1. Title

CoinMaster: A Fast-Paced Coin Catching Game

2. Team Member List

Name	ID	Contribution
Risul Islam Rifat	22-48931-3	Game Design, Feature Implementation
Naznin Akhter	22-48918-3	Environment Development, Testing
Rafin Khan	22-48797-3	Graphics, Documentation

3. Introduction

CoinMaster is an engaging and dynamic game where players aim to catch coins randomly falling from the sky. Designed for entertainment and quick reflex development, CoinMaster challenges users to sharpen their reaction time in a visually appealing and interactive environment.

Similar to other casual arcade games like *Fruit Ninja* or *Flappy Bird*, CoinMaster's appeal lies in its simplicity and addictiveness. Players compete to collect as many coins as possible within a given time frame, unlocking higher levels and unique game modes as they progress.

Our motivation stems from the growing popularity of casual gaming, which attracts a wide range of players, from casual users looking for a quick diversion to more competitive gamers. CoinMaster aims to strike a balance between fun and challenge, providing an engaging experience through innovative gameplay mechanics, visually rich graphics, and smooth performance.

The scope of this project includes the development of a complete 2D game using an appropriate graphics framework, incorporating a scoring system, and designing intuitive controls that maximize user engagement. Future potential includes multiplayer functionality and advanced coin behaviors to enhance replayability.

4. Description of Environment and List of Methods

The development environment for *CoinMaster* includes:

- **Programming Language**: C++
- **Graphics Library**: OpenGL and GLUT for 2D game design and rendering
- **Development Tools**: Code::Blocks IDE, Git for version control
- **Platform**: Windows OS for development and deployment

Methods:

- **Coin Spawning Algorithm**: Randomized positioning using C++ standard libraries and OpenGL.
- **Collision Detection**: Implemented with bounding box algorithms to detect player-coin interactions.
- **Game Timer and Scoring**: Using GLUT's timer functions for countdown and real-time score updates.
- **Animation**: Smooth coin-falling animations using OpenGL transformations.

5. Feature Set

1. Player character with customizable appearance.
2. Random coin generation at varying speeds and trajectories.
3. Collision detection between player and coins.
4. Scoring system displaying real-time updates.
5. Time-limited gameplay with adjustable durations.
6. Multiple difficulty levels (Easy, Medium, Hard).
7. Background music and sound effects for coin collection.
8. Pause and resume functionality.
9. Leaderboard to display high scores.
10. Visual effects when coins are collected.
11. Bonus coins that provide extra points.
12. Power-ups like slow motion or double points.
13. Penalty objects (e.g., falling bombs) to avoid.

14. Increasing difficulty as the game progresses.
15. End-game summary with performance metrics.
16. Full-screen mode for immersive play.
17. Settings menu to adjust sound and controls.
18. Animated backgrounds for visual appeal.
19. Tutorials to help new players.
20. Adaptive coin spawn rate based on player performance.

6. Conclusion

CoinMaster promises to deliver an entertaining and challenging gaming experience. By combining visually engaging graphics with intuitive gameplay mechanics, it aims to cater to a broad audience of casual and competitive players.

Future enhancements could include multiplayer options, integration with social platforms for score sharing, and additional gameplay modes like survival challenges or puzzles. The main limitation is the scope of initial deployment, focusing on a 2D environment with basic mechanics, leaving room for future growth.

Through this project, we hope to refine our programming and design skills while producing a fun and interactive game for all audiences. Here are some references which was used in the project.

1. Statista. (2023). "Casual gaming market share worldwide." Retrieved from https://www.statista.com
2. OpenGL Official Documentation. "Overview of GLUT and OpenGL Graphics Programming." Retrieved from https://www.opengl.org
3. Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. (1996). *Computer Graphics: Principles and Practice*. Addison-Wesley.
4. NeHe Productions. (n.d.). "OpenGL Tutorials." Retrieved from http://nehe.gamedev.net
5. Newzoo. (2023). "Player retention statistics with social features." Retrieved from https://newzoo.com
6. www.google.com for picture collection