



BSIT-F24-002

03-135211-027 Abdullah Mehboob

03-135211-005 Ali Hassan

Wave Fury: Race of Titans

In partial fulfilment of the requirements for the degree of
Bachelor of Science in Information Technology

Supervisor: SUMMAIRA NOSHEEN

Department of Computer Sciences
Bahria University, Lahore Campus

June 2025

Certificate



We accept the work contained in the report titled

“Wave Fury: Race of Titans”

written by

ABDULLAH MEHBOOB

ALI HASSAN

as a confirmation to the required standard for the partial fulfilment of the degree of
Bachelor of Science in Information Technology.

Approved by:

Supervisor: SUMMAIRA NOSHEEN

(Signature)

June 05, 2025

DECLARATION

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

Enrolment	Name	Signature
03-135211-027	ABDULLAH MEHBOOB	
03-135211-005	ALI HASSAN	

Date : June 05, 2025

Specially dedicated to

My beloved grandmother, mother and father and above all supervisor

(ABDULLAH MEHBOOB)

My beloved mother and other family members, BULC faculty and supervisor

(ALI HASSAN)

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express our gratitude to my research supervisor, Ms. SUMMAIRA NOSHEEN for her invaluable advice, guidance and her enormous patience throughout the development of the research.

In addition, we would also like to express my gratitude to our loving parent and friends who had helped and given us encouragement.

ABDULLAH MEHBOOB
ALI HASSAN

Wave Fury: Race of Titans

ABSTRACT

The "Wave Fury: Race of Titans," is the work on a single-player boat racing game designed to entertain and challenge players with fast-paced gameplay and dynamic sea environments. This game offers players an immersive experience where they race a single boat against AI-controlled opponents. Set across three stages with increasing difficulty, the game introduces hurdles both stationary and movable that test players' reflexes and strategy.

The game is built in Unity, using C# for scripting player controls, AI behavior, and physics-based interactions such as boat movement and collisions. The boats, environment, and hurdles are designed in Blender, with UI and textures created in Adobe Photoshop. The AI opponents are programmed to navigate the course dynamically, adjusting to the player's actions, while each stage features distinct challenges. Set entirely in day mode, the bright and visually engaging sea environment enhances the racing experience.

The game incorporates electromyography (EMG) sensors to allow players to control the speed of the boat through muscle contractions, adding a unique and interactive layer to the gameplay. The game delivers a boat racing adventure that combines realistic visuals, AI opponents, and hurdles to create an immersive gameplay experience.

TABLE OF CONTENTS

DECLARATION	ii
ACKNOWLEDGEMENTS	iv
ABSTRACT	v
TABLE OF CONTENTS	vi
LIST OF TABLES	ix
LIST OF FIGURES	x
LIST OF SYMBOLS / ABBREVIATIONS	xi
LIST OF APPENDICES	xii

CHAPTERS

1	INTRODUCTION	1
	1.1 Background	1
	1.2 Problem Statements	2
	1.3 Aims and Objectives	2
	1.4 Scope of Project	2
2	LITERATURE REVIEW	3
	2.1 Proposed System	3
	2.2 Related work	3
	2.3 Technologies Used	3
	2.4 Non-functional requirements	4
	2.4.1 Performance Requirements	4
	2.4.2 Security Requirements	4
	2.4.3 Supportability	4

2.5	Functional requirements	5
2.6	Hardware specs	5
2.6.1	Android	5
2.6.2	Laptop	6
2.6.3	EMG Sensor	6
2.6.4	Arduino UNO	7
3	DESIGN AND METHODOLOGY	8
3.1	Main Component	8
3.2	Android Version	8
3.3	Feature-Driven Development (FDD)	8
3.3.1	Develop an overall model	8
3.3.2	Built a feature list	8
3.3.3	Plan by feature	9
3.3.4	Design by feature	9
3.3.5	Build by feature	9
3.1	Work Flow Diagram	11
3.2	Use Case Diagram	12
3.2.1	Use Case Description	13
3.3	Class diagram	14
4	DATA AND EXPERIMENTS	15
4.1	Implementation	15
4.1	EMG Sensor working	16
4.2	Game Physics	17
4.3	Game Design	17
5	RESULTS AND DISCUSSIONS	18
5.1	Main Screen	18
5.2	Stages Screen	18
5.2.1	Level 1	19
5.2.2	Level 2	20
5.2.3	Level 3	21

REFERENCES

23

APPENDICES

24

LIST OF TABLES

TABLE	TITLE	PAGE
	Table 2.1: Functional Requirements	5
	Table 3.1: Use Case Description	13

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 2.1:	Android 8.0	6
Figure 2.2:	EMG Sensor	6
Figure 2.3:	Arduino UNO	7
Figure 3.1:	Feature Driven Development (FDD)	10
Figure 3.2:	Work Flow Diagram	11
Figure 3.3:	Use Case Diagram	12
Figure 3.4:	Class diagram	14
Figure 4.1:	Wearable (Sensor on arm)	15
Figure 4.2:	Wearable Setup	16
Figure 5.1:	Main Screen	18
Figure 5.2:	Levels Screen	19
Figure 5.3:	Level 1 (Start Interface)	19
Figure 5.4:	Level 1 (Map)	19
Figure 5.5:	Level 1 (Map)	20
Figure 5.6:	Level 2 (Start)	20
Figure 5.7:	Level 2 (Map)	21
Figure 5.8:	Level 3 (Start)	21
Figure 5.9:	Level 3 (Map & Hurdles)	22

LIST OF SYMBOLS / ABBREVIATIONS

<i>EMG</i>	Electromyography, Sensor
<i>AI</i>	Artificial Intelligence
<i>UI</i>	User Interface
<i>FPS</i>	Frames per Second
<i>APK</i>	Android Package Kit
<i>PWM</i>	Pulse Width Modulation
<i>AVR</i>	Advanced RISC Virtual, Microcontroller 8-bit
<i>MA</i>	Mille Ampere
<i>V</i>	Volt
<i>MHz</i>	Mega Hertz

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
	APPENDIX A: User Manual	25

CHAPTER 1

INTRODUCTION

The "Wave Fury: Race of Titans," is a single-player boat racing game designed to entertain and challenge players with fast-paced gameplay and dynamic sea environments. It has AI controlled opponents which make the game more thrilling and exciting. The game has three stages with increasing difficulty, the game introduces hurdles (both stationary and movable) that test players' reflexes and strategy.

The game is built in Unity for its advanced 3D and AI capabilities, using C# for scripting player controls, AI behaviour, and physics-based interactions such as boat movement and collisions. The boats, environment, and hurdles are designed in Blender, while UI and textures created in Adobe Photoshop. The AI opponents are programmed to navigate the course dynamically, adjusting to the player's actions, while each stage features distinct challenges with different hurdles. Set entirely in day mode, the bright and visually engaging sea environment enhances the racing experience. The game incorporates electromyography (EMG) sensors to allow players to control the speed of boat through muscle contractions, adding a unique experience of gameplay. If boat hits the hurdles (static or dynamic), then boat will be dived and "YOU LOSE". If boat does not reach on destination before the other boats on time, message displayed "YOU LOSE" on the screen. In other condition, if the player survived and reach the destination before the other players he will get rewarded with coins, and "YOU WIN" message pop up on the splash screen. There is resume and restart option in the game.

1.1 Background

We decided to make "Wave Fury: Race of Titans" because games are a great way to help people relax, have fun, and reduce stress. Many people enjoy racing games, and we wanted to create something new and exciting by using sensor. Our goal was to

make a game that is not only fun but also uses smart technology to give a better experience. We included AI opponents that react to the player's actions and added EMG sensors to control the speed of boat using muscle movement, which makes the game more exciting. This project helps us learn about game development, AI, and using new input methods, while giving players a fun and unique way to enjoy boat racing.

1.2 Problem Statements

The problem is that most boat racing games feel repetitive and don't offer much excitement or challenge. Players want something more fun, smart, and realistic. So, we aim to create a thrilling boat racing game that is full of action, with smart AI opponents, realistic boat movements, and harder levels with different hurdles. Our goal is to give players a powerful, fun, and unforgettable racing experience.

1.3 Aims and Objectives

The objectives of the thesis are shown as following:

- i. To entertain users and improve their cognitive skills
- ii. To ensure smooth performance and enhanced gaming experience
- iii. Realistic boat movement, collisions and water dynamics
- iv. To develop high-quality UI elements for better user experience

1.4 Scope of Project

The primary scope of this project is to develop a single-player boat racing game. The game will have realistic boat movements using Unity's physics system to make the racing feel more real. It will also include sensor-based controls, so players can control the boat using muscle movements for a unique gaming experience. Smart AI opponents will be added to make the game more competitive by reacting to the player's actions.

CHAPTER 2

LITERATURE REVIEW

2.1 Proposed System

The current systems that exist are supported for higher versions of android. Our game can support all these versions but it will also work in operating environment which include minimum android version of at least 8.0.

Our system is an exciting single player boat racing game which is made to give players a thrilling experience with obstacles and AI opponents. The game has three levels which include moving and static hurdles. The player controls the boat using either screen touch gestures or EMG sensor.

2.2 Related work

Other boat racing games offer fast racing and exciting tracks but most of them use only basic controls like touch or keyboards. Many boat racing games do not use sensor's input or offer dynamic hurdles. But our game include EMG sensor-based speed controls which makes the game more exciting. Moreover, the game has both stationary and moving hurdles that add higher level of challenges.

2.3 Technologies Used

The game is built in Unity using C# for coding related to AI behaviours, players controls and physics-based interactions. Blender is used to design the boats, environment, and water and hurdles while Photoshop is used to make game's UI and textures. The game has daytime environment that looks more real and players feel

more comfortable. The EMG sensor is used so that players can control the speed of boat by their muscle movement.

2.4 Non-functional requirements

2.4.1 Performance Requirements

The game should run smoothly at 30-60 FPS with fast response. AI opponents act without delay and interface must be simple with clear options like resume and restart. Win or lose messages should appear at the end of each level.

2.4.2 Security Requirements

There is no need of any security in my project because it is an offline game. There is no sensitive data that needs to be protected and data must be stored in user's device. So, there will be no need of security requirements in this project.

2.4.3 Supportability

The application must be platform dependent, i.e. it should be able to run on platform supporting android version of 8.0 or above and device with minimum of 4GB RAM.

2.5 Functional requirements

Table 2.1: Functional Requirements

ID	Requirement	Description
1	Functional	The game should include a main menu, restart, and resume options.
2	Functional	The game must allow the player to control boat using touch screen or EMG sensor.
3	Functional	The game must include AI opponents that play against the player.
4	Functional	The game has three levels, each with different types of static and dynamic hurdles.
5	Functional	The game must detect collisions between the boat and hurdles and respond to it.
6	Functional	The game displays “YOU WIN” message if the player wins the game.
7	Functional	The game displays “YOU LOSE” message if the boat hits a hurdle or finishes last.

2.6 Hardware specs

2.6.1 Android

The minimum android version required is 8.0. After developing and switch to android platform and build an APK on unity, the game will work well with minimum 8.0 version.



Figure 2.1: Android 8.0

2.6.2 Laptop

We used laptop with minimum specifications of Intel Core i7 8th Generation processor, 16 GB RAM, 512 GB hard drive, Windows 10 (64-bit). It is used for Used for game development in Unity, coding in C#, designing 3D models in Blender, and creating UI in Adobe Photoshop.

2.6.3 EMG Sensor

Operating Voltage: $\pm 3.5V$

Output Signal: Analog voltage ranging from 0V to the supply voltage (Vs), corresponding to the level of muscle activity.

Audio-style Plug: 3.5 mm jack for easy connection.

Electrode Pad Diameter: 52 mm, suitable for standard EMG applications.

Cable Length: Approximately 2 feet, providing flexibility in placement.

Weight: 30 grams, making it lightweight and portable.

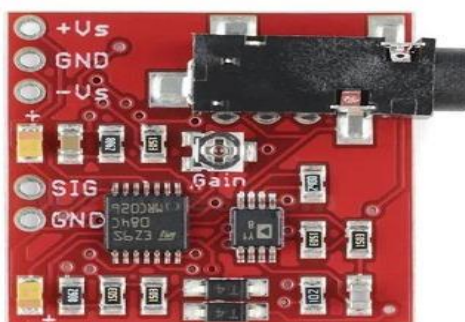


Figure 2.2: EMG Sensor

2.6.4 Arduino UNO

Microcontroller: ATmega328P (8-bit AVR)

USB-to-Serial Converter: ATmega16U2, replacing the older ATmega8U2 for faster data transfer and more memory

Operating Voltage: 5V

Input Voltage (recommended): 7–12V

Input Voltage (limits): 6–20V

Digital I/O Pins: 14 (6 provide PWM output)

Analog Input Pins: 6

DC Current per I/O Pin: 40 mA

Flash Memory: 32 KB (0.5 Kb used by boot loader)

Clock speed: 16 MHz



Figure 2.3: Arduino UNO

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Main Component

The EMG sensor is the main component of this game:

- EMG Sensor with electrode Pads
- Arduino UNO
- Mobile

3.2 Android Version

The minimum Android version is 8.0 as supporting the EMG sensor.

3.3 Feature-Driven Development (FDD)

3.3.1 Develop an overall model

We developed overall model for the game Wave Fury: Race of Titans which is single player boat racing game. It includes three stages in which player race against AI controlled opponents in a dynamic sea environment with different hurdles. We also used EMG sensor to control the speed of boat which makes it more unique.

3.3.2 Built a feature list

The features planned for implementation in the game are:

- AI opponents
- Boat movement using EMG sensor
- Stationary and moveable hurdles
- UI elements for:
 - Main screen (Play, Exit, Settings, Scoreboard)
 - Stage selection screen
 - Win/Lose screens
- Collision detection of boats with hurdles
- Day mode sea environment

3.3.3 Plan by feature

We started planning the development of each feature step by step. First we started planning the design of boat, sea environment and hurdles using blender. Then we focused on EMG sensor integration to control the movement of boat. We planned to add different hurdles, both stationary and moveable to make the game complex. The game has three stages with increasing difficulty and different UI elements.

3.3.4 Design by feature

First EMG sensor would be designed that how it will control boat through muscle movement. Then we designed the AI opponents to follow the path and avoid hurdles. Then different hurdles and levels were designed with increasing difficulty to make game interesting.

3.3.5 Build by feature

We started developing each feature in Unity using C#. EMG sensor is integrated for the movement of boat through muscle movement. Then we built the AI boats with a logic so that they follow the track and avoid obstacles. We added both stationary and

moveable hurdles and applied collision detection of boat with hurdles. We also added UI elements in the game for better experience using Photoshop.

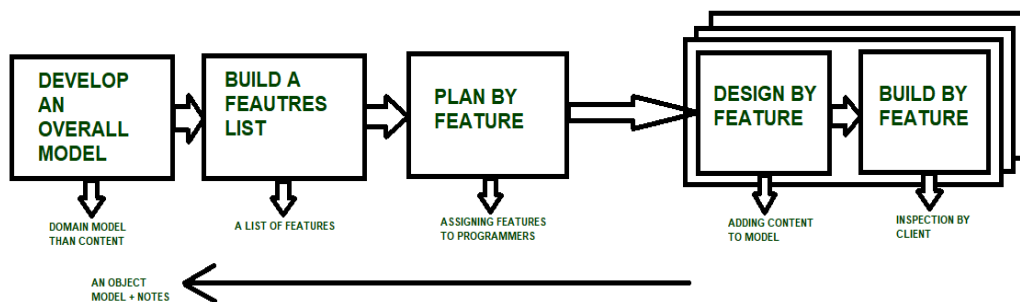


Figure 3.1: Feature Driven Development (FDD)

3.1 Work Flow Diagram

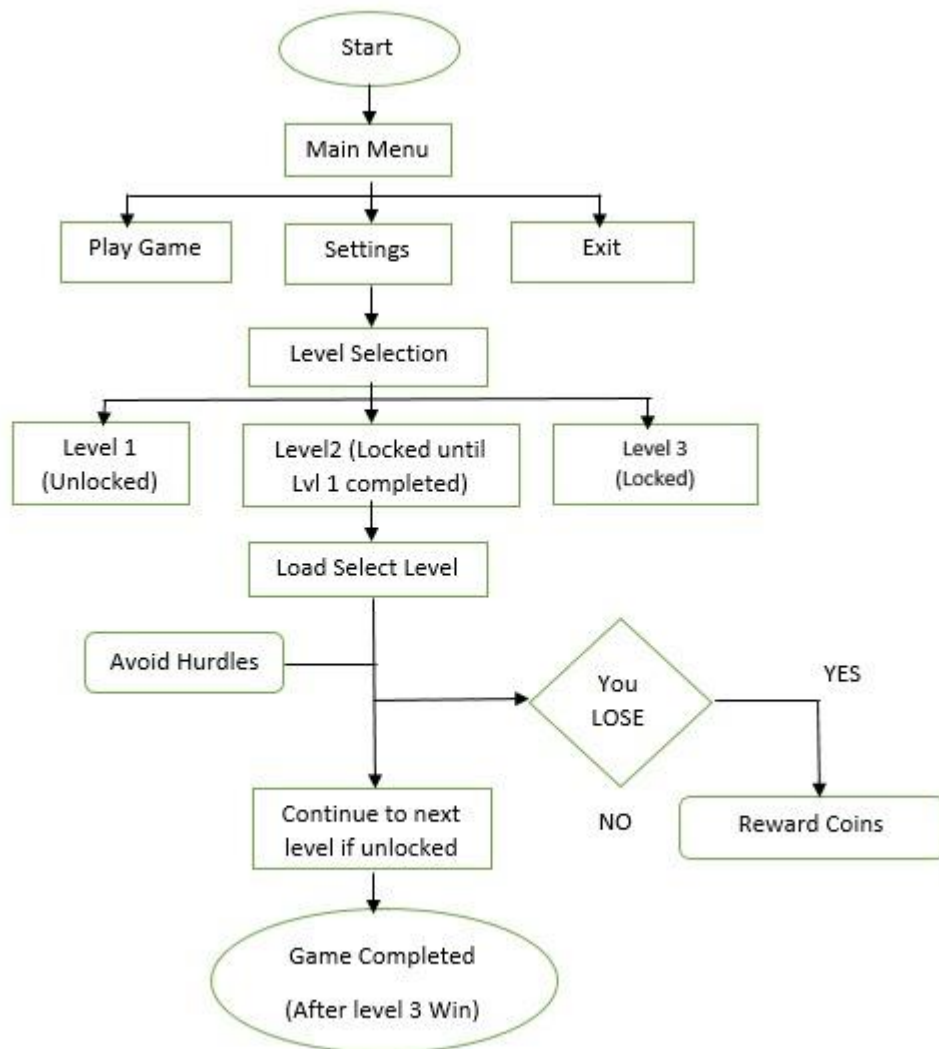


Figure 3.2: Work Flow Diagram

3.2 Use Case Diagram

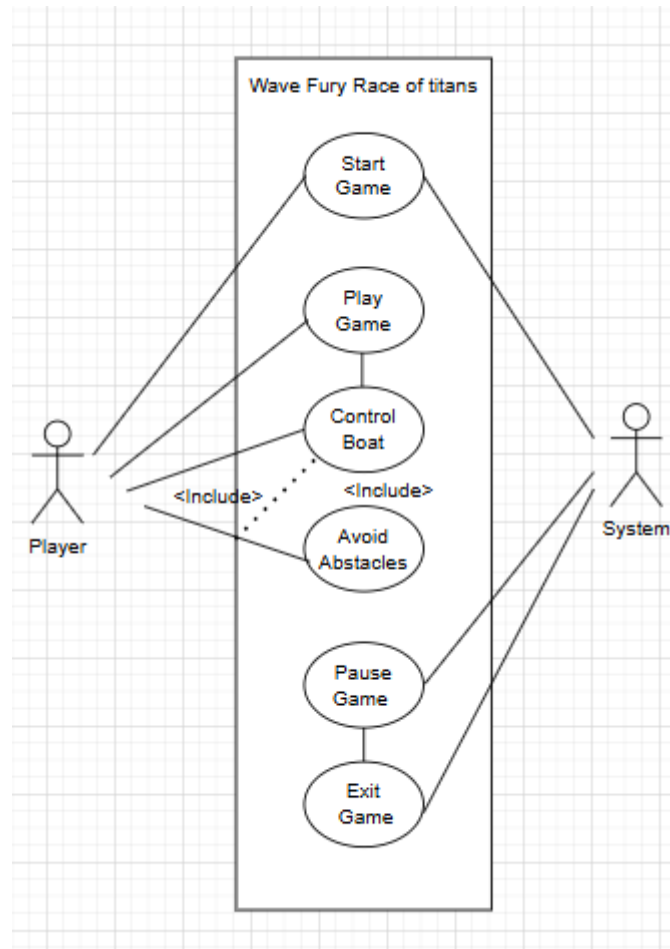


Figure 3.3: Use Case Diagram

3.2.1 Use Case Description

Table 3.1: Use Case Description

Use Case ID	Name	Primary Actor	Other Actors	Description	Trigger	Typical Flow	Post Condition
U1	Start Game	Player	System	Player initiates the game.	Clicks "Play" on main screen	Player clicks Play → Volume Option displayed → System navigates to home screen	Player enters the game
U2	Play Game	Player	None	Player starts the game after entering the home screen.	Clicks "Start" button	Player clicks & System loads the gameplay scene	Gameplay scene is launched
U3	Control Boat	Player	System	Player controls the boat via EMG or screen gestures.	Player interacts with control system	Player moves left/right/forward/back → Boat responds accordingly	Boat responds as per controls
U4	Avoid hurdles	Player	System	Boat must avoid all hurdles during race.	Boat approaches obstacle	Detects obstacle → Player steers to avoid → Boat continues path	Boat continues path or suffers consequence
U5	Pause Game	Player	System	Game is paused during play.	Presses "Pause"	Gameplay pauses → Menu shows resume/exit options	Game remains paused until resumed or exited
U6	Exit Game	Player	System	Player exits the game via pause menu or main screen.	Presses "Exit"	From pause menu or main screen → Confirm exit	Game closes

3.3 Class diagram

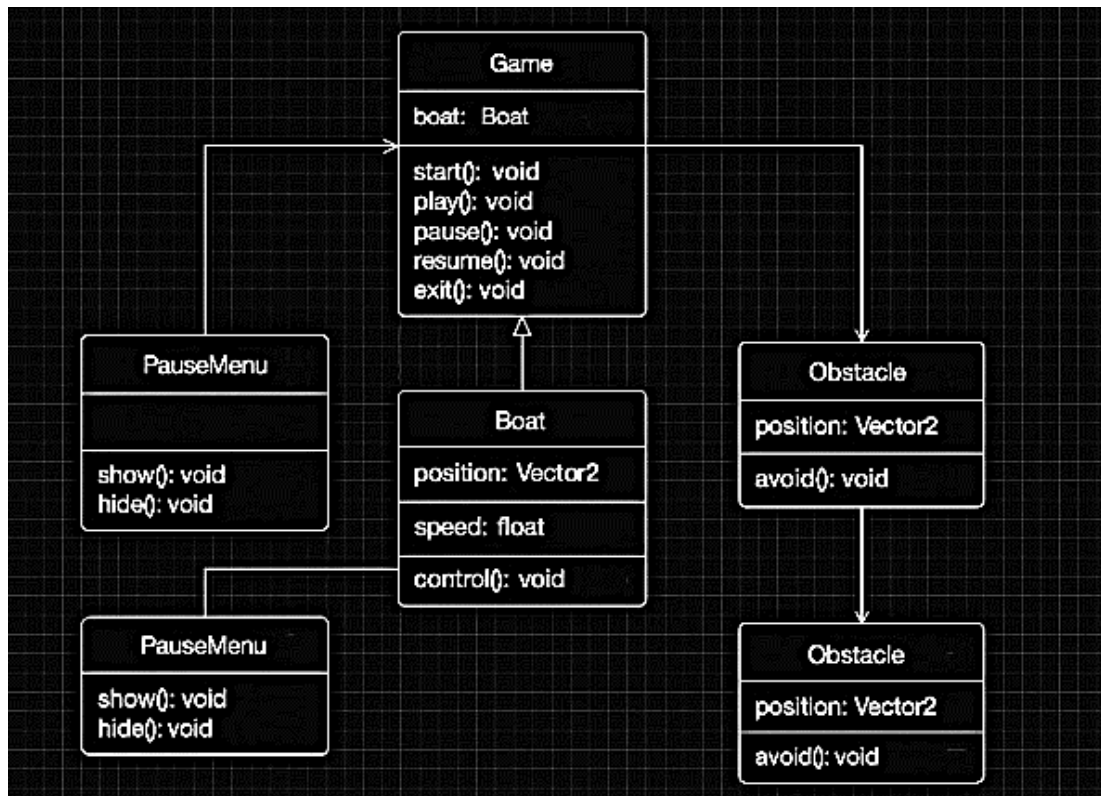


Figure 3.4: Class diagram

CHAPTER 4

DATA AND EXPERIMENTS

4.1 Implementation

The Wave Fury: Race of Titans game is the combination of software and hardware (EMG sensor and Arduino UNO). We are using Feature Driven Development (FDD) methodology to ensure deadline and goals will meet on time.

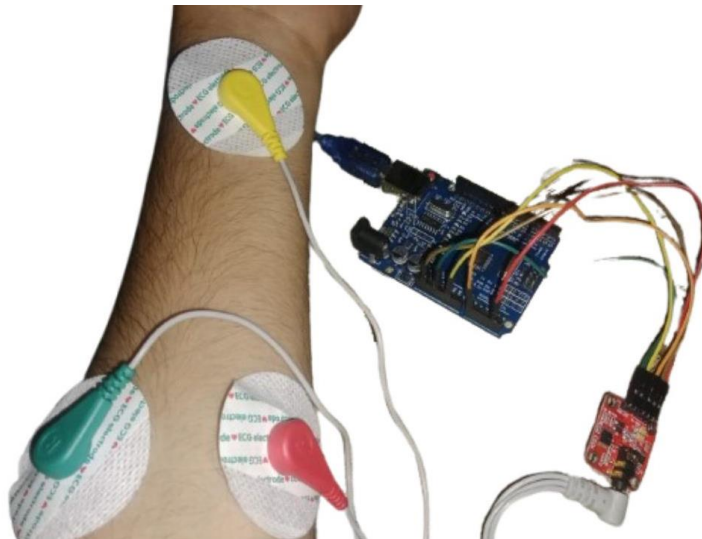


Figure 4.1: Wearable (Sensor on arm)

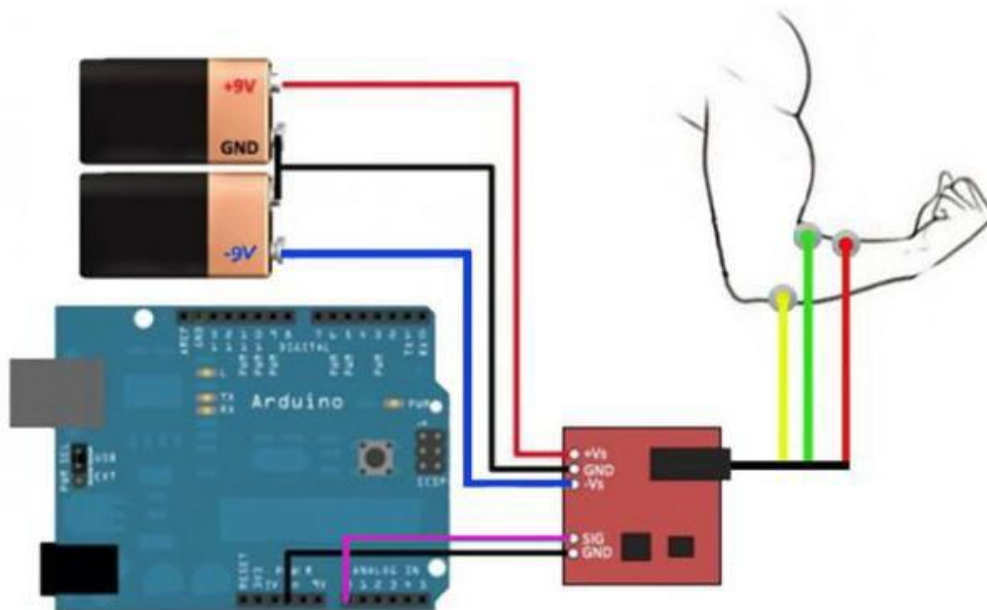


Figure 4.2: Wearable Setup

4.1 EMG Sensor working

Electrodes are positioned on the arm to allow the EMG (Electromyography) sensor to detect electrical signals from muscle contractions. An analog voltage signal is produced by the sensor when the player contract their muscle. The Arduino UNO receives and handles this signal before sending it to the Unity game. In the game, stronger muscle flexing boosts the boat's speed. The sensor is powered by two 9V batteries to provide precise readings. With this configuration, players may manipulate the boat's pace with their muscles in real time, making for an engaging game experience.

4.2 Game Physics

The game is totally based on the game physics which were implemented in the project to manage the 3D object environmental behaviour so that the behaviour of 3D object will be changed whenever the player interact with the environment or boats or hurdles.

4.3 Game Design

We have designed the main screen, environment, boats, and hurdles for our project. Also we have designed the buttons like the Play, sound setting, music setting, Exit, and level selection etc.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 Main Screen

This is the main screen of the game. It includes Play button, Settings, EXIT and score (coins) board. Score board includes coins that are collected during game by collecting the rings.



Figure 5.1: Main Screen

5.2 Stages Screen

This is the stages screen of our game which includes 3 levels. In the beginning, level 1 will be opened and other levels will be opened after completing the previous one. It has Play ad Back buttons which are used to start the level or turning back to main screen.



Figure 5.2: Levels Screen

5.2.1 Level 1

This is level 1 of the game which includes start and finish point, 5 rewarded rings, 2 containers as hurdles, and 1 Rock as a hurdle. It also has 2 Boats (1 AI boat and 1 player boat).



Figure 5.3: Level 1 (Start Interface)

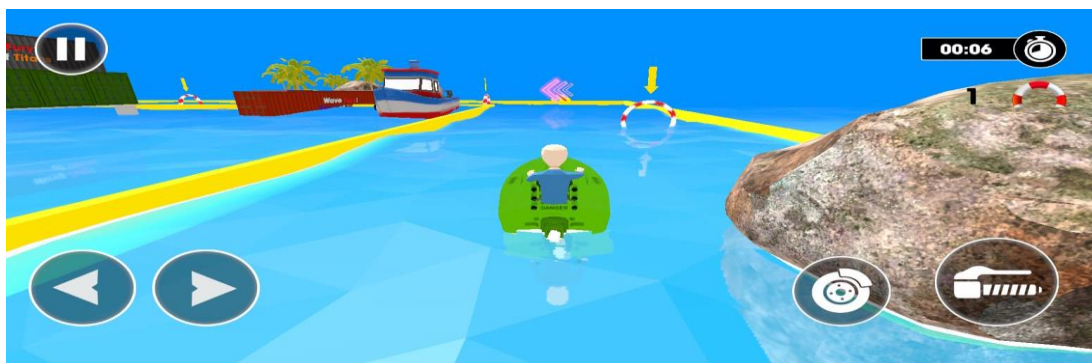


Figure 5.4: Level 1 (Map)

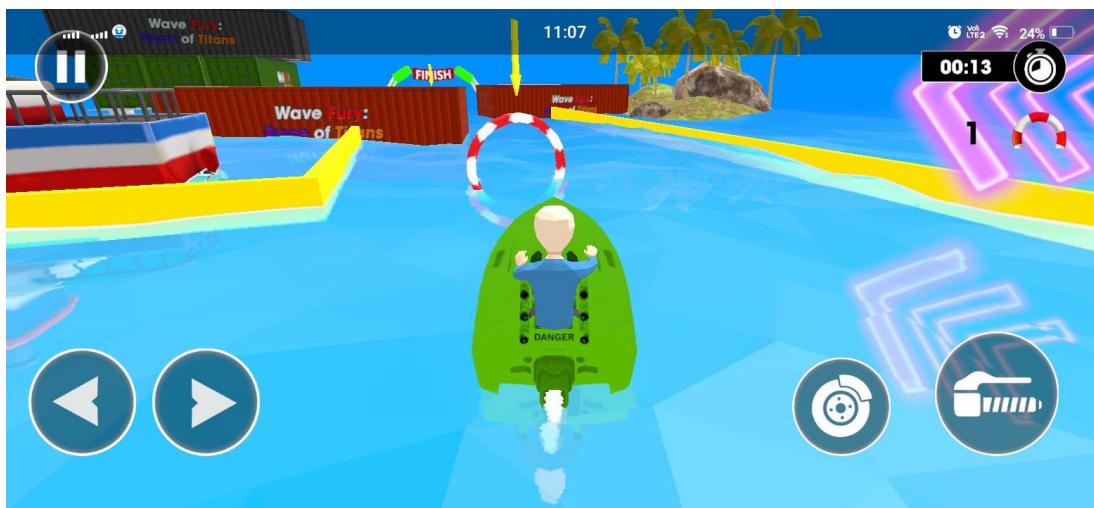


Figure 5.5: Level 1 (Map)

5.2.2 Level 2

Level 2 of the game is bit difficult than level 1 with increased complexity of hurdles. It includes start and finish points, 1 rocks as hurdles, 4 spinning wheels, 6 rewarded rings, and 2 containers as hurdles. It has 3 boats (2 AI boats and 1 player boat).



Figure 5.6: Level 2 (Start)



Figure 5.7: Level 2 (Map)

5.2.3 Level 3

Level 3 of the game is more difficult than level 2 because it contains more dynamic hurdles with increased complexity. It includes start and finish points, 2 rocks as hurdles, 5 spinning wheels, 11 rewarded rings, 1 ship, 4 spinning bar and 3 containers as hurdles. It has 3 boats (2 AI boats and 1 player boat).



Figure 5.8: Level 3 (Start)

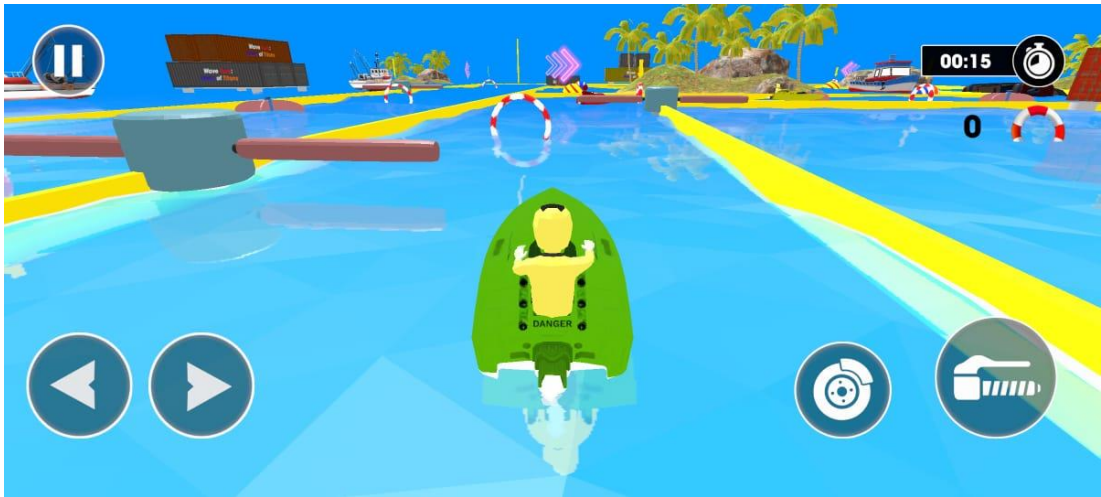


Figure 5.9: Level 3 (Map & Hurdles)

REFERENCES

Books:

- [1] Sue Blackman ,*Beginning 3D Game Development with Unity 4,2nd Edition*

Journal Papers:

- [2] Muguro, J., Laksono, P. W., Rahmaniar, W., & Njeri, P. W. (2021). Development of Surface EMG Game Control Interface. *Signals*, 2(4), 834-851.

- [3] van Toll, W. G., Cook IV, A. F., & Geraerts, R. (2012). A Navigation Mesh for Dynamic Environments. *Computer Animation and Virtual Worlds*, 23(6), 535-546.

<https://www.mdpi.com/2624-6120/2/4/48>

Tutorials:

- [4] AESMIT. (2021). Boat Design using Blender – Part 1 – An Infrastructure Journey. Retrieved from <https://aesmit.org/2021/07/01/how-to-design-a-27-foot-whale-boat-using-blender-part-1/>

- [5] Adobe. Photoshop Learn & Support. Retrieved May 1, 2025, from <https://helpx.adobe.com/support/photoshop.html?>

- [6] For buying and learning about EMG sensor <https://electrobes.com/product/emg-muscle-signal-sensor-module-kit-for-arduino/>

APPENDICES

APPENDIX A: User Manual

Game setup:

- To start the game player should press “PLAY” button.
- To EXIT the game player should press “EXIT” button.
- The game requires an Android device with a minimum supported OS version of Android 8.0.

Basic Game Rules:

- The player controls a boat and races against AI-controlled opponents.
- “YOU LOSE” message will be displayed if boat hit any hurdle or does not reach the finishing line before the AI opponents.
- If the player successfully avoids hurdles and finishes first, they receive reward coins and a “YOU WIN” message appears.
- The game supports EMG sensor input for speeding up the boat and touch controls for boat movement.
- The race includes static and dynamic hurdles.
- Game will be played in a daytime sea environment across three increasing difficulty stages.

race-of-Titans-Final.pdf

ORIGINALITY REPORT

11%

SIMILARITY INDEX

6%

INTERNET SOURCES

3%

PUBLICATIONS

8%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	2%
2	Submitted to Brunel University Student Paper	1%
3	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
4	www.elprocus.com Internet Source	1%
5	Submitted to Pathfinder Enterprises Student Paper	1%
6	Submitted to Laureate Higher Education Group Student Paper	1%
7	stanislavs.tripod.com Internet Source	<1%