MarioCraft Game Project

Developers

İsmail Atasoy
(2023556008)(EC)
Mehmet Kemal Ertunç
(2023556038)(EC)
Erdem Cabioğlu
(2023556017)(EC)

01/06/2025

Index

1-Abstract

This project introduces a 2D platformer game inspired by the classic Super Mario series and minecraft, developed in C using the Raylib graphics library. The game includes a main menu three progressively challenging levels, collectible coins, power-ups, and enemies with basic AI behavior. The purpose of the project is to explore game development concepts such as physics simulation, sprite animation, collision detection, and level progression using low-level programming. The final product demonstrates a functional, interactive game with solid gameplay mechanics and a complete game loop.

2-Introduction

2.1 Definition of the Project

The project is a side-scrolling 2D platformer game clone of Super Mario, developed with Raylib in the C programming language.

2.2 Purpose of the Project

The aim is to implement a playable and extendable platform game to understand real-time rendering, player interaction, and basic AI systems.

2.3 Objectives and Expectations

- Develop a functioning game with smooth controls
- Include features like coin collection, power-ups, and enemy encounters
- Implement a main menu, game over state, and level progression
- Gain experience in low-level game programming and asset handling

2.4 Scope

The project includes a fully playable prototype with 3 levels, different types of obstacles, player growth mechanics, and basic sound and animation logic. It does not include advanced AI.

3. Methodology

3.1 Technologies Used

- Develop a functioning game with smooth controls
- Include features like coin collection, power-ups, and enemy encounters
- Implement a main menu, game over state, and level progression

3.2 Development Process

The game was structured into components: player control, map design, enemy logic, rendering, and game state management. Development followed an iterative approach, with regular testing after implementing each feature.

3.3 Applied Methods

- Tile-based level design using char arrays
- Game states handled via enums
- Physics via gravity, velocity, and jump mechanics
- Collision detection using bounding rectangles
- Sprite animation via frame counters and timers

4. Results & Evaluation

4.1 Project Achievements

- Fully functional 2D Mario-style game
- Smooth player controls and jumping physics
- Three unique levels with increasing difficulty
- Working menu, settings menu, game over, and level complete screens
- Power-ups and coin collection integrated successfully

4.2 Challenges Encountered

- Collision detection required fine-tuning
- Level data management became complex with multiple arrays
- Camera system alignment and level scrolling adjustments

4.3 Limitations

- Basic enemy behavior (only movement, no attack logic)
- The charachter cannot climb on the walls

5. References

- Raylib official website
- Raylib GitHub Repository
- StackOverflow
- YouTube tutorials for Raylib and 2D platformer design