



**TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS**

**A PROJECT REPORT ON
BAAL ARJUN : A 2D ARROW SHOOTING GAME**

Submitted by:
ANUJ SHRESTHA (078BCT014)
ARISHA PRASAIN (078BCT019)
AYUSH ADHIKARI (078BCT024)

**A PROJECT REPORT TO THE DEPARTMENT OF
ELECTRONICS AND COMPUTER ENGINEERING ON OBJECT
ORIENTED PROGRAMMING APPLICATION USING C++**

**SUBMITTED TO:
DEPARTMENT OF ELECTRONICS AND COMPUTER
ENGINEERING**

August 25, 2023

ACKNOWLEDGEMENT

We would like to extend our sincere gratitude to our esteemed Lecturer, Mr. Daya Sagar Baral, for granting us the invaluable opportunity to undertake the project work on Object-Oriented Programming in C++. We express our heartfelt appreciation for his invaluable support and guidance, which played a crucial role in ensuring the success of this project. We would also like to express our deepest gratitude to the Department of Electronics and Computer Engineering at the Institute of Engineering, Pulchowk Campus, for providing us with the opportunity to collaborate on this endeavor. Additionally, we extend our thanks to our laboratory instructors for their guidance and support throughout our work in the computer lab.

We are immensely thankful to all our friends who have contributed to the development of this project, and we sincerely appreciate their assistance. We welcome constructive criticism and recommendations in any form, and we assure you that they will be received with an open heart, valued, and duly acknowledged.

Authors:

Anuj Shrestha
Arisha Prasain
Ayush Adhikari

ABSTRACT

The principal aim of this project was to create a gaming program utilizing an Object-Oriented Programming language, C++. Within the scope of this project, we crafted a 2D game centered around arrow-based combat using the Simple and Fast Multimedia Library (SFML) to handle the graphical user interface. The underlying objective of developing this game also encompassed the insights into the world of game development.

Baal Arjun stands as a 2D, single-player computerized game. This game constitutes a simple two-dimensional rendition of a battlefield, presented from a top-down perspective. In this scenario, the player assumes the role of the character 'Arjuna' and engages in shooting arrows at adversaries and their cohorts. The initial three levels have been designed to function as a means of familiarizing the player with the game interface. Successful completion of each level hinges upon the player's ability to triumph over the adversaries prior to succumbing to their own defeat. Should the player emerge successful in this endeavor, they attain victory; conversely, failure to do so results in defeat.

Keywords: *Baal Arjun, SFML, Object-Oriented Programming, C++*

Contents

ACKNOWLEDGEMENT	i
ABSTRACT	ii
1 INTRODUCTION	1
2 OBJECTIVES	2
3 APPLICATION	3
3.1 User Engagement	3
3.2 Skill Development	3
3.3 Cultural Appreciation	3
3.4 Learning and Engagement with Mahabharat	3
3.5 Entertainment Industry Impact	3
4 LITERATURE REVIEW	4
4.1 Harmonizing Cultural Legacy with Technological Ingenuity	4
4.2 Elevating Cultural Representation in Gaming	4
4.3 Cultivating Educational Exploration	4
4.4 Navigating Challenges and Reaping Rewards	4
4.5 A Paradigm Shift in Gaming	5
5 METHODOLOGY	6
5.1 Research and Planning	6
5.2 Algorithm Design	6
5.3 Sources	6
5.4 Game Design	6
5.5 Maintenance	6
5.6 Testing and Debugging	7
5.7 Documentation	7
6 IMPLEMENTATION	8
6.1 Approach	8
6.1.1 Tools and Technologies	8
6.1.2 System Architecture and Design	8
6.1.3 Code Development	8
6.1.4 Resource Collection	9
6.1.5 Integration and Testing	9
6.1.6 Version Control and Collaboration	9
6.2 Flow Chart	10
6.3 Block Diagram	11
6.4 Game Pictures	12
7 RESULTS	15
8 PROBLEM FACED AND SOLUTIONS	16

9 LIMITATIONS AND FUTURE ENHANCEMENTS	17
9.1 Limitations	17
9.2 Future Enhancements	17
10 CONCLUSION AND RECOMMENDATIONS	18
11 REFERENCES	19

List of Figures

1 Proposed Loading Screen	1
2 Flow Chart	10
3 Generalized Block Diagram	11
4 Main Screen	12
5 Level Selection Screen	12
6 Level 2	13
7 Level 7	13
8 Generalized Block Diagram	14
9 Generalized Block Diagram	14

1 INTRODUCTION

Baal Arjun is a 2D arrow shooting game that takes inspiration from the esteemed character Arjuna in the epic Mahabharata, who is widely regarded as the preeminent archer of that era. This game, known as "Baal Arjun," offers an engaging single-player experience, consisting of three chapters, each comprising numerous levels.

Within each level, the player navigates the screen and utilizes both the mouse and keyboard to precisely adjust the angle of the character named 'Arjun' in order to strike the target. The angle and position of the character can be modified until the left mouse button is clicked. Once the left button is released, the arrow swiftly propels toward the designated target coordinate. Initially, the player is allotted a limited number of arrows and possesses a finite life. With each successful hit on the target, the player's score increases. If the target is missed, the player continues playing the same level until all arrows are depleted. To successfully pass a level, all targets must be accurately hit within the designated time frame, subsequently unlocking the next level.

In the initial stages, the targets remain stationary, gradually transitioning into a non-stationary state as the levels progress, thus adding an element of challenge and dynamism to the gameplay.

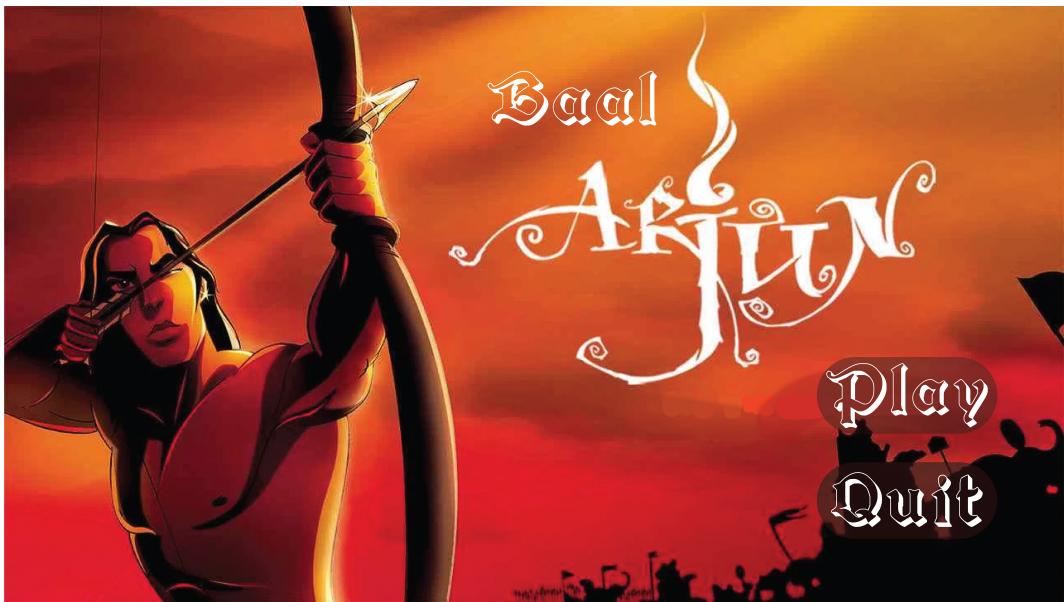


Figure 1: Proposed Loading Screen

2 OBJECTIVES

The primary objective of this project is to facilitate the practical application of the Object-Oriented Programming (OOP) model using the well-established and effective programming language, C++. The key objectives pertaining to the development of this project are outlined as follows:

1. To gain a comprehensive understanding of the Object-Oriented Programming paradigm, encompassing its principles and successfully implementing them in the project.
2. To explore the fundamental as well as advanced features offered by C++.
3. To acquire knowledge and proficiency in designing custom header files, thereby familiarizing ourselves with the concepts of modularity and reusability.
4. To acquire a foundational understanding of game development and testing, with a specific focus on the development of game.
5. To become acquainted with graphics libraries such as SFML, enabling the design and implementation of a user-friendly user interface (UI).
6. To develop a program that exhibits high performance and efficiency, taking the factors such as memory management, time constraints, algorithmic complexity, and optimal resource utilization into consideration.
7. To enhance our abilities in teamwork and collaborative communication, fostering effective cooperation among project members.

By pursuing these objectives, we aim to gain practical knowledge and skills in OOP principles, C++ programming, modular design, game development practices, UI design, and optimization techniques. Moreover, this project serves as an opportunity to strengthen our collaborative abilities, enhancing our team dynamics and communication skills.

3 APPLICATION

Our project involves developing an exciting game based on the character Arjun from Mahabharat. Players will control Arjun and use their mouse to shoot arrows at targets while moving in the game world.

3.1 User Engagement

We aim to create an engaging game that immerses players in the world of Arjun. By providing an enjoyable experience, we hope to entertain players and make them feel connected to the character and the epic.

3.2 Skill Development

The gameplay mechanics, such as arrow shooting and precision aiming, offer players the chance to improve their hand-eye coordination, concentration, and reflexes. These skills can be valuable not just in the game, but also in real-life situations.

3.3 Cultural Appreciation

Our game's theme, inspired by Mahabharat, aims to spark interest and curiosity among players who are fans of the epic or have an interest in Hindu mythology and culture. We want to contribute to cultural appreciation and raise awareness of this rich heritage.

3.4 Learning and Engagement with Mahabharat

Through the gameplay and accompanying content, we want to provide players with an opportunity to learn more about Arjun and his role in Mahabharat. By delving into the mythology, players can gain insights and develop a deeper understanding of the epic.

3.5 Entertainment Industry Impact

If our game gains popularity and positive reception, it could have an impact on the entertainment industry. It may inspire similar projects and encourage the exploration of Indian mythological themes in gaming, diversifying the offerings in the industry.

Our project aims to develop an immersive and enjoyable game centered around Arjun from Mahabharat. We want to provide players with an engaging experience, improve their skills, foster cultural appreciation, facilitate learning about Mahabharat, and potentially contribute to the broader landscape of the entertainment industry.

4 LITERATURE REVIEW

”Baal Arjun” stands as a captivating 2D arrow shooting game, deeply rooted in the narrative tapestry of the Mahabharata. In this innovative gaming venture, players assume the role of Arjun, embarking on a solitary quest to confront adversaries. The game’s premise revolves around Arjun’s adeptness at evading enemy arrows while expertly aiming and retaliating. By seamlessly blending cultural heritage with cutting-edge technology, ”Baal Arjun” emerges as a conduit bridging the gap between tradition and modernity. While it is a notable fact that there aren’t many games that are based on the epic ‘Mahabharata’, this literature review critically examines the conceptualization and significance of ”Baal Arjun” within the contemporary gaming landscape.

4.1 Harmonizing Cultural Legacy with Technological Ingenuity

In an era marked by technological advancement and cultural evolution, ”Baal Arjun” masterfully harmonizes these seemingly disparate elements. By effortlessly infusing the theme of the Mahabharata, the game offers players an immersive experience that resonates with our profound Hindu heritage. This synthesis of cultural motifs underscores the visionary approach of ”Baal Arjun,” aligning with the insights of various authors, who champion digital platforms as potent vehicles for cultural preservation.

4.2 Elevating Cultural Representation in Gaming

Distinctive in its approach, ”Baal Arjun” addresses a conspicuous gap in contemporary gaming – the nuanced representation of Hindu culture. In a milieu replete with diverse gaming themes, the game’s embrace of our cultural nuances is pioneering. Its innovative narrative resonates with the propositions that gaming platforms serve as conduits for cultural appreciation and understanding.

4.3 Cultivating Educational Exploration

Beyond its entertainment quotient, ”Baal Arjun” presents an edifying dimension, offering players a conduit for cultural learning. By immersing players in the Mahabharata’s essence and embodying Arjun’s valor, the game subtly imparts historical and mythological knowledge. This resonates with the educational potential of games underscored by Gee (2003)*, where games serve as interactive mediums for assimilating intricate narratives and honing problem-solving skills.

4.4 Navigating Challenges and Reaping Rewards

Arjun’s journey, interwoven with evasion and precision shooting, mirrors the rigors of real-life challenges. The gameplay is intrinsically aligned with Csikszentmihalyi’s ** concept of ”flow,” wherein challenges are synchronized with a player’s proficiency, engendering immersive experiences. The successful navigation of these challenges begets a sense of achievement, spurring players to persevere and excel.

4.5 A Paradigm Shift in Gaming

"Baal Arjun" transcends the confines of conventional gaming, assuming the role of a transformative force that amalgamates culture and technology. As it propels a digital resurgence of our cultural heritage, the game endeavors to captivate players while fostering cultural resonance. In a gaming milieu characterized by innovation and diversity, "Baal Arjun" aspires to carve a distinctive niche, poised at the crossroads of tradition and the digital age.

* - "Gee (2003)" refers to the real scholar James Paul Gee and his work "What Video Games Have to Teach Us About Learning and Literacy" published in 2003. In the context of the literature review, Gee's ideas are referenced to support the concept that games can serve as interactive mediums for education and learning. His work is often cited in discussions about the educational potential of video games

** - Mihaly Csikszentmihalyi is a psychologist known for his work on the concept of "flow." Flow is a state of complete immersion and engagement in an activity, where challenges are balanced with one's skill level, resulting in a highly focused and enjoyable experience.

In conclusion, "Baal Arjun" symbolizes a unique symbiosis between cultural heritage and technological prowess. By offering an interactive platform steeped in Hindu traditions, the game endeavors to enthrall players while fostering cultural reverberations. Within a dynamic gaming landscape characterized by ceaseless evolution, "Baal Arjun" assumes an unprecedented role, standing as a trailblazer in the realm of cultural gaming exploration.

5 METHODOLOGY

To complete our project, we followed the following methods:

5.1 Research and Planning

1. Conduct research on various aspects of game development, including features, fundamental requirements, and tools required.
2. Familiarize ourselves with the necessary libraries and divide the work among the team members.
3. Develop comprehensive coding guidelines to ensure self-documenting code.

5.2 Algorithm Design

1. Design the algorithm and create a flowchart based on the gathered rules and information.
2. Develop a basic prototype for testing purposes, using the console.

5.3 Sources

Utilize the following sources as references to support our project:

1. Documentation for SFML.
2. Concepts of object-oriented programming (OOP) and C++ programming from college courses, online references, books, etc.

5.4 Game Design

1. Implement the code using an object-oriented programming (OOP) paradigm.
2. Use Simple and Fast Multimedia Library to create build the game.
3. Choose Visual Studio as our Integrated Development Environment (IDE) and visual c++ as the compiler in the Windows Operating System.
4. Manage our code using GitHub for version control.

5.5 Maintenance

Regularly maintain the project by following these procedures:

1. Corrective maintenance:
 - Compile and check specific parts of the code for potential bugs and errors, fixing them as necessary.
2. Preventive maintenance:

- Identify errors and take measures to avoid them during runtime and the development cycle.
- Comment and document the source code thoroughly and back it up to prevent potential data loss.

5.6 Testing and Debugging

1. Develop a minimum viable product sample for testing and debugging purposes.
2. Conduct frequent testing and debugging to add features and make necessary edits to the project code based on our requirements and capabilities.

5.7 Documentation

1. Ensure comprehensive documentation of the app, including the addition of copyright licenses.
2. Ship the completed app with the accompanying documentation.

6 IMPLEMENTATION

6.1 Approach

The implementation of our project was structured into several distinct phases, each centered around specific gameplay mechanics, visual design, and user interaction. This approach facilitated a continuous process of refinement and enhancement as we progressed.

6.1.1 Tools and Technologies

1. Programming Languages We selected C++ as the primary programming language for our game's development. Its exceptional speed and object-oriented nature significantly streamlined the process of coding the game's mechanics.
2. Libraries For multimedia and graphics user interface implementation, we harnessed the capabilities of the SFML library. Chosen for its user-friendly syntax and comprehensive toolkit, SFML provided us with all the essential resources required for effective game development.

6.1.2 System Architecture and Design

The architecture of "Baal Arjun" was thoughtfully structured to ensure both flexibility and scalability. Key components encompassed in this architecture include:

1. Player Controller: Responsible for managing player inputs, character movement, and shooting mechanics.
2. Enemy Automation: Incorporating dynamic enemy movement and arrow-shooting behavior.
3. Collision Detection: Overseeing accurate collision detection between arrows, characters, and the game environment.
4. Level Implementation: Designing and integrating game levels to ensure a dynamic player experience.
5. User Interface (UI): Displaying essential in-game information such as health, arrow count, and visual elements for enemies and player arrows.
6. Sound Implementation: Enriching gameplay through auditory elements.

6.1.3 Code Development

The foundational gameplay mechanics were meticulously developed, spanning aspects like player movement, arrow physics, enemy behavior, shooting mechanisms, collision detection, and the implementation of diverse game levels. Algorithmic solutions were devised for trajectories of normal and Sudarshan arrows, each rigorously tested and refined. Emphasis was placed on crafting self-documented code, achieving modularity for code reusability.

6.1.4 Resource Collection

Resources such as images, sounds, and fonts were thoughtfully curated from online sources. Furthermore, Adobe Illustrator was leveraged to modify these resources to align precisely with our project's creative vision.

6.1.5 Integration and Testing

The integration process commenced by combining shooting algorithms with collision detection algorithms, subjecting them to comprehensive testing to ensure their seamless functionality. Concurrently, the graphical user interface (GUI) underwent development. Eventually, the GUI and game logic converged into the final cohesive product. A testing phase was initiated, involving friends as testers who aided in identifying and rectifying any bugs encountered.

6.1.6 Version Control and Collaboration

To facilitate seamless collaboration, GitHub served as our chosen version control system. Regular code synchronization was achieved through Git, ensuring that contributions from all team members were harmoniously integrated.

This comprehensive implementation phase not only actualized our project but also exemplified our commitment to precision, collaboration, and the realization of an engaging gaming experience.

6.2 Flow Chart

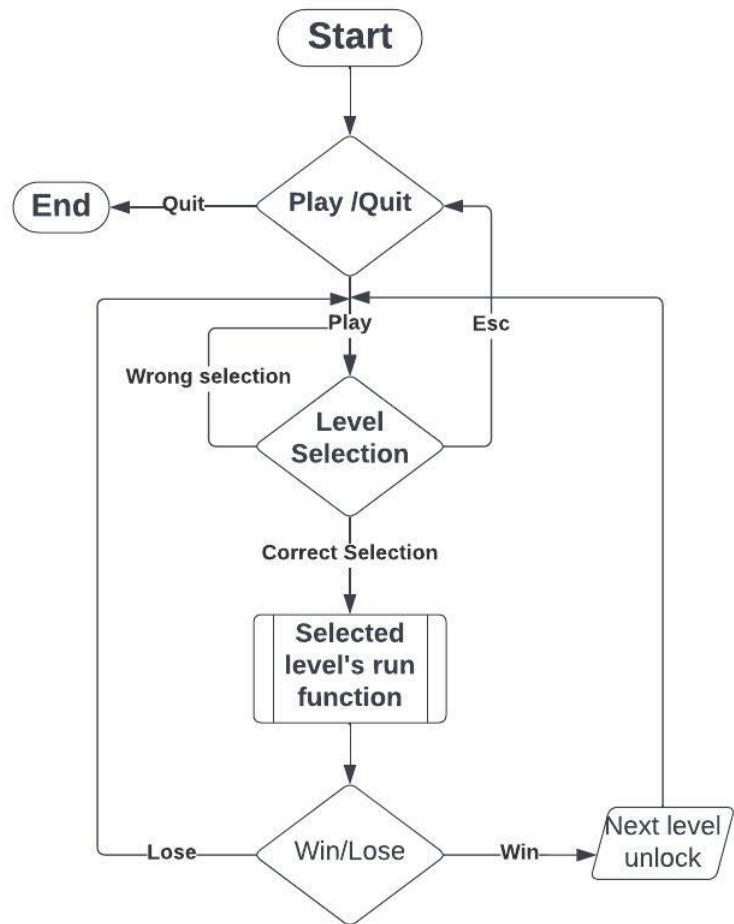


Figure 2: Flow Chart

6.3 Block Diagram

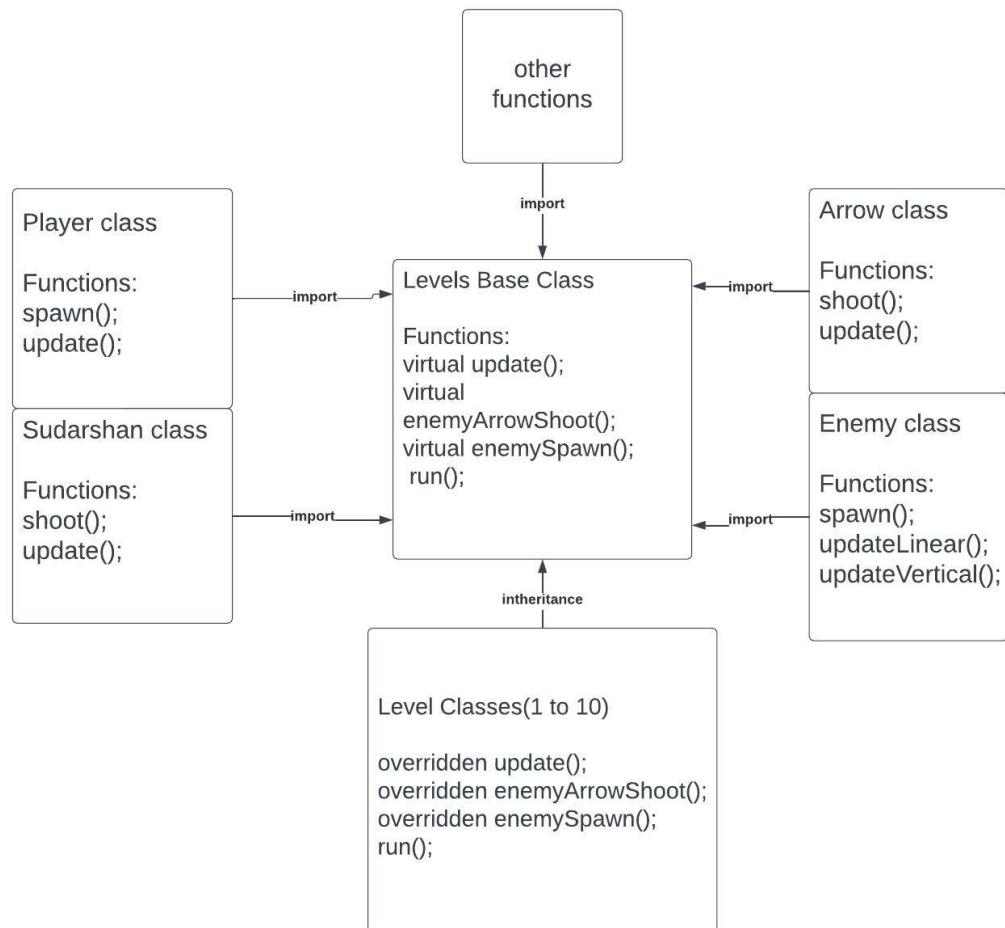


Figure 3: Generalized Block Diagram

6.4 Game Pictures

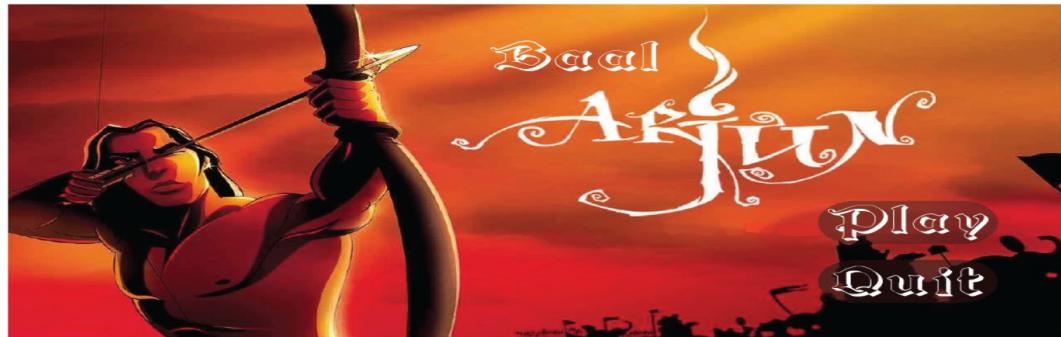


Figure 4: Main Screen

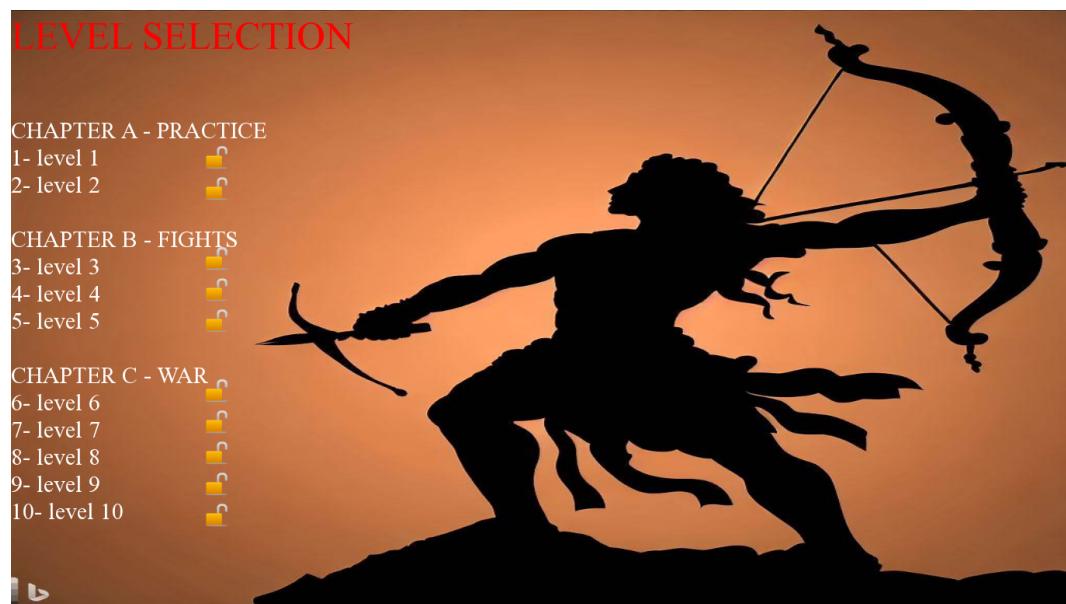


Figure 5: Level Selection Screen

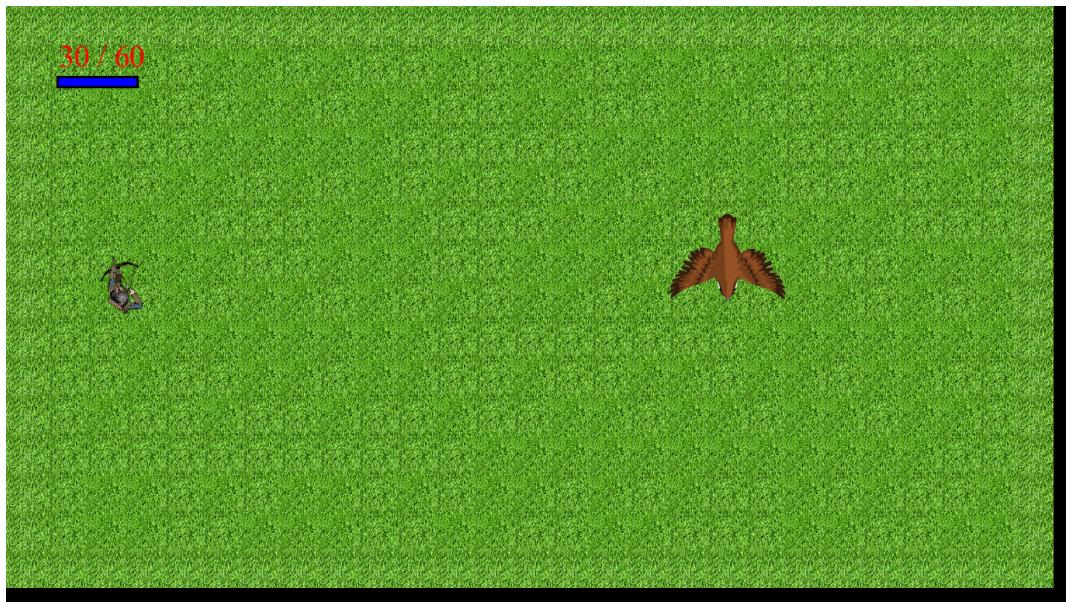


Figure 6: Level 2



Figure 7: Level 7



Figure 8: Generalized Block Diagram

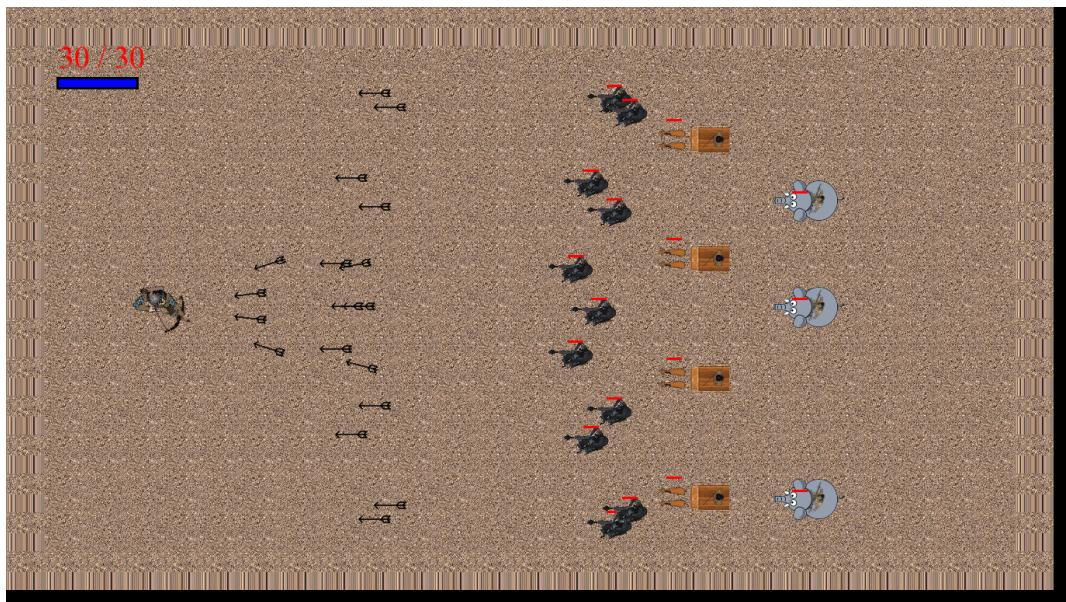


Figure 9: Generalized Block Diagram

7 RESULTS

The implementation of Object Oriented Programming concept using C++, the development of GUI environment employing the Safe and Fast Multimedia Library (SFML), the application of mathematical formulas and algorithm to ensure seamless and glitchfree gaming experience and the contionus efforts involved in writing, debugging and testing the program has resulted in a functional product, 'Baal Arjun' - a 2D arrow shooting game. While always being openminded for future enhancements and constructive crticism, we believe , with the successful creation of this playable game, the objectives of this project have been duly achieved.

8 PROBLEM FACED AND SOLUTIONS

Problem Faced and solution:

1. Code Manageability

The use of Object Oriented programming helped us in dividing the code into multiple parts(header and cpp files) and manageable chunks. Navigating between certain parts of the code often and finding specific lines were tedious due to the sheer volume of the code spread throughout 25+ header and cpp files.

This problem was solved by using naming the files, classes, objects, variables and functions with relevant names. The naming was greatly assisted by SFML library declared variable and function names. Proper Comments were mentioned in each and every section of code which in turn, increased code readability as well as facilitated code manageability.

2. Arrow Overriding

While we were satisfied by SFML library defined functions, one of the unexpected problems faced was - whenever a new arrow was shot by enemy, the previous arrow simply disappeared from screen. This ensured undue immunity to player which contradicted the motive of the game.

The solution entailed passing the count of arrows using a "pass by reference" mechanism, thereby enabling accurate tracking and management of all arrows within the game environment.

3. Debugging and Updating

A true developer knows how much debugging and testing goes on, even while writing a relatively simple code. As such, we were not immune to this phenomena, specially in each level while setting the attributes of enemy ie , the position of every enemies and their speed. Similar problem was faced while writing the level selection and play/quit code.

This problem was solved by intensively following hit and trial method. All of the code was rigorously tested so as to minimize errors and create smooth and seamless gaming experience.

4. Lack of Appropriate Images

Since the visual orientation of the game was top view, there were not a lot of images that followed the rather unconventional view. Most images we found were either front view or angular view, but as we choose top down view due to its benefit that both x and y axis along with rotation could be involved, the overall graphical aesthetics of the game was compromised. Even after spending a lot of time searching images such as (warrior from top view, horse carriage from top view, etc.) and free tokens in AI image generators, the results were not satisfactory.

This problem was solved by selecting the image which best fit our requirement and image manipulation (such as combining top view of horses, top view of empty carriage , and top view of a standing warrior to make top view of a warrior in a horse carriage).

9 LIMITATIONS AND FUTURE ENHANCEMENTS

9.1 Limitations

1. The game doesn't show the player's score and highest achieved score.
2. Both easy and hard levels have the same number of arrows, which might not suit the difficulty level.
3. The game lacks a coherent storytelling and instructional guide for players.
4. There are no shields or protective mechanisms for the player-character.
5. The game lacks methods to restore the player's health or replenish arrows during gameplay.
6. The absence of a comprehensive reward system, such as earning coins.
7. The game lacks improvements or upgrades for the player's bow and arrows.
8. The graphical user interface (GUI) is not fully immersive

9.2 Future Enhancements

1. New types of arrows: Additional arrow varieties could be introduced, enhancing gameplay diversity and strategy.
2. Graphics improvement: Enhancements to the game's visual elements could lead to a more immersive and appealing user experience.
3. More difficult levels: The incorporation of increasingly challenging levels would provide players with greater engagement and progression challenges.
4. Avatar customization: The addition of avatar customization options would allow players to personalize their in-game characters.
5. Health increment mechanism: Implementing a mechanism to increase in-game health could extend gameplay and strategic options. This could be done by placing health boosters at random positions on the gamescreen.
6. Reward system: A structured reward system could motivate players by recognizing their accomplishments and fostering a sense of achievement. This may be achieved by rewarding the player with some coins, which can be used to restore in-game health, and to purchase new and more powerful bows and arrows.
7. Shield feature: Introducing a shield mechanic could introduce a new layer of gameplay tactics and defensive strategies. This could also be done by placing shield enablers at random position in player's arena, which activates shield for a limited time period.
8. Enriched storytelling: Enhancing the narrative elements could deepen player immersion and emotional investment in the game's world using voice records or AI generated voice of the story.

10 CONCLUSION AND RECOMMENDATIONS

In conclusion, the development of "Baal Arjun," a 2D arrow shooting game, an OOP in C++ project using the SFML library, has been a rewarding journey for our team. This project has not only resulted in a functional game but has also significantly enriched our collaborative and teamwork skills. The process of designing, coding, and refining the game has brought us together and taught us the importance of effective communication and coordinated effort in achieving a shared goal.

Throughout this project, we gained valuable insights into Object-Oriented Programming principles, honed our proficiency in the C++ programming language, and deepened our understanding of game development through the practical application of SFML. Beyond the technical aspects, we also acquired proficiency in utilizing essential industry tools such as GitHub for version control and Lucidchart for visual documentation, equipping us with practical skills highly relevant to the software development field.

As we move forward, we recommend that future enhancements for "Baal Arjun" include the incorporation of diverse arrow types, improved graphics, additional difficulty levels, personalized avatars, health progression mechanisms, a rewarding system for player achievements, shield functionalities, and a more immersive storytelling experience. These advancements will not only enhance the gameplay but also provide opportunities for continued learning and growth.

In essence, the "Baal Arjun" project has provided us with a comprehensive understanding of software development principles, effective collaboration, and utilization of industry tools, which will undoubtedly serve as a solid foundation for our future endeavors in the field of technology and game development.

11 REFERENCES

- [1] Baral, D S.(2018).Secrets of Object Oriented Programming in C++
- [2] <https://www.sfml-dev.org/documentation/2.6.0/>
- [3] Stroustrup, B.(2013).The C++ Programming Language, 4th edition