**PURBANCHAL UNIVERSITY**

**Biratnagar, Nepal**

****

A Project report on

**“RETRO SMASH”**

In the partial fulfillment for the requirement of the 4th Semester Project-IV (BIT 256 CO) in the completion of **Bachelor of Information Technology (BIT)** degree at **KIST college of Information Technology**, under **Purbanchal University.**

**Submitted By: Submitted To: 1) Deepak Shes Bk Purbanchal University**

**2) Pawan Thapa**

**3) Pravat Nagarkoti**

**Under The Guidance of**

**Mr. Roshan Shrestha**

**Project Teacher**

**KIST COLLEGE OF INFORMATION AND TECHNOLOGY KAMALPOKHARI, KATHMANDU**

**KIST COLLEGE OF INFORMATION AND TECHNOLOGY KAMALPOKHARI, KATHMANDU**

****

**CERTIFICATE**

This is to certify that the project work entitled **“RETRO SMASH”** is carried out **by DEEPAK SHES BK(6100), PAWAN THAPA(6091), PRAVAT NAGARKOTI(6084),** bona fide students of **KIST COLLEGE OF INFORMATION AND TECHNOLOGY** in partial fulfillment for the award of **BACHELOR IN INFORMATION AND TECHNOLOGY** of the **PURBANCHAL UNIVERSITY, BIRATNAGAR NEPAL**, during the year **2024**. It is certified that all corrections indicated for internal assessment have been incorporated in the report submitted in the department library. The project report has been approved, as it satisfied the academic requirements in respect of the project work prescribed for the said degree.

The details of the students are as follows: -

**NAME REGISTRATION SYMBOL NO.**

Deepak Shes Bk 058-3-2-03788-2021 340382

Pawan Thapa 058-3-2-03797-2021 340390

Pravat Nagarkoti 058-3-2-03800-2021 340393

Course: - BIT (Bachelor’s in Information Technology)

Semester: - 4th Semester

Subject: - Project-IV

Subject Code: - BIT 256 CO



Mr. Deepak Khadka (Co-Ordinator)

**KIST COLLEGE OF INFORMATION AND TECHNOLOGY KAMALPOKHARI, KATHMANDU**

****

**Examiner’s Certification**

The Project Report

On

**“RETRO SMASH”**

**Developed by**

**Deepak Shes Bk**

**Pawan Thapa**

**Pravat Nagarkoti**

Is approved and is acceptable in qualified form.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Internal Examiner External Examiner Name**: Roshan Shrestha Name:

Designation: Designation:

**ACKNOWLEDGEMENT**

It is with greatest satisfaction and euphoria that we are submitting our project report entitled “**Retro Smash”.** We have completed it as a part of the curriculum of **PURBANCHAL UNIVERSITY.**

We also take this opportunity to express a deep sense of gratitude to our **BIT Coordinator Mr. Deepak Khadka** and **Project Teacher Mr. Roshan Shrestha** for their amiable support, valuable information and guidance which helped us in completing this task throughout its various stages. We are indebted to all members of **KIST College,** for the valuable support and suggestions provided by them using their specific fields’ knowledge. We are grateful for their cooperation during the period of our project.

Finally, we would also like to express our gratitude towards **Purbanchal University** for designing such a wonderful course structure. It will help us to get more knowledge in the field of Information Technology & help us to have a bright future in the field of technology.

We hope our university will accept this attempt as a successful project.

Last but not the least, our sincere thanks to our parents, teaching and non-teaching staff of our college and our friends.

**Deepak Shes Bk (340382)**

**Pawan Thapa (340390**)

**Pravat Nagarkoti (340393)**

**ABSTRACT**

This documentation provides an in-depth overview of Retro Smash, a classic-inspired arcade game that combines elements of nostalgia with modern gameplay mechanics. Designed for both seasoned gamers and newcomers, Retro Smash offers a unique experience that captures the essence of retro gaming while incorporating contemporary features.

The documentation includes detailed sections on game design, mechanics, and development processes. It covers the game's core components such as character design, level architecture, sound design, and user interface. Additionally, it explores the technical aspects of the game's development, including the programming languages used, the game engine selection, and the integration of graphics and sound assets.

Furthermore, this documentation provides insights into the game's testing and deployment phases, user feedback integration, and post-launch updates. It also includes appendices with troubleshooting tips, frequently asked questions, and a glossary of key terms.

Whether you are a developer looking to understand the technical intricacies of Retro Smash, or a player interested in the game's design philosophy, this comprehensive documentation serves as a valuable resource. Join us in exploring the journey behind creating a game that bridges the past and present of arcade gaming.

**LIST OF FIGURES:**

**Figure 1.1:** Flow Chart

**Figure 1.2:** UML Diagram

**Figure 1.3:** Spiral Model for SDLC

**Figure 1.4:** Gantt Chart

**TABLE OF CONTENTS**

**Approval Certificate**

**Acknowledgements**

**Abstract**

**List of Figures**

**Chapter 1: Introduction**

1.1 Introduction

1.2 Problem Statement

1.3 Project Objectives

1.4 Scope

**Chapter 2: System Analysis**

2.1 Literature Review

**Chapter 3: System Design**

3.1 Algorithm

3.2 Flowchart

3.3 UML diagram

3.3 Spiral Model

3.5 Gantt Chart

**Chapter 4: REQUIREMENT ANALYSIS AND SYSTEM IMPLEMENTATION**

4.1 Hardware and Software Requirements

4.2 System Methodology

**Chapter 5: CONCLUSION AND FUTURE SCOPE**

5.1 Conclusion

5.2 Future Scope

**References**

**SNAPSHOTS**

**Chapter 1**

# Introduction

**1. Introduction**

The game "RETRO SMASH" is a captivating arcade-style game designed to provide an engaging and nostalgic gaming experience. This document aims to provide a comprehensive overview of the game, detailing its development process, key features, and underlying architecture.

**1.2 Problem Statement**

In the current gaming market, there is a growing demand for engaging and nostalgic games that offer both entertainment and a sense of accomplishment. However, many classic arcade-style games lack modern features such as online leaderboards, multiplayer capabilities, and customizable elements. "Retro Smash" aims to bridge this gap by providing a retro-themed arcade game that combines the charm of classic gameplay with contemporary enhancements, creating a more captivating and competitive gaming experience.

**1.3 Objectives**

The primary objectives of "RETRO SMASH" are:

* To create an engaging and challenging game that appeals to a wide audience.
* To implement smooth and responsive gameplay mechanics.
* To provide entertainment and enjoyment for the players of all ages.
* To integrate a competitive element through a dynamic leaderboard.
  1. **Project features**

The main features of the game are as follows:

* **Player Controls:** The player can move their character left, and right using the arrow keys.
* **Dynamic Scoreboard:** The scores on leaderboard are continuously updated as the game progresses.
* **Collision Detection:** If the ball collides with the paddle or walls, the game ends, it’s direction changes accordingly. The player can check if they made it to the leaderboard from the main menu..

**Chapter 2**

**System Analysis**

**2.1 Literature Review**

Retro Smash is designed with inspiration drawn from the classic arcade games of the 1980s and 1990s, such as Pac-Man, Donkey Kong, and Space Invaders, which established foundational game design principles characterized by simple controls, challenging gameplay, and distinctive visual and audio styles. These principles, often summarized as "easy to learn, hard to master," ensure that games are accessible to new players while offering depth for experienced gamers. The psychological impact of nostalgia is significant in gaming, enhancing player engagement and satisfaction by evoking fond memories. This is supported by market trends indicating a resurgence of interest in retro gaming, demonstrated by the success of platforms like Nintendo's NES Classic Edition and retro-inspired games like Shovel Knight. Modern game design benefits from combining retro aesthetics with contemporary mechanics, such as advanced AI, smoother animations, and online multiplayer modes, creating a compelling experience that appeals to both veteran and new gamers. Effective user interface design is crucial, with studies emphasizing the importance of intuitive controls and clear visual feedback to enhance immersion. Technically, versatile game engines like Unity and Unreal Engine are essential for creating visually appealing and performance-optimized games, supporting cross-platform development crucial for broad audience reach. Core components of retro games, such as pixel art and chiptune music, significantly contribute to their nostalgic appeal, preferred by players seeking authentic experiences. Market analysis reveals that the retro gaming demographic includes both older gamers who experienced the original arcade era and younger players attracted to the unique aesthetics and challenging gameplay. Successful retro games often employ diverse revenue models, including one-time purchases, in-game purchases, and limited-edition physical releases, enhancing profitability and sustainability. By understanding these historical, psychological, technical, and market factors, Retro Smash aims to blend the best of past and present gaming, catering to nostalgic gamers and attracting new players with a fresh yet familiar experience

**Chapter 3**

# SYSTEM DESIGN

**3.1 Algorithm**

### Step-by-Step Algorithm for the Game

1. **Start Program**
   * Initialize the game application.
   * Display the main menu.
2. **Main Menu**
   * Display options:
     + Start Game
     + View Leaderboard
     + Exit
3. **Start Game**
   * **Initialize Game**
     + Load game assets (graphics, sounds, etc.).
     + Initialize game variables (score, player position, etc.).
     + Display the game screen.
   * **Gameplay Loop**
     + Start the background music.
     + While the game is running:
       - Capture user input (keyboard, mouse, etc.).
       - Update game state based on user input.
       - Update game elements (characters, enemies, obstacles, etc.).
       - Detect collisions and handle them.
       - Render the updated game screen.
     + Stop the background music when the game ends.
   * **End Game**
     + Display the game over screen with the player's final score.
     + Prompt the user to enter their name for the leaderboard.
     + Save the player's score and name to the leaderboard.
     + Provide options to:
       - Return to the main menu
       - Exit
4. **View Leaderboard**
   * Retrieve and display the list of top scores with player names.
   * Provide options to:
     + Return to the main menu
     + Exit
5. **Exit Program**
   * Terminate the game application.

**3.2 Flowchart**

## 

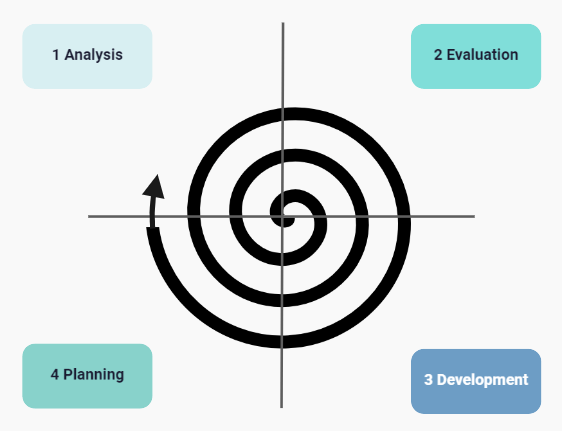
**Figure 1.1: Flow Chart**

## 3.3 UML Diagram

## 

**Figure 1.2: UML Diagram**

**3.4** **Spiral Model**

****

**Figure 1.3: Spiral Model for SDLC**

"Retro Smash" follows the Spiral Model as follows:

**1. Analysis**

In the analysis phase, the game's core mechanics—paddle and ball movement, collision detection, and scoring—are identified. Additional features like leaderboards and background music are determined, and a feasibility study assesses the technical requirements and potential performance issues.

**2. Evaluation**

The evaluation phase focuses on identifying and mitigating risks such as performance bottlenecks and UI challenges. Prototypes of core mechanics and UI mock-ups are developed, followed by initial user testing to gather feedback and make adjustments.

**3. Development**

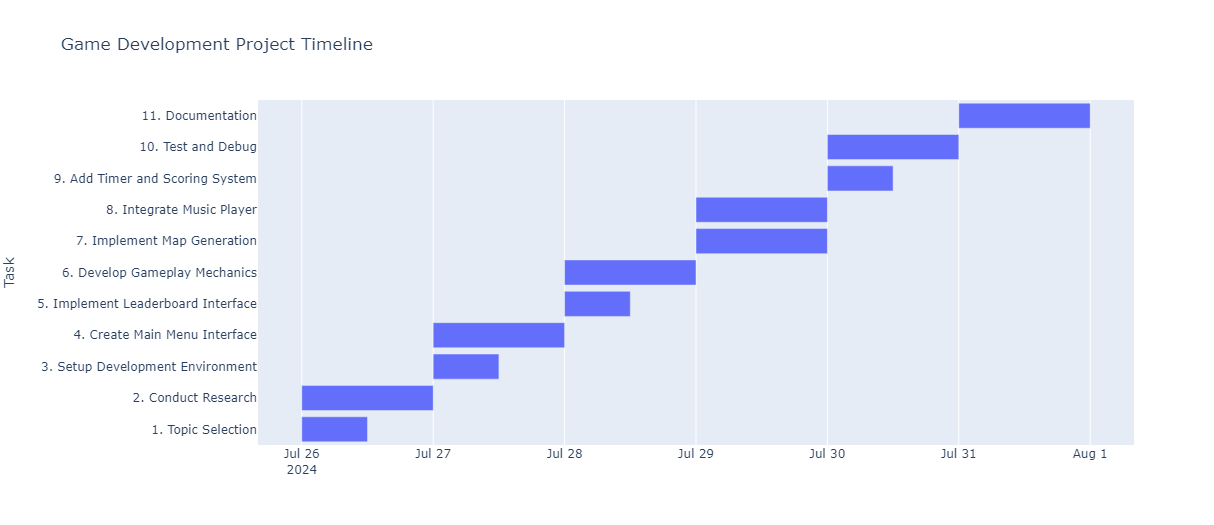
During development, the game is built iteratively, incorporating feedback from the evaluation stage. Core mechanics, UI elements, and additional features are implemented. Each iteration focuses on refining the game based on continuous feedback to meet user expectations.

**4. Planning**

In the planning phase, feedback from the current iteration is reviewed, and the project plan is updated. Goals for the next development cycle are set, and resources are allocated to ensure smooth progress in the next iteration.

**3.5 Gantt chart**

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Start Date | End Date | Duration(in days) |
| Topic Selection | 7/26/2024 | 7/26/2024 | 0.5 |
| Conduct Research | 7/26/2024 | 7/27/2024 | 1 |
| Setup Development Environment | 7/27/2024 | 7/27/2024 | 0.5 |
| Create Main Menu Interface | 7/27/2024 | 7/28/2024 | 1 |
| Implement Leaderboard Interface | 7/28/2024 | 7/28/2024 | 0.5 |
| Develop Gameplay Mechanics | 7/28/2024 | 7/29/2024 | 1 |
| Implement Map Generation | 7/29/2024 | 7/30/2024 | 1 |
| Integrate Music Player | 7/29/2024 | 7/30/2024 | 1 |
| Add Timer and Scoring System | 7/30/2024 | 7/30/2024 | 0.5 |
| Test and Debug | 7/30/2024 | 7/31/2024 | 1 |
| Documentation | 7/31/2024 | 8/1/2024 | 1 |



**Figure 1.4: Gantt Chart**

**Chapter 4**

# REQUIREMENT ANALYSIS AND SYSTEM IMPLEMENTATION

## 4.1 Hardware and Software Requirements

**Hardware Requirements:**

* **For Development:**
  + **Processor:** Intel Core i5 or AMD Ryzen 5
  + **Memory:** 8 GB RAM (16 GB recommended)
  + **Graphics Card:** NVIDIA GTX 1050 or AMD Radeon RX 560
  + **Storage:** 256 GB SSD (512 GB recommended)
* **For End Users:**
  + **Processor:** Intel Core i3 or AMD Ryzen 3
  + **Memory:** 4 GB RAM (8 GB recommended)
  + **Graphics Card:** Integrated graphics (NVIDIA GTX 960 or AMD Radeon R7 370 recommended)
  + **Storage:** 1 GB free disk space

### Software Requirements

**For Development:**

* **IDE:** IntelliJ IDEA or Eclipse
* **Programming Languages:** Java
* **Version Control:** Git
* **Database:** MySQL
* **Audio Library:** Java Sound API

**For End Users:**

* **Operating Systems:** Windows 7 or later, macOS 10.10 or later, or any modern Linux distribution
* **Java Runtime Environment:** JRE 8 or later

**4.2 System Methodology**

**1. Planning:**

* Objective: Define the project’s goals, scope, and deliverables.
* Activities: Conduct initial research, outline game concepts, establish a project timeline, and allocate resources.

**2. Design:**

* Objective: Create a detailed design for the game's features and interface.
* Activities: Develop game mechanics, design character models and levels, and create the user interface. Build prototypes to test and refine these elements.

**3. Development:**

* Objective: Implement the game according to the design specifications.
* Activities: Code game logic in Java, integrate graphics and sound, and develop gameplay features. Ensure all components function seamlessly together.

**4. Testing:**

* Objective: Ensure the game is functional and meets quality standards.
* Activities: Conduct unit testing, integration testing, and user acceptance testing to identify and address bugs and refine gameplay.

**5. Deployment:**

* Objective: Release the game to users.
* Activities: Prepare the game for release, handle packaging and distribution, and execute marketing strategies. Deploy the game on the intended platform.

**6. Maintenance:**

* Objective: Provide ongoing support and updates for the game.
* Activities: Monitor player feedback, release patches, and update the game with new content or improvements based on user input and performance.

# Chapter 5

**Conclusion & Future Recommendation**

**5.1 Conclusion**

Retro Smash successfully blends the charm of classic arcade games with modern gameplay innovations, offering a unique experience that appeals to both nostalgic gamers and newcomers. Through careful attention to game design, technical execution, and user feedback, Retro Smash has created a compelling and engaging gaming experience. The combination of familiar retro aesthetics with contemporary features ensures that the game stands out in the crowded gaming market.

The project has demonstrated that a well-executed blend of nostalgia and modernity can capture a diverse audience, from those who cherish the classics to those seeking fresh, dynamic gameplay. The development process, guided by a structured methodology, has addressed key aspects such as design, functionality, and performance, resulting in a polished final product.

Looking forward, Retro Smash is positioned for success with its solid foundation and ongoing commitment to enhancing player experience. Continuous updates and responsive support will help maintain player interest and engagement, securing the game’s place in the evolving landscape of arcade gaming.

**5.2 Future Scope**

 **Add New Levels and Characters:** Introduce additional game levels and new characters with unique abilities to keep the gameplay fresh and engaging for players.

 **Expand Multiplayer and Cross-Platform Play:** Enhance multiplayer features and allow players to connect and compete across different gaming platforms, broadening the player base.

 **Release on More Platforms:** Make Retro Smash available on additional platforms, such as newer consoles and mobile devices, to reach a wider audience.

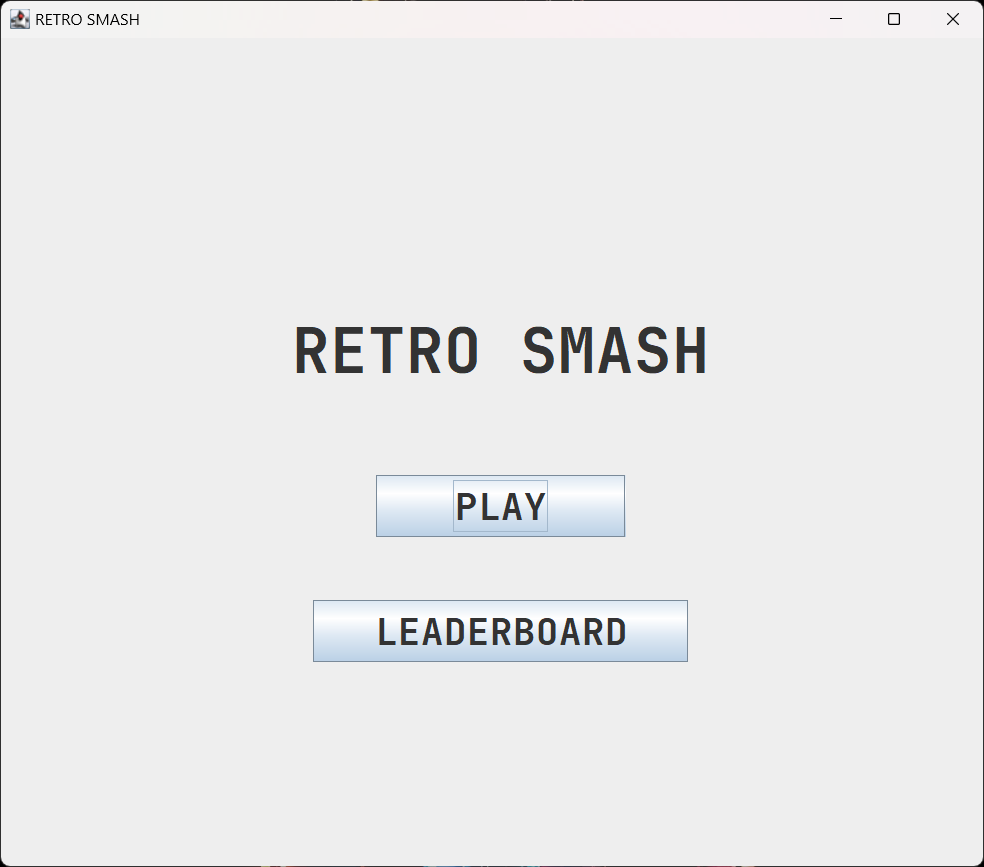
 **Upgrade Graphics and Explore VR/AR:** Improve the game's visual quality and performance, and explore possibilities for virtual or augmented reality to offer a new, immersive gaming experience.

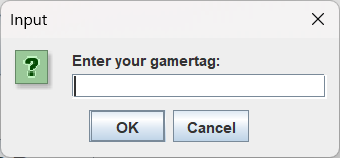
 **Support User-Generated Content and In-Game Events:** Allow players to create and share their own content, such as custom levels or mods, and host special in-game events to foster a vibrant community and keep players engaged.

**References**

|  |  |
| --- | --- |
|  | Websites: [www.google.com](http://www.google.com) & [www.geeksforgeeks.org](http://www.geeksforgeeks.org)  Youtube: <https://youtu.be/K9qMm3JbOH0?feature=shared>  Other references: https://github.com/niyathic/brick-breaker-game |
|  |
|  |
|  |

# SNAPSHOTS



****

