



BUBT | BANGLADESH UNIVERSITY OF
BUSINESS AND TECHNOLOGY
Committed to Academic Excellence

REPORT ON BIKE RACE GAME

Course Title: Software Development Project 5

SUBMITTED BY:

BULBUL AHMED	21225103242
SUBORNA SUTRADHAR	21225103104
MASURA AKTAR NODI	21225103071
M M TAREQ	21225103476

| Bijon Mallik
Lecturer, Department of CSE |

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BANGLADESH UNIVERSITY OF BUSINESS AND TECHNOLOGY
(BUBT)**

29th, May, 2025

Abstract

The Bike Race Game is a dynamic 2D racing game developed as part of the CSE 400 Software Development Project. It is designed to provide users with an engaging and interactive experience by allowing them to control a bike through challenging tracks filled with obstacles and time-based checkpoints. The goal of the game is to complete laps or stages in the shortest possible time while avoiding crashes and earning points.

This project aims to bridge academic learning with practical game development, integrating key elements of software engineering, such as user interface design, physics-based movement, event handling, and modular programming. The game is developed using Kotlin .

From the software perspective, this project covers the full development lifecycle — from planning and design to coding and testing. A simplified agile development approach was followed, allowing iterative refinement and frequent testing during development. The interface is designed to be user-friendly and compatible across multiple devices, ensuring a smooth gaming experience.

The core components of the game include sprite-based animation, keyboard input control, real-time collision detection, scoring logic, and game state management (start, play, game over). Additionally, the project is structured to be extensible, allowing future development of features like multiplayer modes, AI opponents, and customizable game environments.

Overall, the Bike Race Game project not only demonstrates the application of theoretical knowledge in game development but also emphasizes problem-solving, design thinking, and team collaboration. It serves as a foundation for more complex games and real-time applications.

Table of Contents

• Abstract	i
• Chapter 1: Introduction	1
• Chapter 2: Background	2
• Chapter 3: System Analysis & Design	3
• Chapter 4: Implementation	4
• Chapter 5: User Manual	5
• Chapter 6: Conclusion	6
• References	7

Chapter 1

INTRODUCTION

1.1 Problem Specification

Gaming is a dominant industry, yet many educational institutions lack resources to develop hands-on game development projects. This motivated us to create a fully functional Bike Race Game that combines fun with coding skills, real-time logic, and visual creativity.

1.2 Objectives

- Develop a responsive and interactive bike racing game.
- Integrate obstacle avoidance and scoring mechanisms.
- Enhance programming skills in game development.
- Deliver a working prototype for future expansion.

1.3 Scope

- 2D game rendering and control systems.
- Collision detection and score tracking.
- Usable on Android platforms.
- Can be extended to multiplayer or AI-based competition.

1.4 Organization of Project Report

- **Chapter 1:** Introduction and motivation.
- **Chapter 2:** Related work and analysis of existing systems.
- **Chapter 3:** Tools, models, and diagrams used.
- **Chapter 4:** Implementation process and modules.
- **Chapter 5:** User manual and requirements.
- **Chapter 6:** Conclusion, limitations, and future work.

Chapter 2

BACKGROUND

2.1 Existing System Analysis

Existing games like "Road Rash", "Bike Race Free", and "Trial Xtreme" offer similar features. However, many are either commercial or too complex for beginner-level implementation. Our system simplifies development while keeping core features intact.

2.2 Supporting Literatures

We referred to Unity documentation, physics engine tutorials, and online open-source examples. Technologies like Unity Game Engine, kotlin, and Photoshop for UI design were selected due to their accessibility and vast community support.

Chapter 3

SYSTEM ANALYSIS & DESIGN

3.1 Technology & Tools

- **Platform:** Android studio
- **Languages:** Kotlin,XML
- **Editor:** Visual Studio Code
- **Graphics:** Photoshop / Illustrator
- **Hardware:** Core i5, 8GB RAM, GPU Enabled System

Chapter 4

IMPLEMENTATION

4.1 Interface Design/Front-End

Game environment and UI were created using Unity's editor with customized sprites and audio.

4.2 Back-End

Scripts written in Kotlin handle physics, movement, score updates, and game events like collisions or game over states.

4.3 Modules

- **Start Screen**
- **Game Scene with Track and Bike**
- **Score Counter**
- **Collision Detection**
- **Game Over and Restart Logic**

Chapter 5

USER MANUAL

5.1 System Requirements

5.1.1 Hardware:

- 2 GHz Processor
- 8GB RAM
- 2GB Graphics Card

5.1.2 Software:

- Unity Hub / Python 3.x
- Windows 11
- Visual Studio Code

5.2 User Interfaces

Include screenshots of:

- Home Screen
- Game Play Interface
- Game Over Screen

Login Credentials (if any):

None required; local play.

Chapter 6

CONCLUSION

6.1 Conclusion

The Bike Race Game project strengthened our coding, problem-solving, and project management skills. We successfully implemented a functional racing game prototype. We learned how to use a game engine and write clean code. This project also improved our teamwork and problem-solving skills.

6.2 Limitations

- Single-player mode only
- No multiplayer or AI opponents
- Fixed map with no dynamic generation

6.3 Future Works

- Add AI-based opponents
- Online multiplayer support
- Dynamic map generation
- Leaderboards and achievements

References

- Unity Documentation: <https://docs.unity3d.com/>
- Lucidchart – Flow Diagrams: www.lucidchart.com
- InterviewBit – System Architecture: <https://www.interviewbit.com/blog/system-architecture/>