G D GOENKA
UNIVERSITY
FROM FOUNDATION TO FLIGHT

IDP FILE

Submitted By:-
Aayush Verma
Dorya Vardan Jain
Chirag Kaushik

Submitted To:-
Vaibhav Sir

DECLARATION BY STUDENT

We, Aayush Verma, Dorya Vardan Jain, and Chirag Kaushik, students of B.Tech at GD Goenka University, solemnly declare that the project work titled "Flappy Bird Game using HTML, CSS, and JavaScript" is a result of our sincere effort and dedication. This project report is our original creation and has been carried out under the valuable guidance of Mr. Vaibhav Sharma. We confirm that the content presented herein has not been submitted elsewhere for the award of any other academic degree or diploma.

Our work in this project involved conceptualization, coding, testing, and documentation. We have given due credit to all external sources consulted during this process. This project helped us understand several key aspects of web development, such as event handling, animation, DOM manipulation, and real-time feedback in user interactions. We are proud to have independently developed a browser-based game that not only mimics a popular title but also serves as a learning model for fellow developers.

DECLARATION BY GUIDE

This is to certify that the project work entitled "Flappy Bird Game using HTML, CSS, and JavaScript" is a bona fide and original work submitted by Aayush Verma, Dorya Vardan Jain, and Chirag Kaushik, students of B.Tech in Computer Science and Engineering, GD Goenka University. The project has been carried out under my supervision and fulfills the necessary academic requirements of the program.

Throughout the duration of this project, the students have shown remarkable commitment and technical understanding. They have demonstrated competence in the development of web-based applications and proficiency in using modern technologies like HTML5, CSS3, and JavaScript ES6+. The students took initiative in planning and executing the game development with structured methodology and creative problem-solving.

I recommend the report for submission and evaluation.

ACKNOWLEDGEMENT

It gives us immense pleasure to present this project report as a part of our curriculum. We are sincerely thankful to our guide, Mr. Vaibhav Sharma, for his continuous support, insightful suggestions, and regular feedback which enabled us to complete this project successfully.

We also extend our gratitude to the faculty and staff of the Department of Computer Science and Engineering, GD Goenka University, for their help and encouragement. Special thanks go to Sunstone for equipping us with the necessary resources and providing a learning-friendly environment.

We are also thankful to our peers and friends for their valuable suggestions and motivational support. Finally, we express deep appreciation to our families, whose patience, encouragement, and understanding were critical throughout the course of this project.

This project has been a remarkable journey of learning, and we are proud to present this accomplishment.

ABSTRACT

The primary objective of this project is to design and implement a browser-based version of the popular mobile game 'Flappy Bird' using front-end web technologies such as HTML, CSS, and JavaScript. This project is intended to serve as an introductory yet meaningful exercise for students aiming to explore the fundamentals of web development and game design.

The game involves a bird navigating through gaps between pipes by flapping upwards in response to user input. Its straightforward gameplay mechanics belie a surprising depth of logic, particularly in areas such as gravity simulation, collision detection, and dynamic obstacle generation.

The overall project demonstrates practical application of core programming concepts including the Document Object Model (DOM), event listeners, timing functions, and UI responsiveness. It encourages teamwork, structured coding, testing, and debugging—all essential skills in software development.

Beyond technical development, the project promotes problem-solving, iterative design, and creative thinking. Its successful execution reflects our understanding of web technologies and serves as a testament to our commitment and collaborative effort.

Flappy Bird Game Project Report

CHAPTER 1: INTRODUCTION

1.1 Problem Statement

There is a lack of simple, engaging, and educational projects for students new to web development. Many beginners struggle to find projects that are both technically instructive and enjoyable to build. Traditional academic projects may lack interactivity or fail to inspire creativity. By recreating the Flappy Bird game using HTML, CSS, and JavaScript, we aim to bridge this gap and provide a project that is both fun and rich in learning opportunities. This game project also promotes logic building, real-time rendering understanding, and efficient coding practices.

1.2 Background

Flappy Bird, developed by Dong Nguyen in 2013, became a viral sensation due to its addictive and challenging gameplay despite the simplicity of its mechanics. The game featured pixel-art graphics, minimal controls, and endless level generation, which captivated users globally. Although the original game was removed from app stores, its influence remains significant. Our version adapts this classic game for web browsers using modern web technologies. The intent is to introduce budding developers to front-end development through an engaging and familiar project.

1.3 Objectives

- To design and implement a playable version of Flappy Bird using web technologies.

- To understand the integration of HTML, CSS, and JavaScript in creating interactive applications.

- To gain experience in developing a user interface, game logic, and responsive controls.

- To develop teamwork, coding, debugging, and project management skills.

- To explore key aspects of client-side rendering and performance optimization.

1.4 Scope

The scope of the project is limited to front-end technologies, implying that there will be no server-side scripting or data persistence. It is a single-player game operating entirely within the browser environment. While minimalist by design, the game framework may be expanded in future iterations to include multiplayer functionality, data logging, and mobile compatibility. This phase of the project targets mastering the fundamentals of HTML structure, CSS styling, and JavaScript game logic.

CHAPTER 2: LITERATURE REVIEW

2.1 Review of System Architecture

The architecture of the Flappy Bird game is purely client-side, employing a modular approach. HTML serves as the structural foundation, defining canvas elements and interactive UI components. CSS is layered on top for visual enhancements, applying style rules to game assets such as the bird, pipes, and background. JavaScript is the core logic engine, orchestrating real-time dynamics including user input response, game state management, scorekeeping, collision logic, and rendering loops. This separation of concerns results in clean, maintainable code.

2.2 Introduction to the Game

Our version of Flappy Bird is a 2D side-scrolling game in which the player controls a bird attempting to fly between columns of green pipes. The game features gravity-based mechanics, where the bird continuously descends unless flapped upward by the player using the spacebar. Obstacles are generated dynamically, and the player's score increases with each successful pass through a set of pipes. The simplicity of gameplay belies the complexity of timing and precision required to succeed, contributing to its addictiveness.

2.3 SWOT and PEST Analysis

SWOT Analysis:

Strengths: Lightweight, easy to understand, ideal for beginners.

Weaknesses: Lacks advanced features like sound, save progress, or adaptive difficulty.

Opportunities: Room for enhancements such as multiplayer, animations, or mobile integration.

Threats: Market saturation with similar games, limited originality.

PEST Analysis:

Political: No significant restrictions affecting browser-based games.

Economic: Highly cost-effective development with free tools and open-source libraries.

Social: Strong social appeal for retro-style and minimalistic games.

Technological: Benefits from the rapid evolution of browser rendering engines and JavaScript frameworks.

2.4 4Ps of Marketing

Product: A playable web adaptation of Flappy Bird built for educational use.

Price: Free and open-source, targeting accessibility.

Place: Hosted on websites, GitHub repositories, or coding platforms.

Promotion: Shared via developer forums, educational platforms, and social networks.

2.5 Customer and Competitor Analysis

The primary customers are students and novice programmers seeking to practice front-end development. Educators may also use this game in coursework to demonstrate interactive programming concepts. Competitors include other browser games like Chrome's Dino Run, 2048, and similar clones available on platforms like Itch.io and CodePen. However, the educational orientation of this project provides a unique niche.

CHAPTER 3: METHODOLOGY

3.1 Technical Requirements

This game is developed using modern web standards. HTML5 offers semantic structuring, CSS3 adds aesthetic control and responsiveness, and JavaScript (ES6) introduces modular coding and advanced control structures. No additional libraries are required, simplifying deployment and ensuring wide browser support.

3.2 Hardware and Software Requirements

Hardware: A basic computer or laptop with 4GB RAM, mouse/keyboard input, and internet access for testing and documentation.

Software:

- Visual Studio Code (or any text/code editor)

- A modern browser (e.g., Chrome, Firefox)

- Optional: Git, GitHub for version control

3.3 Software Components

index.html - Creates the structure of the canvas, score counter, and instructions.

style.css - Styles elements with gradients, positioning, animation effects, and sprite dimensions.

script.js - Contains core game logic, physics, user controls, pipe generation, and collision detection.

3.4 Project Flow

The game initializes upon loading the webpage. When the player presses the spacebar, a flag triggers the game loop, causing the bird to start falling. Pipes move from right to left using a setInterval or requestAnimationFrame loop. Each flap lifts the bird slightly, providing brief upward momentum. A collision detection routine monitors pipe and ground contact. The game ends when a collision is detected and displays the current score.

CHAPTER 4: DATA ANALYSIS AND OBSERVATIONS

4.1 Generating Output from the Project

Upon successful implementation, the game displays a responsive canvas-based output. The background scrolls, the bird animates, pipes spawn periodically, and the score updates with each successful pass. Visual cues such as pipe movement and bird flapping are programmed using CSS transitions and JavaScript event handling.

4.2 Generating Website Statistics

Although the project is front-end only, analytics can be implemented using third-party tools like Google Analytics. These tools track session duration, bounce rates, interaction clicks, and page views. Custom events can be added to track gameplay frequency, start-end sessions, and average duration.

CHAPTER 5: CONCLUSION AND FUTURE DIRECTIONS

5.1 Project Significance

This project bridges theory and practice by allowing students to apply web development concepts to create an interactive product. It also emphasizes the importance of user experience, responsiveness, and cross-browser compatibility. Such projects promote self-learning, confidence building, and code optimization techniques.

5.2 Conclusion

The Flappy Bird game has been effectively recreated using HTML, CSS, and JavaScript. It demonstrates an understanding of game loop mechanics, DOM manipulation, and collision logic. The game is lightweight, intuitive, and fully functional across most modern browsers.

5.3 Limitations and Future Research

Limitations:

- No background music or sound effects.
- Not optimized for touch controls or small screens.
- Single game mode without progression.

Future Enhancements:

- Integration of sound using the Web Audio API.
- Support for mobile and touch inputs via responsive design.
- Multiple levels, increasing difficulty, and bonus challenges.
- Use of Canvas API for enhanced rendering and animation performance.