

# 2D Game Development

**Prof. Samita Patil<sup>1</sup> and Soham Sorte<sup>2</sup>**

Faculty, Department of Computer Engineering<sup>1</sup>

Student, Department of Computer Engineering<sup>2</sup>

Shivajirao S Jondhale College of Engineering, Dombivli(E), Thane, Maharashtra, India

**Abstract:** *The "2D Static Screen Fighting Game" is a project developed aimed at creating an engaging and interactive 2D fighting game for two players to enjoy on the same PC. The game focuses on simple yet challenging gameplay, featuring static screen environments, player-controlled characters capable of jumping, and a weapon-based combat system. This game includes player-controlled characters capable of jumping, each armed with a weapon. The weapon serves as both a tool of offense and defense, adding an extra layer of strategy to the gameplay. When a player successfully lands a hit on their opponent, the adversary will lose one health point. Each player starts with a total of five health points, and the health status is prominently displayed on the screen. As the battle unfolds, players must carefully time their attacks, defend against incoming strikes, and make tactical decisions.*

**Keywords:** Multiplayer options, Intense Battles, Strategy PvP, Tactical

## I. INTRODUCTION

In the dynamic realm of video game development, 2D fighting games have maintained their allure for gamers of all ages. These games, characterized by intense combat, precise control, and strategic gameplay, continue to captivate players worldwide. As the project progresses, it will showcase the student's ability to design, develop, and optimize a video game, providing valuable insights into the complexities of game development while delivering a thrilling gaming experience to players. The aim of this project is to develop a 2D fighting game that invites two players to engage in a battle on the same PC. Unlike sprawling open-world adventures, this game will be set in a static screen environment, ensuring that players are never distracted by navigating extensive landscapes or complex camera controls. Instead, they will be immersed in the thrilling challenge of one-on-one combat.

## II. LITERATURE SURVEY

2D game development is a popular and enduring field, especially among indie and mobile developers. It uses flat graphics (sprites) and simpler mechanics compared to 3D games, making it accessible and efficient.

Key game engines include **Unity**, **Godot**, and **GameMaker Studio**, all offering powerful tools for sprite management, animation, and physics. **Entity-Component-System (ECS)** and **Finite State Machines (FSM)** are commonly used architectures to organize game logic.

2D graphics rely on techniques like **sprite sheets**, **frame-based animation**, and **tweening**. For physics and collision, engines often use libraries like **Box2D**. AI in 2D games typically includes basic behaviors such as pathfinding (e.g., using A\* algorithm) and enemy movement patterns.

Level design focuses on clarity, pacing, and the use of **tilemaps** or **procedural generation**. Trends show growing interest in cross-platform tools and procedural gameplay mechanics.

In summary, 2D game development blends creativity with technical design and remains relevant due to its simplicity, scalability, and broad platform support.

## III. PROBLEM STATEMENT

In the world of entertainment and gaming, 2D fighting games have been a staple for decades, providing players with thrilling and competitive experiences. As a third-year engineering student, your task is to develop a 2D offline fighting game that enables two players to compete against each other on the same PC. The project aims to address several key



challenges and objectives. The project involves creating intuitive controls, character animations, and a dynamic game environment, while also implementing multiplayer functionality. The goal is to provide an immersive and enjoyable gaming experience while overcoming technical and design challenges.

#### IV. METHODOLOGY

##### 1. Conceptualization Idea Generation:

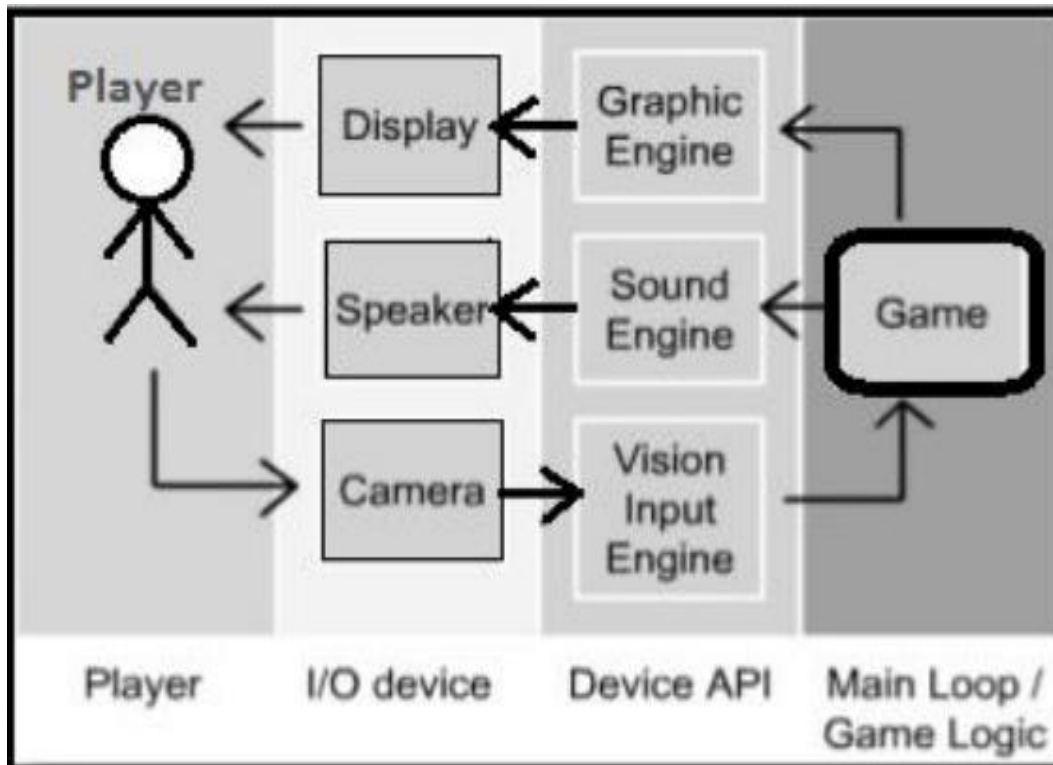
Brainstorm ideas for your game's genre, theme, and core mechanics. Consider what makes your game unique. Target Audience: Define who your game is for based on age, interests, and gaming experience.

##### 2. Pre-production Game Design Document (GDD):

Create a GDD that outlines the game's world, story, characters, gameplay mechanics, levels, art style, music, and sound effects. This document serves as a blueprint throughout the development process. Prototyping: Develop a simple prototype to explore and test gameplay mechanics. This can be done using basic shapes or placeholder graphics.

##### 3. Tools and Technology Choosing a Game Engine:

Select a game engine that is suitable for 2D game development. Popular choices include Unity (with 2D tools), Godot, and GameMaker Studio 2. Art Tools: Decide on tools for creating 2D art. Photoshop, Aseprite, and Adobe Illustrator are common choices. Sound Tools: Choose software for sound production, such as Audacity or FL Studio.



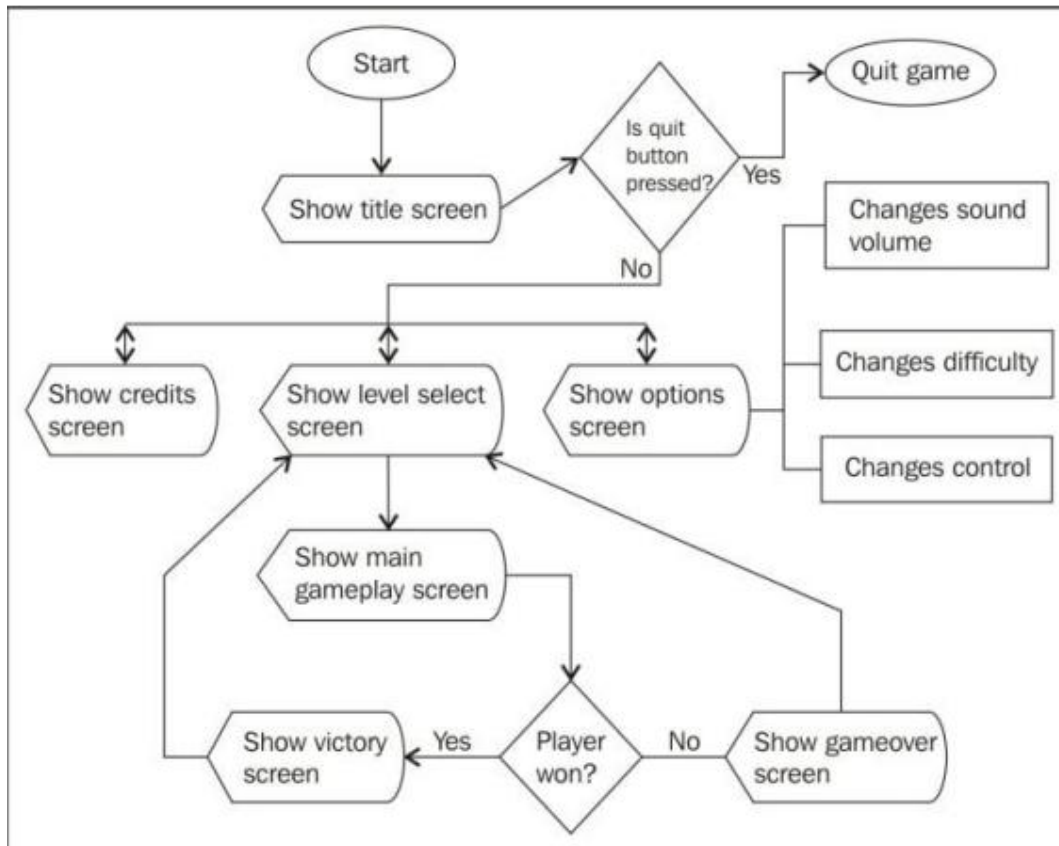


Fig. System Flow

## V. SYSTEM REQUIREMENTS

### Software:

- **Unity Game Engine:** Download and install the latest version of Unity from the official website (<https://unity.com/>).
- **IDE (Integrated Development Environment):** Unity comes with its built-in IDE, but some developers prefer using Visual Studio or Visual Studio Code for scripting.
- **Graphics Software:** Use software like Adobe Photoshop or GIMP for creating and editing sprites and textures. Consider tools like Unity's Animator or external software like Spine or DragonBones for character animations.
- **Sound and Music:** Audacity, Adobe Audition, or any audio editing software for sound effects and music.
- **Version Control:** Git and platforms like GitHub or Bitbucket to manage your source code.

### Hardware:

- **Development Machine:** A decent computer with sufficient RAM, a good CPU, and a dedicated GPU for smooth development.
- **Graphics Tablet (Optional):** If you're creating hand-drawn art, a graphics tablet like Wacon can be beneficial.
- **Testing Devices:** If you plan to release your game on multiple platforms, test it on the devices you intend to support (PC, mobile, etc.).



**V. RESULTS**



Img. Start Frame

Start frame shows the first page of the game.



Img. Main Frame

Main frame shows that how the actual game starts.





Img. End Frame

This frame shows the winner of the game.

## VI. CONCLUSION

In conclusion, the development of our 2D fighting game has been an exciting and rewarding journey. We successfully created an engaging and visually appealing gaming experience that offers players a chance to showcase their combat skills. Throughout the project, we encountered challenges such as character balance, animation integration, interface design, which provided valuable opportunities for problem-solving and learning. Overall, this project has not only allowed us to develop technical and creative skills but has also provided valuable experience in project management and teamwork. We are proud of what we have achieved, and we look forward to continuing to refine and expand our 2D fighting game in the future.

## VI. ACKNOWLEDGEMENT

A mini project is something that could not have been materialized without cooperation of many people. This project would remain incomplete without expressing our sincere gratitude to those who have provided immense support and encouragement. It is an honor and a privilege to have the esteemed Prof.SamitaPatil as our project guide. We are grateful to her for continuously inspiring and motivating us. A sincere thanks to our project mentor for helping us throughout the project. We would also like to give our sincere thanks to Prof.Dr.UttaraGogate, Head of Department, for their kind support. Last but not the least we would also like to thank The Project Coordinator Prof.ReenaDeshmukh and all the staffs of Shivajirao S. Jondhale college of Engineering (Computer Engineering Department) for their valuable guidance with their interest and valuable suggestions brightened us.

## REFERENCES

- [1]. "The Algorithmic Beauty of Plants" (1990) by PrzemyslawPrusinkiewicz and AristidLindenmayer.
- [2]. "Real-Time Collision Detection" (2005) by Christer Ericson.
- [3]. "Improved Collision detection and Response" (2005) by Kasper Fauerby.
- [4]. "Procedural Content Generation in Games: A Textbook and an Overview of Current Research" (2016) by Noor Shaker, Julian Togelius, and Mark J. Nelson.



- [5]. "A Survey of Real-Time Strategy Game AI Research and Competition in StarCraft" (2013) by Santiago Ontañón, Gabriel Synnaeve, Alberto Uriarte, Florian Richoux, David Churchill, and Mike Preuss.
- [6]. "Techniques for Achieving Realistic Animations and Interactions in Virtual Reality" (1997) by J. P. Lewis, Matt Cordner, and Nickson Fong.
- [7]. "Interactive Dynamic Video" (2005) by Abe Davis, Justin G. Chen, and Fredo Durand.
- [8]. "Game Feel: A Game Designer's Guide to Virtual Sensation" (2008) by Steve Swink.
- [9]. "A Procedural Object Placement Generator on 2D Maps" (2019) by D. Nguyen and E. Wong.
- [10]. "The Influence of Music on Player Performance in Action Games" (2011) by L. Smith and J. White.
- [11]. "Efficient 2D Lighting Effects for Mobile Games" (2016) by H. Zhao and S. Kumar.
- [12]. "Enhancing 2D Game Worlds with Shaders" (2018) by F. Johnson.

