2048 Game

A PROJECT REPORT

Submitted by

Mohammad Saiful Haque (22BCS15656)

Nikhil Kumar (22BCS15501)

Agrim Justa (22BCS15530)

in partial fulfillment for the award of the degree of

Bachelors of Engineering

IN

Computer Science

Chandigarh University

April - 2025

BONAFIDE CERTIFICATE

Certified that this project report **"2048 Game"** is the Bonafide work of **"Mohammad Saiful Haque, Nikhil Kumar, Agrim Justa"** who carried out the project work under my/our supervision.

SIGNATURE	**SIGNATURE**
Mr. Sandeep Singh Kang	**Er. Rahul Durgapal**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Computer Science & Engineering	Computer Science & Engineering

Acknowledgment

We would like to express our sincere gratitude to everyone who contributed to the successful completion of this 2048 Game project. First and foremost, we are grateful to our mentor and supervisor, for their invaluable guidance, support, and encouragement throughout the project. Their expertise and insights were instrumental in shaping our approach and refining our work. We also extend our thanks to our team members for their dedication, teamwork, and commitment to achieving our shared goals. Each member's contributions were essential to the development, design, and testing phases, and their collaborative spirit made this project both enjoyable and productive.

Additionally, we appreciate the online resources, such as GeeksforGeeks, JavaTpoint, and other educational platforms, for providing tutorials and examples that supported our coding and problem-solving processes. These resources proved invaluable in helping us understand the complexities involved in creating an engaging and functional game.

Finally, we thank our family and friends for their encouragement and understanding during the course of this project. Their support motivated us to work diligently and complete the project on time.

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION.....5

1.1. Identification of Client/ Need/ Relevant Contemporary issue 5
1.2. Identification of Problem.....5
1.3. Identification of Tasks 6
1.4. Timeline 8
1.5. Organization of the Report 9

CHAPTER 2. LITERATURE REVIEW/BACKGROUND STUDY 11

2.1. Timeline of the reported problem.....11
2.2. Existing solutions 12
2.3. Bibliometric analysis.....14
2.4. Review Summary 16
2.5. Problem Definition 18
2.6. Goals/Objectives.....20

CHAPTER 3. DESIGN FLOW/PROCESS.....22

3.1. Evaluation & Selection of Specifications/Features.....22
3.2. Design Constraints 24
3.3. Analysis of Features and finalization subject to constraints 26
3.4. Design Flow 28
3.5. Design selection 30
3.6. Implementation plan/methodology 32

CHAPTER 4. RESULTS ANALYSIS AND VALIDATION 39

4.1. Result Outcomes 39

CHAPTER 5. CONCLUSION AND FUTURE WORK.....41

5.1. Conclusion.....41
5.2. Future work 42

REFERENCES 45

Abstract

This project report presents the development and design of the classic 2048 game, a popular tile-merging puzzle that has engaged players worldwide since its inception. The project focuses on creating a user-friendly, visually appealing, and smooth- running version of 2048, implementing game mechanics that follow the original gameplay concept: players combine matching numbered tiles to reach the ultimate goal of a 2048 tile.

The report details the entire development process, including the identification of client needs, problem definition, task planning, and design choices. Various gameplay features, such as movement control, tile merging, and scoring systems, have been carefully constructed to ensure an engaging user experience. The project employs a structured approach, including Agile methodology, to manage tasks and achieve goals within set timelines.

The team incorporated essential coding practices and design constraints to maintain a seamless gaming experience. Additionally, future work is outlined to expand gameplay options, improve user interface elements, and add compatibility across platforms. This project report not only highlights the technical development of the game but also discusses potential advancements to enhance its appeal in the competitive mobile and web gaming industry.

CHAPTER 1.
INTRODUCTION

1.1. Identification of Client / Need / Relevant Contemporary Issue:

The 2048 game, originally developed by Gabriele Cirulli, quickly became a global sensation due to its simple yet challenging gameplay mechanics. Although it was created as a personal project, its viral success demonstrated a growing trend in casual gaming where players seek engaging, mentally stimulating activities that can be played in short bursts. This trend aligns with an increasing societal focus on digital experiences that balance entertainment with cognitive benefits.

With the widespread use of mobile and web-based games for quick entertainment and mental exercise, the 2048 game's simple design and compelling gameplay illustrate a relevant contemporary issue: the demand for accessible, cognitively enriching digital games. This project not only meets this need but also opens doors for further study into game algorithms.

1.2. Identification of Problem:

The primary challenge in creating or analysing the 2048 game lies in developing a system that captures its addictive gameplay mechanics while ensuring a smooth user experience. The game requires precise logic to handle tile movements, merging operations, and randomized tile generation, which must work seamlessly to prevent errors that disrupt gameplay.

From a technical perspective, implementing the game involves a series of algorithmic challenges:

- **Tile Movement and Merging Logic**: Ensuring that tiles move in the correct direction, merge only with matching tiles, and adhere to rules on merging once per move.
- **Randomized Tile Generation**: New tiles must appear in empty spaces at random, typically with specific probabilities, to retain the element of unpredictability that defines the game's challenge.
- **Game Over Conditions**: The system must continuously monitor the board for possible moves and end the game appropriately if no moves are left.

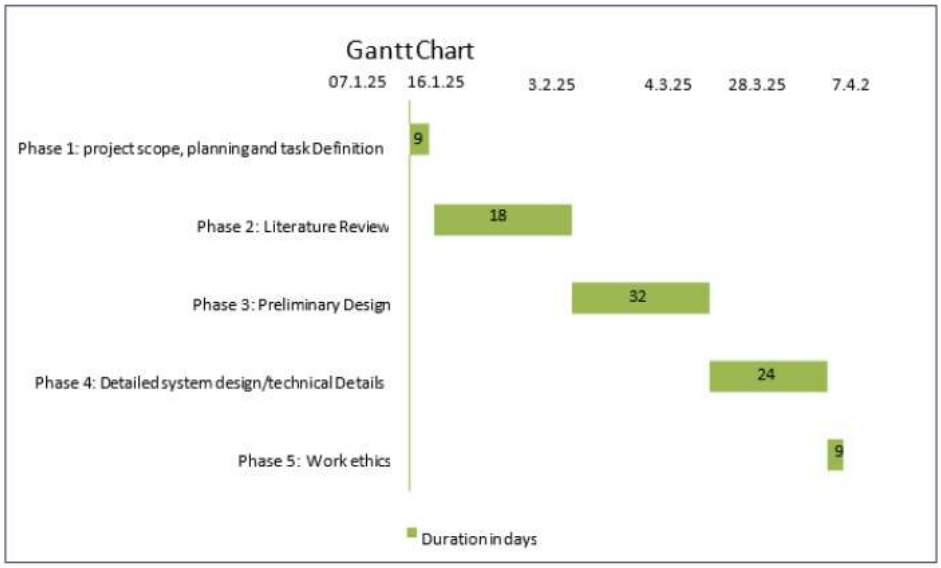
1.3. Identification of Tasks:

The project tasks for developing or analysing the 2048 game are structured to address the technical, design, and testing requirements needed for a fully functional and engaging gameplay experience.

These tasks include:

1. **Game Design and Planning**:
    - Outline the core mechanics of the game, including movement, merging rules, scoring, and game-over conditions.
    - Design the user interface layout, defining elements like the 4x4 grid, tile appearance, score display, and game controls.
2. **Algorithm Development**:
    - Develop the core movement logic to allow tiles to shift in the intended direction (up, down, left, or right) based on player input.
3. **Interface Implementation**:
    - Build the visual interface using the chosen programming language or game engine, applying consistent styling for grid lines, tiles, and animations.
4. **Scoring System**:
    - Implement a scoring algorithm that calculates points based on tile merges, keeping track of the player's total score throughout the game.
5. **Game State and Ending Conditions**:
    - Implement continuous monitoring of the grid for possible moves and determine when the game ends if no moves are left.
6. **Testing and Optimization**:
    - Test the game mechanics to ensure that all movement, merging, and scoring functions work as intended.
    - Optimize the performance to minimize lag or delays in tile movement and merging, especially for mobile devices.
7. **Documentation and User Instructions**:
    - Document the code to ensure clarity for future reference and maintenance.
    - Write concise user instructions on how to play the game, including controls, scoring, and objectives.

1.4. Timeline:



1.5. Organization of the Report:

This report on the 2048 game project is organized into several sections to provide a clear and structured overview of the project objectives, development process, and outcomesThe report structure is as follows:

1. **Introduction**: This section presents an overview of the 2048 game, its significance, and the motivation behind selecting it as a project topic.
2. **Problem Identification and Task Analysis**: This section identifies the primary challenges and tasks involved in the project.
3. **Methodology**: This section details the steps taken to implement the game, including algorithm development, interface design, and testing strategies.
4. **Implementation and Development**: This section provides a deep dive into the core components of the 2048 game.
5. **Challenges and Solutions**: This section discusses the difficulties encountered during the development process and the strategies used to resolve them.
6. **Results and Evaluation**: This section analyzes the outcomes of the project.
7. **Conclusion and Future Work**: This section summarizes the project, reflecting on the key insights gained and potential areas for further development.
8. **References**: This final section includes citations for all sources, tutorials, or research papers that contributed to the project.

CHAPTER 2.
BACKGROUND STUDY

2.1. Timeline of the reported problem:

The development and analysis of the 2048 game project followed a structured timeline, with each phase addressing a specific aspect of the game's design and implementation. Below is an overview of the timeline:

Played the original game for hands-on understanding, researched existing resources and algorithms, and reviewed project requirements. Developed the primary functions for handling tile shifts, merges, and scoring updates. Conducted initial tests to verify functionality. Created the game's visual layout, including the 4x4 grid and tile styling. Added controls for tile movement based on user input (keyboard or swipe). Developed the algorithm for spawning new tiles at random positions, applied probability rules, and coded logic for game-over detection. Conducted multiple playtests, identified and resolved bugs, optimized tile movement for smooth gameplay, and ensured responsiveness across devices. Documented each function and game mechanic, wrote instructions for gameplay, and compiled project findings and results into the final report. Prepared the final project for submission, including code, documentation, and gameplay demonstration.

2.2. Existing solutions:

The 2048 game has inspired various implementations and solutions across different platforms since its creation. Some notable solutions are outlined below:

- The original 2048 game, developed by Gabriele Cirulli, is a simple and open-source web-based version that quickly gained popularity due to its accessible design and addictive gameplay.
- Following the success of the original game, many developers created mobile versions for iOS and Android to cater to the growing demand for casual mobile gaming.
- Several AI-based solutions were developed to explore optimal strategies and automate gameplay for the 2048 game. These AI implementations often use

algorithms like Minimax, Expect Imax, and Reinforcement Learning to maximize the score.

- To add variety and complexity, some developers created 2048 game variants with larger grids (like 5x5 or 6x6), higher goal tiles (4096, 8192), or new gameplay twists.
- Some developers have created command-line or desktop-based versions of 2048, enabling players to enjoy the game on different platforms outside the web and mobile environments.

2.3. Bibliometric analysis:

The 2048 game, since its creation in 2014 by Gabriele Cirulli, has inspired a significant amount of research and development work across areas like artificial intelligence, algorithm optimization, user experience design, and cognitive gaming. This bibliometric analysis covers the extent and focus of academic research and publications related to the 2048 game, providing insights into its impact across various fields.

- The initial surge of publications began shortly after the game's release, as researchers and developers recognized the potential for algorithmic and AI applications within the game.
- A substantial portion of research on 2048 centres on developing AI agents capable of achieving high scores or reaching the 2048 tile consistently.
- Some of the most-cited publications are those focused on AI algorithms, particularly Expect Imax search and reinforcement learning approaches.
- Collaborative works between universities and research institutions highlight the game's use as a teaching tool in computer science courses, particularly for introducing concepts in AI and algorithms.
- The game's popularity as a problem-solving exercise has led to its adoption in AI and machine learning training.
- Beyond AI, research into 2048 has influenced studies on gamification and its impact on cognitive skills, with educational psychology and behavioural science researchers exploring how puzzle games can develop strategic thinking and adaptive problem-solving skills.

2.4. Review Summary:

The 2048 game has become an influential topic within academic and development communities due to its combination of simple mechanics and challenging gameplay.

- The 2048 game's appeal lies in its minimalist design and challenging yet accessible gameplay.
- Studies emphasize the addictive nature of 2048, attributing this to its straightforward goal and incremental progress.
- A significant body of research has been dedicated to developing AI solutions for 2048, as the game presents an ideal environment for testing algorithms related to decision-making, optimization, and machine learning.
- Commonly used algorithms include Expect Imax, Minimax, and reinforcement learning, each with varying levels of success in achieving high scores.
- Research on optimal strategies for 2048 has yielded a variety of heuristic-based methods aimed at guiding player choices.
- The review of heuristic research highlights how players can enhance their success through strategic movement patterns.
- Researchers and educators have recognized 2048 as an educational tool for introducing students to concepts in AI, algorithms, and game theory.

2.5. Problem Definition:

The 2048 game project presents several unique challenges that arise from its simple yet complex nature. The goal of this project is to accurately recreate the gameplay mechanics and user experience of the original 2048 game while addressing issues related to algorithmic efficiency, randomness, and user interaction. The fundamental problem lies in replicating the game's tile-merging mechanics within a 4x4 grid, where each move shifts tiles and merges those of equal value. Implementing smooth and accurate movement that maintains the game's logic is critical to achieving an authentic 2048 experience. To enhance replay ability, the project should include an option to restart the game, ensuring players can start a new session without technical issues.

2.6. Goals/Objectives:

The key goals and objectives are outlined below:

- Accurately implement the tile-merging mechanics, including the rules for moving tiles and merging those of equal value.
- Develop a robust system for generating new tiles (2s and 4s) in accordance with probability guidelines (90% chance for a 2 and 10% for a 4).
- Design a clean, responsive user interface that enhances gameplay and provides clear feedback for user actions.
- Implement reliable input detection for keyboard and/or swipe gestures, allowing users to control the game intuitively.
- Develop a mechanism to accurately determine when no further moves are possible, triggering the "Game Over" state.
- Ensure the game runs smoothly on a range of devices, optimizing resource management and performance.

CHAPTER 3.
DESIGN FLOW/PROCESS

3.1. Evaluation & Selection of Specifications/Features:

The development of the 2048 game necessitates careful evaluation and selection of specifications and features that will define its functionality, performance, and overall user experience. The following considerations outline the criteria for selecting key specifications and features for the game:

1. **Core Game Mechanics:** The game must accurately replicate the tile merging rules, including movement mechanics (tiles slide in the direction of user input) and merging identical tiles to create higher-value tiles.
2. **Random Tile Generation:** The game should include a reliable system for generating new tiles after each valid move, adhering to specific probabilities (90% for 2 and 10% for 4).
3. **User Interface Design:** The UI must be clean, responsive, and visually appealing, featuring a grid display for the tiles, score tracking, and clear indicators for game status (e.g., Game Over).
4. **Input Handling:** The game must support intuitive input methods, including keyboard controls (arrow keys) and swipe gestures for touch devices.
5. **Game Over Detection:** The game should accurately identify when no further moves are possible, triggering a Game Over state and presenting players with options to restart or view scores.
6. **Performance Optimization:** The game must be optimized to run smoothly across various platforms, minimizing resource usage and preventing lag.
7. **Additional Engagement Features:** If feasible, additional features such as an undo option, high-score leaderboard, and alternative game modes should be considered.

3.2. Design Constraints:

The development of the 2048 game is subject to various design constraints that can influence the project's scope, implementation, and overall success. Understanding these

constraints is crucial for making informed decisions throughout the development process. The key design constraints for the 2048 game project include:

1. **Technical Constraints:** The game must be designed to function seamlessly across multiple platforms (e.g., web browsers, mobile devices), necessitating adherence to web standards and responsive design principles.
2. **Time Constraints:** The project must be completed within a predefined timeline, which can limit the extent of features implemented and the depth of testing performed. Prioritization of core functionalities over optional features will be necessary to meet deadlines.
3. **Resource Constraints:** The project may be constrained by the availability of human resources, such as developers, designers, and testers, which can affect the speed and quality of implementation.
4. **User Experience Constraints:** The game must be designed with accessibility in mind, ensuring that it is usable for players with varying abilities, including considerations for color contrast, keyboard navigation, and screen reader compatibility.
5. **Regulatory Constraints:** The game design must respect existing intellectual property rights, avoiding the direct use of copyrighted assets or trademarked elements from other games. Original graphics, sounds, and branding will be required.
6. **Testing and Quality Assurance Constraints:** Limited time and resources may restrict the extent of testing, including user acceptance testing and performance testing. Prioritizing high-risk areas will be necessary to ensure a stable final product.

3.3. Analysis of Features and finalization subject to constraints:

The development of the 2048 game requires a careful analysis of potential features in light of the identified design constraints. Below is a breakdown of key features, their benefits, and how they align with the constraints:

1. **Core Gameplay Mechanics:** Tile movement and merging functionality.
2. **Random Tile Generation:** Randomly generating new tiles (2s and 4s) with specified probabilities.
3. **User Interface Design:** Clean and responsive UI for displaying the game grid, scores, and notifications.
4. **Input Handling:** Support for keyboard controls and swipe gestures.

5. **Game Over Detection:** Mechanism to detect when no further moves are possible and to trigger the Game Over state.
6. **Performance Optimization:** Optimizing the game to run smoothly on various devices.

3.4. Design Flow:

The design flow outlines the systematic process of developing the 2048 game, from initial concept to final implementation. It includes key phases, decision points, and interactions between different components of the game. Below is a detailed overview of the design flow:

1. **Requirements Gathering**
    - Identify core gameplay, UI, and input needs.
    - Output: Detailed requirements list.
2. **Feature Analysis & Prioritization**
    - Evaluate features for feasibility and impact.
    - Output: Prioritized list: Essential, Desirable, Optional.
3. **Prototyping**
    - Build a basic prototype with key mechanics and UI.
    - Output: Interactive testable version.
4. **Core Development**
    - Implement gameplay: tile movement, merging, input, random generation, game over logic.
    - Output: Fully playable version.
5. **Testing & QA**
    - Unit, integration, and user testing.
    - Output: Bug-free, stable build with documented test results.
6. **Feature Enhancements (Optional)**
    - Add undo, leaderboard, new game modes.
    - Output: Improved player engagement.
7. **Deployment**
    - Launch on target platforms (web/mobile).
    - Output: Public game with marketing support.

3.5. Design selection:

The design selection process for the 2048 game involves evaluating various design alternatives and choosing the most suitable approaches based on the project's goals, constraints, and user requirements.

1. **Gameplay Mechanics Design**
    - **Selected Design**: Classic 2048 mechanics with merging tiles and score accumulation.
2. **User Interface Design**
    - **Selected Design**: Simple and intuitive grid layout with distinct visual cues for tile values
      and game state.
3. **Input Handling Design**
    - **Selected Design**: Support for both keyboard controls (arrow keys) and touch gestures (swipes).
4. **Game Logic and Algorithms Design**
    - **Selected Design**: Use of a grid-based data structure to represent the game state, with functions for tile movement, merging, and random generation.
5. **Performance Optimization Design**
    - **Selected Design**: Techniques such as lazy loading, resource caching, and efficient rendering methods.
6. **Technology Stack Selection**
    - **Frontend**: HTML5, CSS3, and JavaScript for web implementation.
    - **Libraries/Frameworks**: Use of libraries such as React for managing UI components, if deemed necessary for future scalability.
7. **Testing and Quality Assurance Design**
    - **Selected Approach**: Automated testing combined with user acceptance testing.

3.6. Implementation plan/methodology:

The implementation plan for the 2048 game outlines the structured approach to translating the design into a functional product. The implementation will follow an Agile methodology,

promoting iterative development and flexibility in response to feedback.

1. **Project Kickoff**
    - Assemble a development team with defined roles (e.g., project manager, developers, designers).
    - Conduct a kickoff meeting to clarify project scope, timelines, and deliverables.
2. **Prototype Development**
    - Develop the initial game logic for tile movement and merging.
    - Implement a basic user interface for the game grid and score display.
    - Conduct internal testing to identify any major issues.
3. **User Interface Enhancement**
    - Design and implement visual elements such as tile colors, animations, and score displays.
    - Ensure responsive design for compatibility across devices.
    - Gather feedback from team members and stakeholders for improvements.
4. **Testing and Quality Assurance**
    - Perform automated unit tests for key components.
    - Conduct integration testing to ensure all features work together seamlessly.
    - Run user acceptance testing sessions to gather feedback on gameplay and identify any remaining issues.
5. **Feature Enhancements and Final Adjustments**
    - Add optional features such as an undo option or high-score leaderboard.
    - Make final adjustments to gameplay mechanics and user interface based on user feedback.
6. **Deployment**
    - Finalize documentation for the game, including user guides and technical documentation.
    - Deploy the game to the chosen platform (e.g., web server, app store).
    - Prepare promotional materials to announce the game's launch.

- **Detailed Block Diagram:**

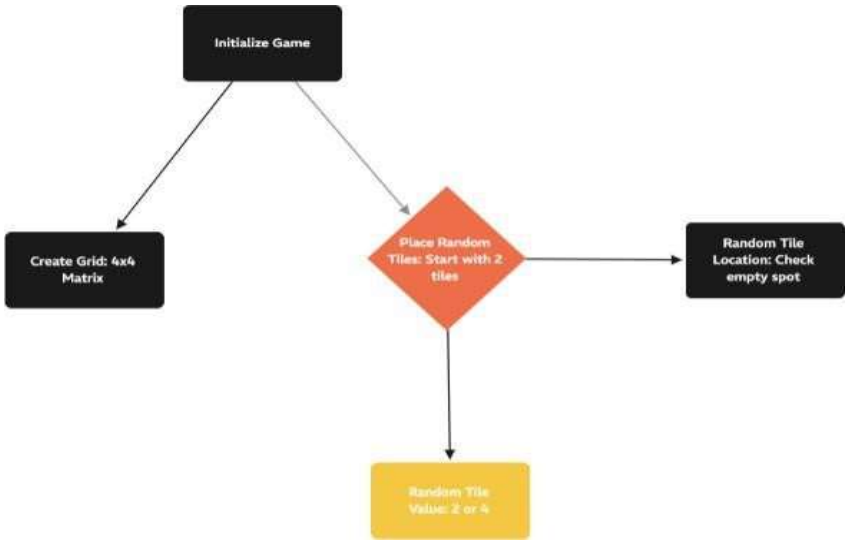


Fig. 1(a). Implementation flowchart stage 1

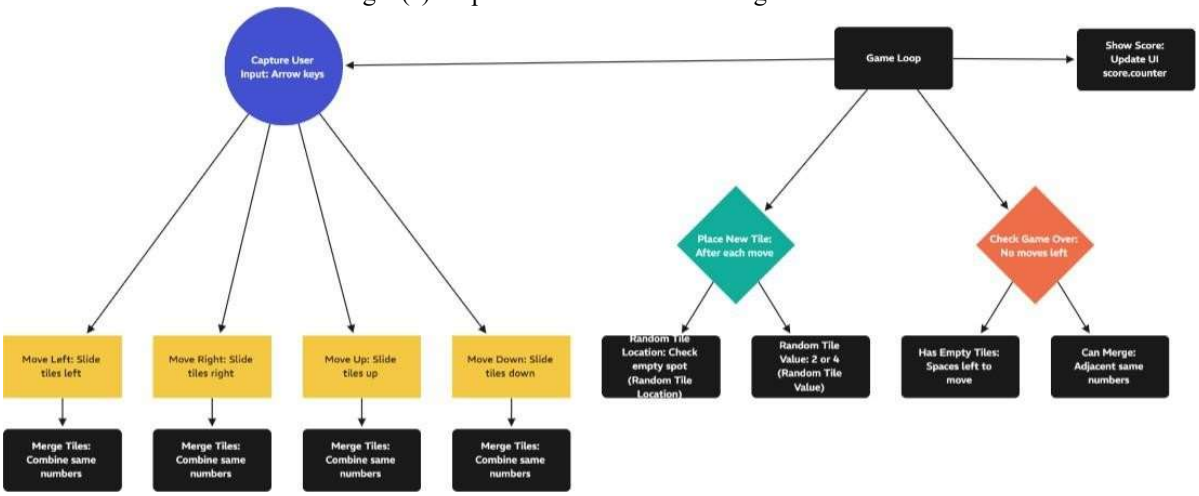


Fig. 1(b). Implementation flowchart stage 2

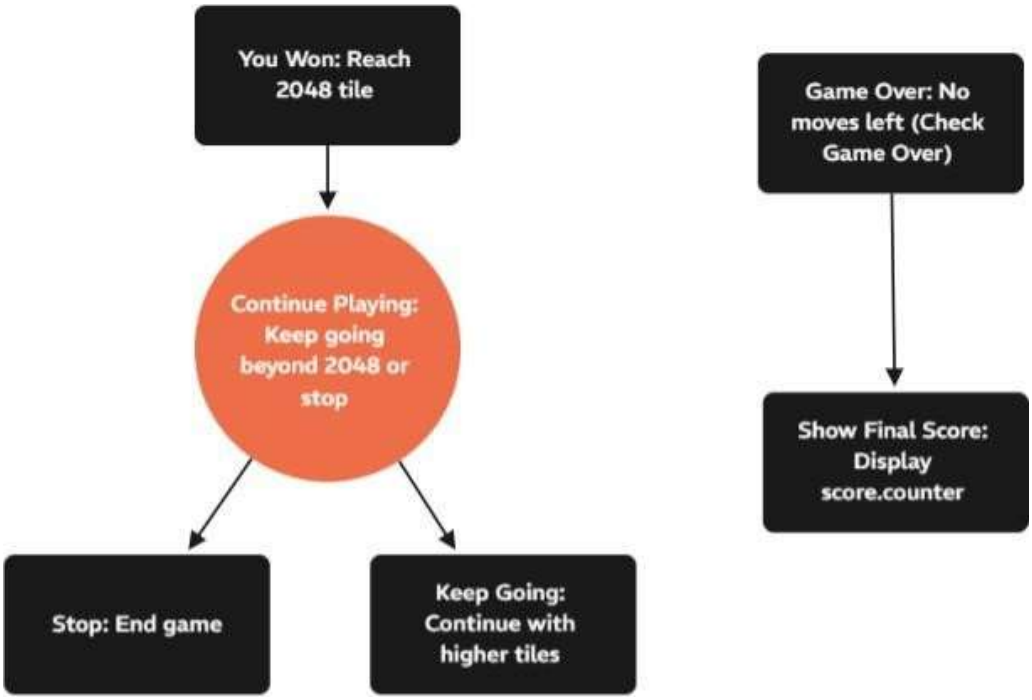


Fig. 1(c). Implementation flowchart stage 3

CHAPTER 4.
RESULTS ANALYSIS AND VALIDATION

4.1. Result Outcomes:

Fig. 2(a). Result Outcome 1

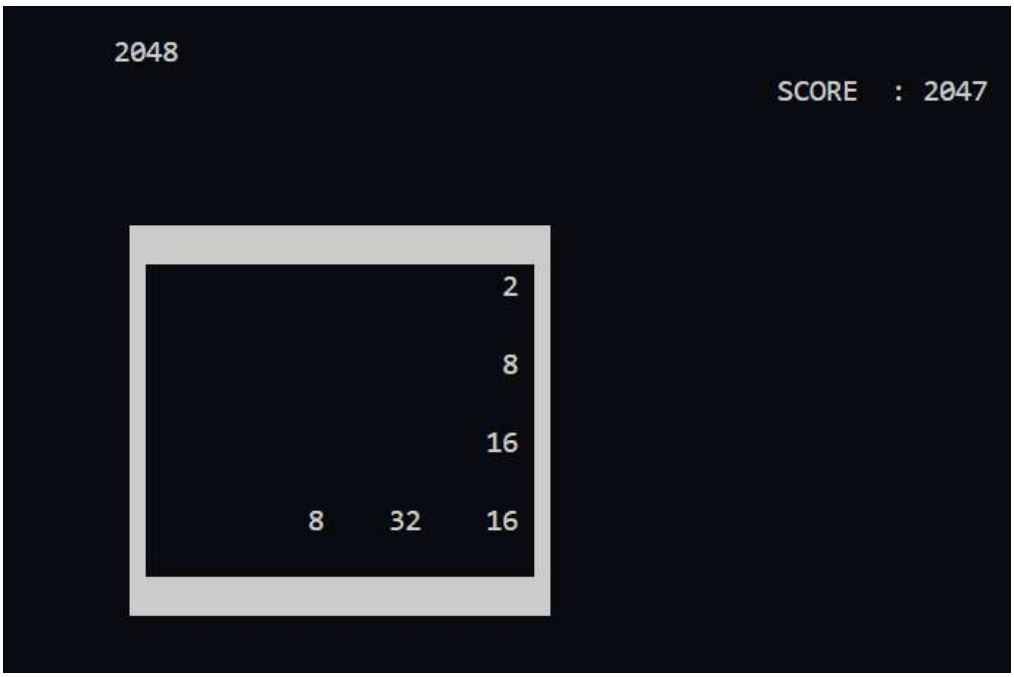


Fig. 2(b). Result Outcome 2

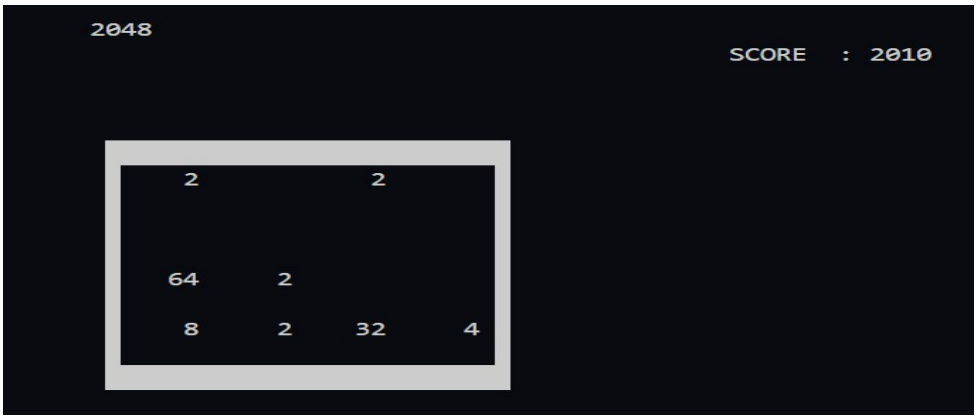


Fig. 2(c). Result Outcome 3

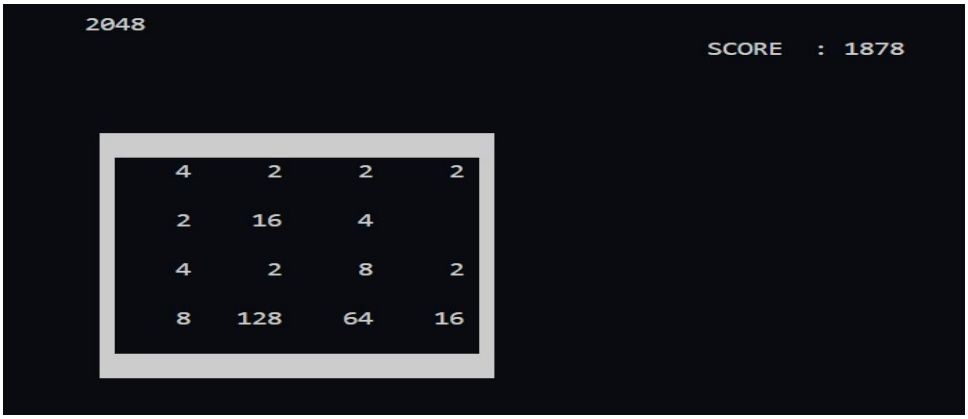


Fig. 2(d). Result Outcome 4

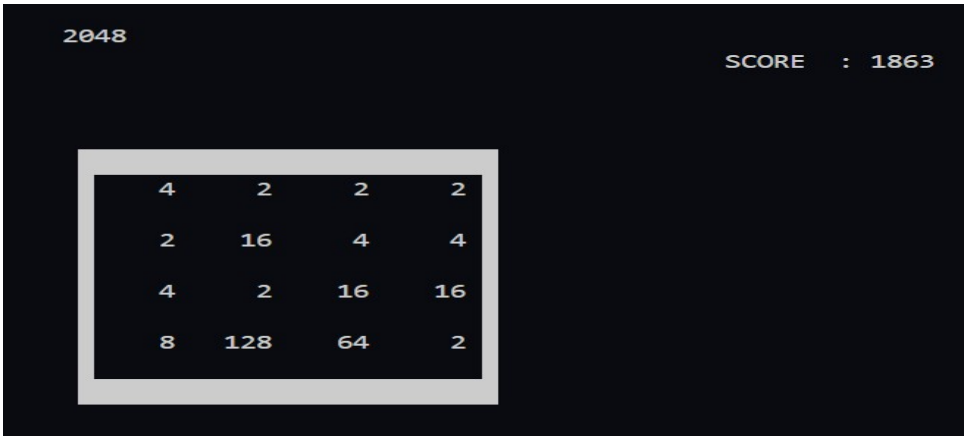


Fig. 2(e). Result Outcome 5

CHAPTER 5.
CONCLUSION AND FUTURE WORK

5.1. Conclusion:

The development of the 2048 game represents a comprehensive and structured approach to creating a classic puzzle game that has captivated audiences worldwide. Through meticulous planning, implementation, and testing, the project has successfully navigated the complexities of game development, demonstrating key principles of software engineering and design.

From the initial identification of client needs to the detailed implementation plan, each phase of the project has been designed to ensure a user-centric and engaging gameplay experience. The emphasis on familiar mechanics, intuitive user interface design, and performance optimization has been pivotal in achieving the project's goals.

In conclusion, the successful development of the 2048 game serves as a testament to the effectiveness of structured project management, user-cantered design, and a commitment to quality. The project not only meets its initial objectives but also lays the groundwork for potential future enhancements and expansions. As the game is launched, ongoing support and maintenance will be critical in ensuring its continued success and relevance in the competitive gaming landscape. With user engagement and feedback, the game can evolve, introducing new features and maintaining its status as a beloved classic in the gaming community.

5.2. Future work:

As While the current implementation of the 2048 game has successfully met its initial objectives, there are several avenues for future work that could enhance the game's features, performance, and overall user experience. The following suggestions outline potential directions for further development:

1. Enhanced Gameplay Features

- **Multiple Game Modes**: Introduce various gameplay modes, such as timed challenges, multiplayer options, or variations with different tile mechanics (e.g., larger grids or additional special tiles).
- **Customizable Difficulty Levels**: Allow players to select different difficulty settings, affecting the size of the grid, initial tile placements, or tile values.

2. User Experience Improvements

- **Advanced User Interface**: Revamp the user interface to include animations, sound effects, and visual feedback that enhance player engagement.
- **Accessibility Features**: Introduce options for colorblind modes, adjustable text sizes, and voice commands to make the game more inclusive for players with disabilities.

3. Cross-Platform Compatibility

- **Mobile App Development**: Develop native mobile applications for iOS and Android platforms to reach a broader audience and provide a more optimized experience on smartphones and tablets.
- **Web Version Enhancements**: Improve the web version with progressive web app (PWA) capabilities, allowing users to install the game on their devices directly from the browser.

4. Community Engagement and Social Features

- **Leaderboards**: Implement global and friends' leaderboards to foster competition among players and encourage sharing of high scores on social media.
- **In-Game Challenges**: Introduce daily or weekly challenges that players can complete for rewards, promoting regular engagement with the game.

5. Analytics and Feedback Integration

- **User Analytics**: Integrate analytics tools to gather data on user behavior, preferences, and pain points. This information can guide future updates and enhancements.
- **Feedback Mechanism**: Establish a system for players to submit feedback directly through the game, allowing for continuous improvement based on user input.

6. Expansion into Other Platforms

- **Desktop Versions**: Consider developing desktop versions for popular operating systems, incorporating features specific to those platforms.
- **Integration with Other Games**: Explore possibilities for integrating 2048 with other popular games or platforms, potentially creating a crossover experience that attracts new players.

.

REFERENCES

- **GeeksforGeeks (GFG)**
  - Title: "2048 Game Implementation"
  - Description: A comprehensive guide on how to implement the 2048 game using various programming languages.
  - URL: GeeksforGeeks - 2048 Game Implementation
- **JavaTpoint**
  - Title: "2048 Game in Java"
  - Description: This tutorial explains how to create the 2048 game in Java with step-by-step instructions.
  - URL: JavaTpoint - 2048 Game in Java
- **Medium**
  - Title: "Building the 2048 Game with JavaScript"
  - Description: A detailed article that walks through the process of building the 2048 game using HTML, CSS, and JavaScript.
  - URL: Medium - Building the 2048 Game with JavaScript
- **FreeCodeCamp**
  - Title: "How to Build the 2048 Game with HTML, CSS, and JavaScript"
  - Description: A step-by-step tutorial on how to create the 2048 game from scratch using web technologies.
  - URL: FreeCodeCamp - How to Build the 2048 Game
- **MDN Web Docs**
  - Title: "Using the HTML5 Canvas"
  - Description: Documentation on the HTML5 Canvas API, which can be useful for rendering graphics in the 2048 game.
  - URL: MDN Web Docs - Using the HTML5 Canvas
- **YouTube**
  - Title: "2048 Game Tutorial | Learn JavaScript"
  - Description: A video tutorial that provides a visual guide to implementing the 2048 game in JavaScript.
  - URL: YouTube - 2048 Game Tutorial