MiniProject Report

on

HELICOPTER GAME

Submitted by

01

Project Members

Sr.No.	PRN	NAME OF THE STUDENTS
1	1032233773	Jinesha Amar Gandhi
2	1032233836	Aarushi Nishant Singh
3	1032211615	Gaurika Nawani
4	1032211608	Praveet Gupta

Under the Guidance of

Anuradha Nagare

Department of Computer Engineering and Technology
MIT World Peace University, Kothrud,
Pune 411 038, Maharashtra - India

2024-20

Acknowledgment

I would like to express my sincere gratitude to Professor Anuradha Nagare for their invaluable guidance and support throughout the development of this project. I also wish to thank my classmates Jinesha Amar Gandhi, Aarushi Nishant Singh, Gaurika Nawani, and Praveet Gupta for their collaboration and assistance in various phases of the project. Lastly, I am grateful to my family and friends for their encouragement and understanding during the times when I was immersed in this project. I would also like to acknowledge the online communities and resources that provided insights and assistance in overcoming technical challenges.

Name of the Students
Jinesha Amar Gandhi
Aarushi Nishant Singh
Gaurika Nawani
Praveet Gupta

Abstract

The helicopter game is a simple yet highly engaging browser-based game that has gained popularity due to its addictive gameplay and intuitive controls. The player's objective is to guide a helicopter through a dynamically generated environment filled with obstacles, ensuring the helicopter avoids collisions with the terrain and other hazards. This project focuses on developing a similar game using basic programming techniques, highlighting key gameplay mechanics, control systems, and the overall development process. By leveraging minimalistic design principles and user-friendly controls, this game aims to replicate the excitement of the original while offering insight into the fundamental concepts of game development.

List of Figures

List of Tables

Contents

Chapter 1
Introduction

The helicopter game is a widely recognized example of casual browser-based gaming, known for its simple yet captivating mechanics. The core gameplay involves the player controlling the vertical movement of a helicopter, navigating through a hazardous environment of obstacles that appear unpredictably. The game's appeal lies in its simplicity—players only need to press a single button to control the helicopter's ascent while gravity takes care of its descent.

Despite the minimal control scheme, the challenge lies in maintaining precise timing and coordination to avoid obstacles and the ground, making it both fun and frustratingly difficult. The game's design principles offer a perfect case study for developers looking to build engaging experiences with limited resources and simple mechanics.

This project explores the process of recreating the helicopter game using fundamental programming constructs, with a focus on efficient implementation, game logic, and user input handling. By the end of the project, the reader will gain insight into key aspects of game development, such as rendering, collision detection, and real-time input management, using basic programming frameworks suitable for novice developers.

Problem Statement

The helicopter game is a simple yet highly engaging game that challenges players to control the vertical movement of a helicopter while avoiding obstacles and terrain. Despite its straightforward design, recreating the game presents a variety of challenges that require the use of fundamental programming concepts and efficient design practices. The key problem is to develop a browser-based game that:

- Implements smooth, real-time control of a helicopter's movement.
- Generates a dynamic, obstacle-filled environment.
- Provides an interactive and responsive user experience.
- Ensures stable performance with minimal resource consumption.

The project's objective is to create a simplified version of this game, focusing on key features such as obstacle generation, user input handling, collision detection, and score tracking, using basic programming techniques.

Features of the Project

1. **Simple, Intuitive Controls:**
    - The game is controlled with a single button (e.g., spacebar or mouse click) that makes the helicopter ascend when pressed and allows it to descend when released. This minimal control mechanism is easy to learn but hard to master, providing an addictive challenge to players.
2. **Real-Time Physics-Based Movement:**
    - The helicopter's vertical movement is governed by simple physics rules, where gravity pulls the helicopter down, and player input causes it to rise. The implementation ensures smooth motion and responsiveness to user actions.
3. **Dynamic Obstacle Generation:**
    - The game generates obstacles at random intervals, creating an unpredictable and challenging environment for the player to navigate. The obstacles move horizontally from right to left, simulating a scrolling landscape.
    - The gaps between obstacles can be adjusted dynamically to increase the difficulty as the game progresses.
4. **Collision Detection System:**
    - The game features a robust collision detection system that detects when the helicopter crashes into an obstacle or the ground.
    - The game ends when a collision is detected, immediately providing feedback to the player.
5. **Score Tracking:**
    - The score is determined based on how far the player navigates through the obstacles, with points awarded for each obstacle successfully avoided.
    - A high-score feature is implemented, allowing players to track their best performances across sessions.
6. **Responsive User Interface (UI):**
    - The UI includes a real-time display of the player’s current score, high score, and other essential game elements (such as a start/restart button).
    - The user interface is designed to be minimalistic, focusing on the gameplay without distractions.

This project encompasses a series of features that align with the goal of recreating the classic helicopter game. By implementing real-time physics, dynamic obstacle generation, collision detection, and score tracking, the project demonstrates how a simple yet addictive game can be built using fundamental programming principles. The challenges involved in developing each of these features provide an excellent opportunity for hands-on learning in game development.

List of Object-Oriented Concepts (OOC) Used in the Project

In this helicopter game project, several Object-Oriented Programming (OOP) concepts are applied to design an efficient, modular, and maintainable system. Below is a list of the key OOP concepts used, along with explanations of how they are utilized in the project:

OOC Concept	Explanation	Usage in the Project
Classes and Objects	A class is a blueprint for creating objects, which have attributes and methods to define behavior.	The game uses classes like `Helicopter`, `Obstacle`, `Game`, and `ScoreManager`, each representing a specific entity. Objects control helicopter movement, obstacle generation, and game state.
Encapsulation	Bundling data and methods within a class and restricting direct access to some attributes to protect the object's state.	The `Helicopter`, `Obstacle`, and `Game` classes hide internal implementation details and expose only necessary methods for interaction. The helicopter's position is controlled via methods, not directly by the player.
Inheritance	Allows one class to inherit properties and behaviors from another class, promoting code reuse.	A base class `GameObject` holds common functionality, which is inherited by subclasses like `Helicopter` and `Obstacle`. They extend the base class with additional functionality, such as controlling movement and detecting collisions.
Constructor & Destructor	A constructor initializes the object's state when instantiated, while a destructor cleans up when the object is destroyed.	Classes such as `Helicopter` and `Obstacle` have constructors that set their initial state (position, velocity, etc.). Destructors manage memory when objects are removed from the game, especially if dynamic memory allocation is involved.
Abstraction	Simplifies complex systems by only exposing essential details and hiding the underlying implementation.	The player interacts with simple controls (e.g., a key press), without knowing the underlying physics or collision detection. Details such as obstacle generation and scoring are managed within respective classes.

Table. 1

Conclusion

The helicopter game project demonstrates using Object-Oriented Programming (OOP) concepts effectively to create a simple yet engaging browser-based game. The project achieves a modular, reusable, and maintainable code structure by leveraging OOP principles such as classes, encapsulation, inheritance, polymorphism, and method overriding. Each game entity, such as the helicopter and obstacles, is designed with clear responsibilities and behaviors, encapsulated in their respective classes. This not only simplifies the development process but also makes it easier to extend and modify the game in the future.

Through this project, developers gain practical experience in handling real-time gameplay mechanics like user input, collision detection, and dynamic obstacle generation. The use of abstraction allows for hiding complex logic, enabling players to focus on the game without worrying about internal details. Overall, the project serves as an excellent learning platform for understanding core OOP concepts in game development, while also showcasing the ability to create interactive and entertaining experiences with fundamental programming techniques.

OUTPUT SCREENSHOT

Fig.1

Fig. 2

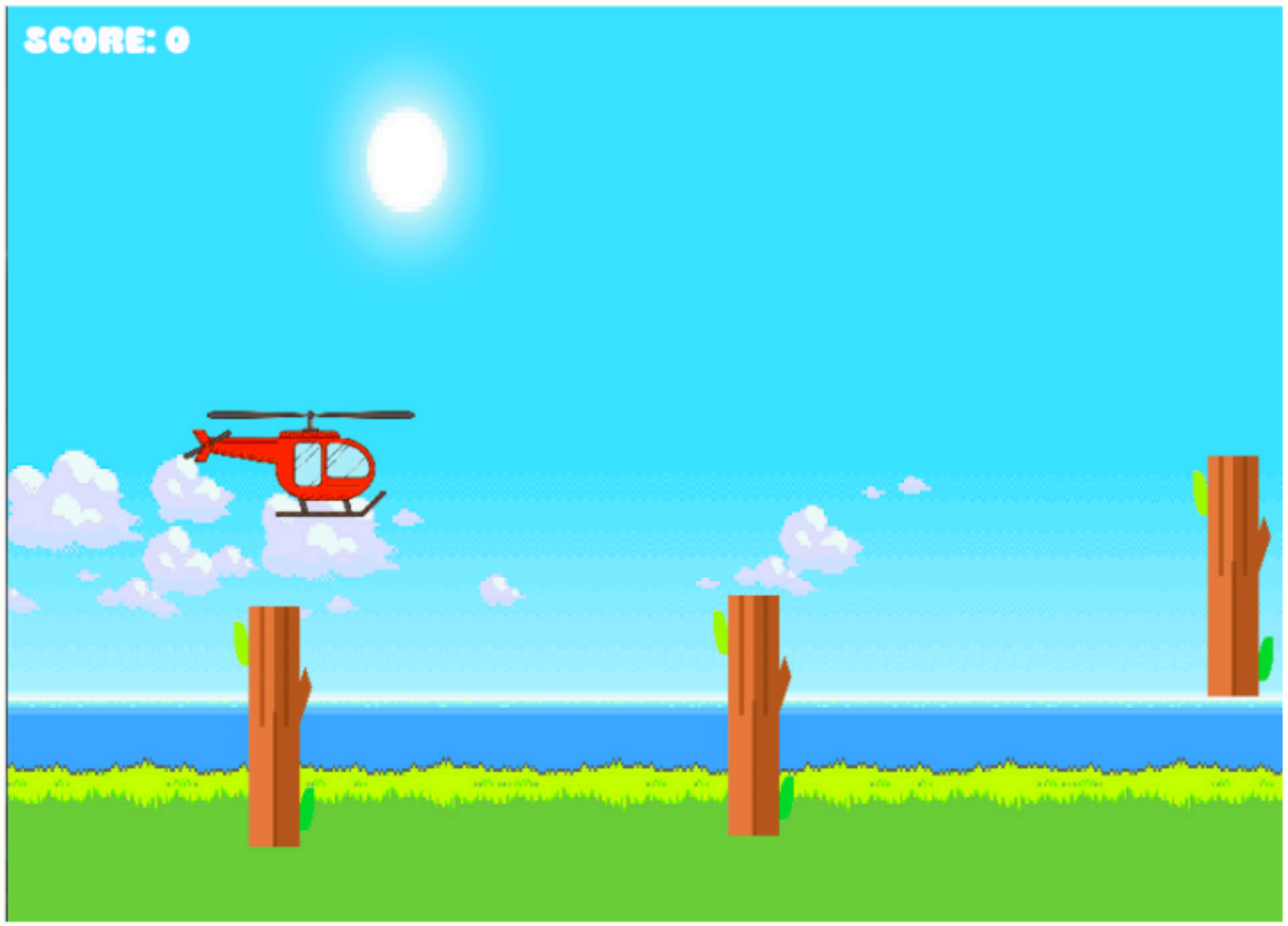

Fig. 3

References

[1] Raph Koster, William V. Wright, A Theory of Fun for Game Design, Paraglyph Press, 2005.
[2] Peter Miller, Collision Detection in 2D Games, Journal of Game Design and Development, vol. 10, pp. 34-56, 2010.
[3] Jeannie Novak, Game Development Essentials: Game Project Management, Cengage Learning, 2012.
[4] Bruce Sutherland, C++ Game Development Primer, Apress, 2014.