

# **BLEACH GAME**

**Submitted by:**

**Jeeshan Shrestha(081bct012)**

**Pranay Tuladhar(081bct020)**

**Hari Bahadur Basnet(081bct011)**

**Submitted to:**

**Department of Electronics and Computer  
Engineering**

**Thapathali Campus**

**Kathmandu, Nepal**

**Date:2025/03/12**

# Introduction

## ➤ Game Concept

- Inspired by the anime Bleach
- featuring sword-based combat

## ➤ Objectives

- Defeat enemies
- gain points
- face the boss

## ➤ Language Used

- C programming

# Gameplay Mechanism

## Character Movement

- Left, Right, Up and Down

## Enemy Respawn

- Enemies appear from any direction

## Attack Mechanism

- Sword-based slash attack
- Flash-Step (Increases player movement speed)

## Scoring

- +10 points per defeated enemy.

## Boss Fight

- After 50 points, the main boss (Aizen) appears
- After 1000 points, Game final Boss appears
- Boss fires a projectile that stuns player for 1 seconds.

# Game Features

## Player Lives

- 10 lives, reduced upon enemy contact and Boss Contact

## Progression

- Enemies increase in number and difficulty

## End Condition

- Player loses all lives, or else the game runs infinitely and compete to get the highest score.

# Development Process

## ➤ Programming Language

- C

## ➤ Libraries Used

- raylib
- stdbool
- stdio

## ➤ Challenges Faced

- Implementing movement and enemy AI
- Collision detection
- Scorekeeping and game progression

## ➤ Solutions Implemented

- Optimized logic
- Debugging techniques.

# Tools and resources used

## IDE

- Visual Studio Code
- Visual Studio

## Version Control

- Git

## References

- Raylib Documentation: <https://www.raylib.com/>
- Bleach Anime: <https://www.crunchyroll.com/bleach> Game
- Development Concepts: <https://www.gamedev.net/> Error Debugging: <https://chatgpt.com/>

# Future Improvements

## Enhanced Graphics

- Using external libraries like SDL

## Sound Effects & Music

- Adding immersive audios

## Multiplayer Mode

- Introducing a two-player battle mode

## AI Improvements

- Smarter enemies with varying attack patterns

# Lessons Learned

## Programming Skills

- Improved C proficiency
- Improved skills to write high quality codes

## Problem-Solving

- Debugging and optimizing code
- Patience enhancement and better understanding

## Game Development Workflow

- Understanding the steps from concept to execution
- Making the prototype to implement in on the actual project



# Screenshots / Game Demo



Player

Enemy

Main Boss

# Conclusion & Final Thoughts

## Summary

- Developed a C-based game with anime-inspired mechanics.
- Understood new concepts like `deltaTime` and `gameLoop`.

## Key Takeaways

- Experience in game development, programming logic, and debugging.

## Final Words

- Excited for future improvements and new projects.