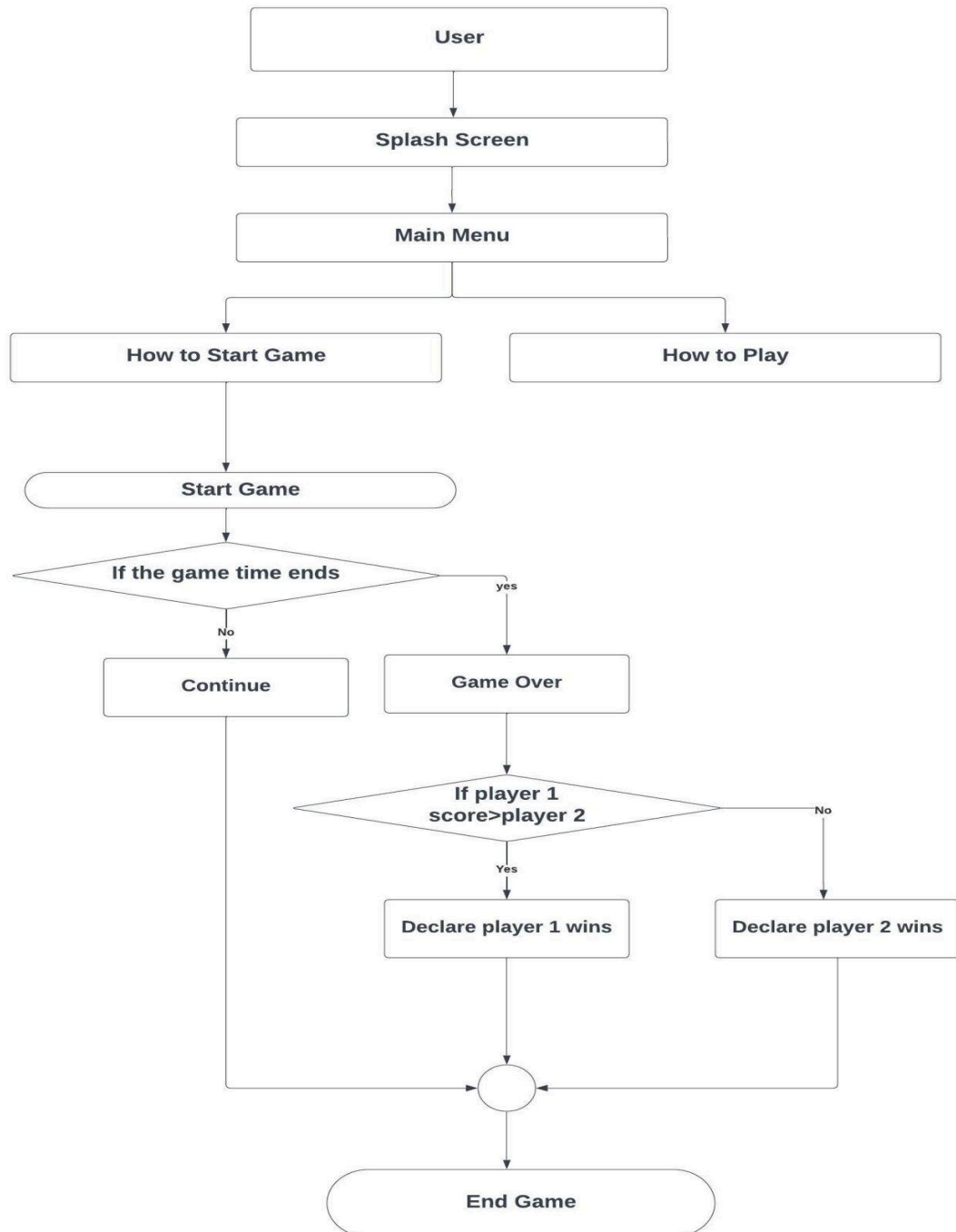


Fig 4.1: Block diagram of system architecture

4.2 Parts of program

4.2.1 Splash Screen

When the game starts, a splash screen is displayed for 5 seconds showing the logo of our game.

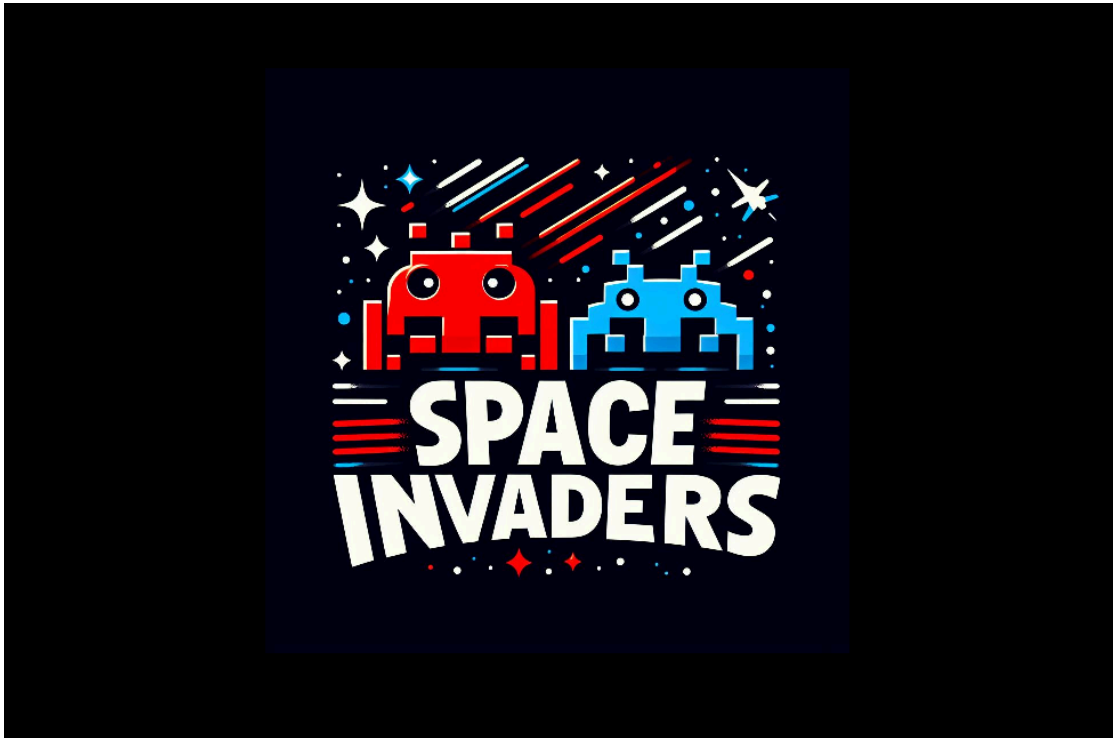


Fig 4.2.1: Splash screen

4.2.2 Menu Menu

The menu page of the game contains the title “*Space Invaders*” and the menu which contains options to the user which are: Start Game and How to play.



Fig 4.2.2: Main menu

4.2.3 Playing State

In this state two spaceships are displayed. In the left and right corner of the upper part of the window players score is displayed and in the upper middle part remaining time is displayed.



Fig 4.2.3: Main game screen

4.2.4 Game Over

In this state the player's score and the winner of the game is displayed.

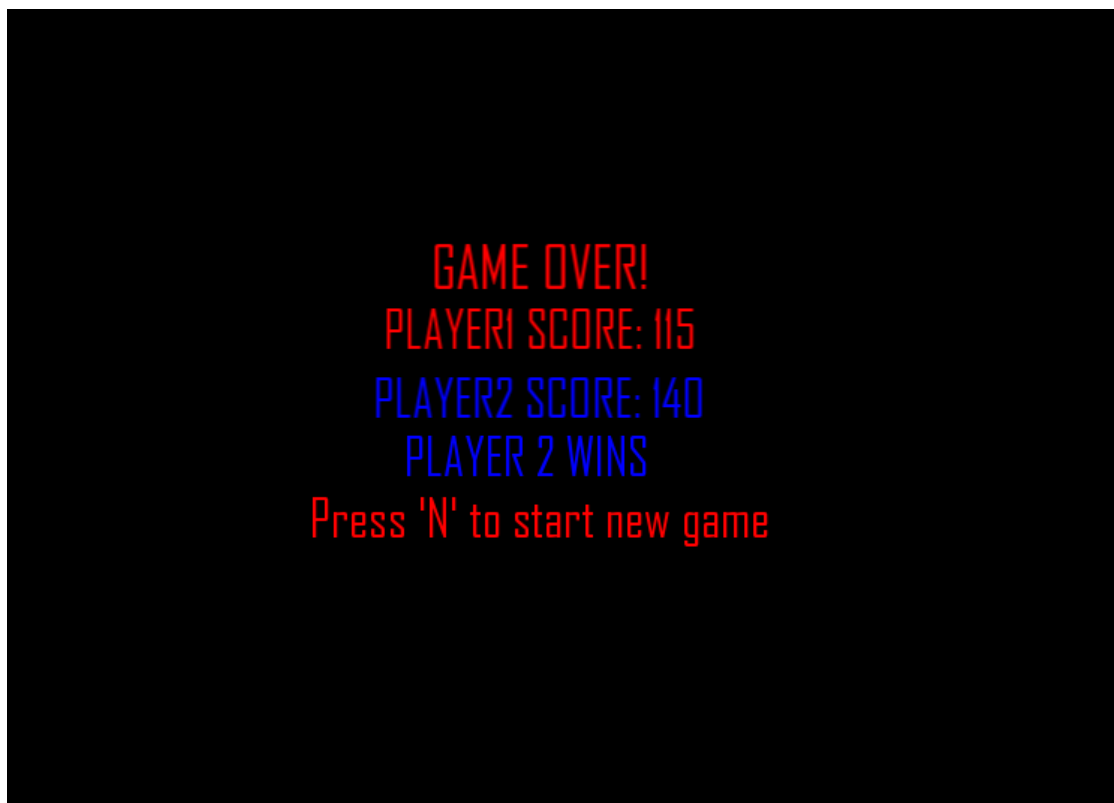


Fig 4.2.4: Game over

4.3 Tools and Environment

The tools and environment that will be used in this project are:

4.3.1 Libraries

The libraries that are likely to be used in this project are:

- SFML library

4.3.2 Tools

- Visual Studio
- C++ compiler
- Adobe photoshop and Canva
- Github

5 RESULTS AND ANALYSIS

5.1 Results

5.1.1 Gameplay Functionality:

In "Space Invaders," gameplay functionality revolves around the intense competition between two players as they navigate their respective spaceships through a field of asteroids. Each player controls their spaceship's movements and firing capabilities, aiming to destroy as many asteroids as possible within the allotted time frame. The game's mechanics are simple yet engaging, with players needing to balance offense and defense to outmaneuver their opponent and secure victory. As players blast their way through the asteroid field, strategic positioning, precise aiming, and quick reflexes are crucial for success. The dynamic nature of the gameplay ensures that no two matches are alike, keeping players on the edge of their seats as they strive to claim the title of ultimate space invader. Through comprehensive analysis of gameplay functionality, including player interactions, controls, and balance, we aim to provide valuable insights into the effectiveness and enjoyment of the gaming experience.

5.1.2 User Interface:

The user interface of "Space Invaders" serves as the gateway for players to immerse themselves in the thrilling multiplayer space shooting experience. Upon launching the game, players are greeted with our logo, which animates seamlessly into the main menu. Here, players find clear and concise instructions on how to play the game, ensuring a smooth onboarding experience for newcomers. To initiate gameplay, players simply need to tap the "S" key, seamlessly transitioning from the menu to the heart of the action. Once the game begins, players control their respective red and blue spaceships, each equipped with their own firing mechanics to shoot asteroids. The asteroids dynamically spawn from the right side of the screen, adding an element of unpredictability and challenge to the gameplay. Overall, the user interface of "Space Invaders" is designed to be intuitive, visually appealing, and seamlessly integrated with the gameplay mechanics to enhance the overall player experience.

Through comprehensive analysis of the user interface, including usability, aesthetics, and player feedback, we aim to evaluate its effectiveness in facilitating player engagement and enjoyment.

5.1.3 Graphics:

In "Space Invaders," the graphics play a crucial role in immersing players into the intense space battle atmosphere. Leveraging the SFML library, we've crafted a visually striking experience with a dark-themed aesthetic. The use of SFML allows for smooth rendering of high-quality graphics, enhancing the overall visual appeal of the game. Players are greeted with detailed and dynamic space backgrounds, setting the stage for epic space skirmishes. The spaceships are intricately designed, each with its own distinct appearance and color scheme to differentiate between players. As players navigate through the asteroid-filled cosmos, the smooth animations and particle effects add depth and excitement to the gameplay. The asteroids themselves are rendered with attention to detail, providing a satisfying visual feedback as they explode upon impact with the players' bullets. Overall, the graphics in "Space Invaders" contribute significantly to the immersive gaming experience, drawing players into the thrilling world of intergalactic warfare. Through analysis of graphics performance and player feedback, we aim to assess the effectiveness of the visual elements in enhancing player engagement and enjoyment.

5.1.4 Game Dynamics:

In "Space Invaders," the game dynamics are designed to provide players with an intuitive and engaging gameplay experience. Players control their respective spaceships using the WASD and arrow keys, allowing for smooth and responsive navigation through the vastness of space. The shooting mechanics are equally straightforward, with one player utilizing the "F" key and the other player using the "Space" key to unleash a barrage of bullets upon the incoming asteroids. This simple control scheme ensures that players can focus on the action-packed gameplay without being overwhelmed by complex controls. As time progresses, the intensity of the game increases, with asteroids spawning at an escalating rate, challenging players to react swiftly and strategically to avoid collisions and destroy as many asteroids as

possible. Through comprehensive analysis of game dynamics, including player interactions, control responsiveness, and pacing, we aim to evaluate the effectiveness of the gameplay mechanics in delivering an enjoyable and competitive multiplayer experience.

5.2 Analysis

5.2.1 Modular Code Structure:

- Modular code structure simplifies updates and maintenance, ensuring each component can be modified independently.
- Enhanced readability results from breaking down complex functions into smaller, manageable units.
- Scalability and extensibility are facilitated by the modular approach, enabling seamless integration of new features.
- Testing and debugging are simplified as each module can be isolated and examined independently.
- Code reuse is promoted, fostering modularity and facilitating the integration of third-party libraries.
- Clear separation of concerns observed in distinct modules (player controls, game logic, graphics rendering).
- Object-oriented programming principles effectively applied with encapsulation of game entities within classes.

5.2.2 SFML Integration:

- Efficient use of SFML's Sprite and Texture classes for rendering game entities and graphical assets.
- Seamless communication protocol enables interactions between game instances.

5.2.3 Resource Management:

- Effective handling of loading and unloading game assets.
- Potential use of SFML's Resource classes or custom resource manager.

5.2.4 Error Handling and Debugging:

- Robust error handling mechanisms aid in issue identification and resolution.
- Utilization of SFML's debugging features for efficient debugging processes.

5.2.5 Documentation:

In the documentation for our "Space Invaders" project, we aim to provide comprehensive guidance and information to developers, users, and stakeholders. The documentation encompasses various aspects of the project, including installation instructions, gameplay mechanics, and code structure. To ensure clarity and accessibility, the documentation is organized into distinct sections, each addressing specific aspects of the game.

The installation section of the documentation provides step-by-step instructions for setting up and running the game on different platforms. This includes downloading and installing any necessary dependencies, such as the SFML library, configuring development environments, and compiling the source code. Additionally, troubleshooting tips and common issues are addressed to assist users in resolving any potential installation challenges.

The gameplay section offers detailed explanations of the game's mechanics, controls, and objectives. Players are guided through the core gameplay loop, including navigating their spaceships, shooting asteroids, and competing against opponents. Additionally, strategies and tips for maximizing scores and outmaneuvering opponents are provided to enhance the player experience. Visual aids, such as

screenshots and diagrams, are included to illustrate key concepts and gameplay elements effectively.

The code documentation delves into the structure and organization of the game's source code, providing insights into the underlying architecture and design decisions. Each module and component of the codebase is thoroughly documented, including class definitions, function descriptions, and variable explanations. This enables developers to understand how different parts of the game interact and how to extend or modify the code to suit their needs. Additionally, coding standards and best practices are outlined to ensure consistency and maintainability across the codebase. Overall, the documentation serves as a valuable resource for anyone involved in the "Space Invaders" project, from users seeking to play the game to developers looking to understand and contribute to its development.

6 CONCLUSION AND FUTURE ENHANCEMENT

6.1 Conclusion

The Multiplayer *Space Invaders* project demonstrates the successful fusion of classic arcade gaming with contemporary multiplayer dynamics. The development process focused on delivering a visually appealing and engaging gameplay experience that resonates with both traditional gamers and newcomers alike. The use of C++ and SFML ensures a robust and scalable codebase, laying the groundwork for potential future enhancements. The engaging gameplay, strategic elements, and competitive multiplayer features collectively contribute to a thoroughly enjoyable gaming experience.

6.2 Future Enhancement

The potential avenues for future enhancement and expansion of this *Space Invaders* game are numerous:

- Online Multiplayer Mode:

Implement network functionality to allow players to compete against each other online. This would involve setting up server-client architecture to facilitate real-time gameplay between players over the internet. Players could join lobbies, invite friends, or be matched with random opponents for intense multiplayer battles.

- Enhanced Graphics and Visual Effects:

Improve the visual experience of the game by adding more detailed spaceship and asteroid designs, dynamic lighting effects, and particle effects for explosions and gunfire. Enhanced graphics can significantly enhance the immersion and appeal of the game, providing players with a more visually engaging experience.

- Additional Power-Ups and Abilities:

Introduce various power-ups and special abilities that players can collect during gameplay. These could include temporary shields for protection against asteroids, rapid-fire modes for increased shooting speed, or smart bombs for clearing the screen of obstacles. Adding power-ups adds depth to the gameplay and provides players with strategic choices during gameplay.

- Dynamic Environments and Obstacles:

Introduce dynamic environmental elements and obstacles that affect gameplay. For example, moving asteroids that change direction or speed, asteroid clusters that require precise maneuvering to navigate, or gravitational fields that alter the trajectory of bullets and asteroids. Dynamic environments add variety and unpredictability to gameplay, keeping players engaged and challenged.

Implementing these enhancements would not only elevate the overall quality of our "Space Invaders" game but also increase its replay value, player engagement, and appeal to a wider audience. We want to prioritize these enhancements based on our resources, development capabilities, and target audience preferences.

7 TIME ESTIMATION

The time is allocated for different topics with respect to their needs, as shown in the table below:

Topics/Time	2023			2024				
	5-Dec	25-Dec	30-Dec	5-Jan	25-Jan	10-Feb	15-Feb	17-Feb
Research								
Familiarization of Tools								
Writing Proposal								
Designing								
Coding								
Testing and Debugging								
Documentation and Report								

Table 7.1: Time Estimation Chart

8 REFERENCE

- [1] Wikipedia Contributors, “Space Invaders,” *Wikipedia*, Apr. 16, 2019.
https://en.wikipedia.org/wiki/Space_Invaders
- [2] “2.6 Tutorials (SFML / Learn),” *www.sfm-dev.org*.
<https://www.sfm-dev.org/tutorials/2.6/>
- [3] “SFML — Simple and Fast Multimedia Library,” *GitHub*, Dec. 15, 2022.
<https://github.com/SFML/SFML>
- [4] Raimondas Pupius, *SFML game development by example : create and develop exciting games from start to finish using SFML*. Birmingham, Uk: Packt Publishing, 2015.
- [5] “SFML C++ Tutorial 12 (PART 1/4) | 2D Space shooter game using Sprites, Textures and UI with Text,” *www.youtube.com*.
<https://www.youtube.com/watch?v=sWYpNIrBr6U&t=10s>