Title:

SPACE WARZ GAME

SEMESTER II

by

Student 1: Karthik Maheshwaram

Student 2: Aayush Racha

Student 3: Ashal George

UNDER THE GUIDANCE

OF

PROF. Omprakash Yadav

(Assistant Professor)

(AY :2024-25)

CONTENTS

PROBLEM STATEMENT

Many classic 2D space shooter games lack modern gameplay features such as smooth animations, adaptive difficulty, and engaging user interfaces. They often become repetitive and don't challenge players dynamically as the game progresses.

This project aims to create **"Space Warz"**, a modern take on the space shooter genre using Python and Pygame. The goal is to design a game that combines intuitive controls, responsive mechanics, dynamic difficulty, and appealing visuals. The game will also include essential features such as sound effects, music, customizable settings, and high score tracking to deliver a more complete and enjoyable player experience.

OBJECTIVES

- Design a visually appealing 2D space shooter game.
- Implement smooth player movement, shooting, and animations.
- Integrate difficulty scaling based on player score.
- Add dynamic sound effects and background music.
- Include user settings for volume and difficulty.
- Maintain and display a high score list.

INTRODUCTION

"**Space Warz"** is a 2D arcade-style space shooter game developed using Python and Pygame. Players control a spaceship to shoot down meteors while avoiding collisions. The game features smooth animation, immersive effects, and progressive difficulty. It includes menu navigation, sound settings, difficulty selection, and real-time high score tracking.

METHODOLOGY

- **Design**: Game mechanics, player controls, scoring, and visuals were drafted.

- **Development**: Implemented core components using Pygame, including:
  - Player movement and shooting
  - Meteor spawning and collisions
  - Explosions and boost animations
  - GUI components (buttons, sliders, menus)

- **Testing**: Played repeatedly to fine-tune difficulty, adjust speed, and fix bugs.

- **Polishing**: Added sounds, particle effects, and high score saving.

ALGORITHM AND PROCESS DESIGN

- **Game Loop:** Runs at 60 FPS; handles events, updates objects, and renders the screen.

- **Collision Detection:** Uses masks for accurate collisions between player/lasers and meteors.

- **Difficulty Scaling:**
  - Score-based changes (e.g., higher spawn rates and meteor types as score increases).

- **Boost System:** Visual particle system to indicate movement and thrust.

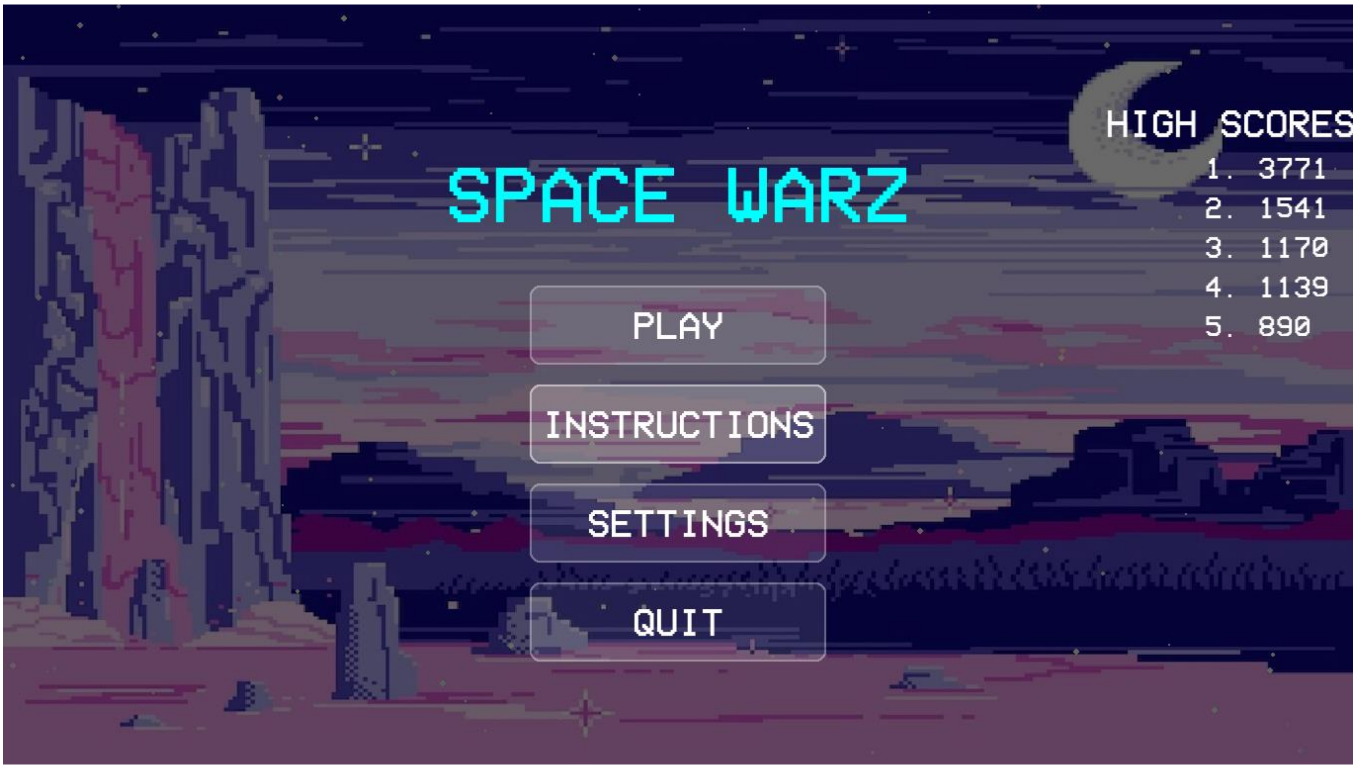
- **UI Handling:** Menu interactions and transitions between game states.

DETAILS OF HARDWARE AND SOFTWARE

- **Hardware Requirements:**
  - Basic system with keyboard and mouse.
  - Display capable of at least 1280x720 resolution.

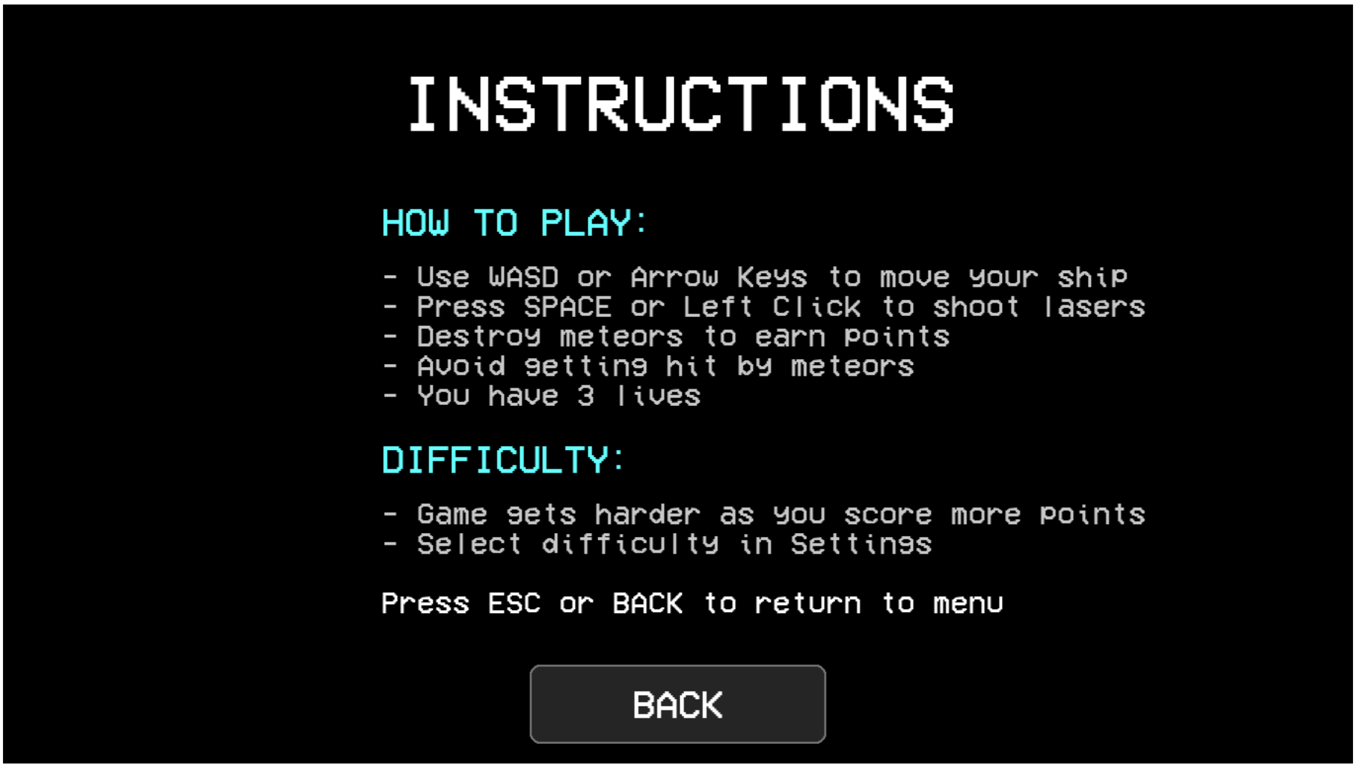
- **Software Requirements:**
  - Python 3.8 or higher
  - Pygame library
  - OS: Windows/Linux/macOS

SCREENSHOTS OF IMPLEMENTATIONS



HIGH SCORES

1. 3771
2. 1541
3. 1170
4. 1139
5. 890



INSTRUCTIONS

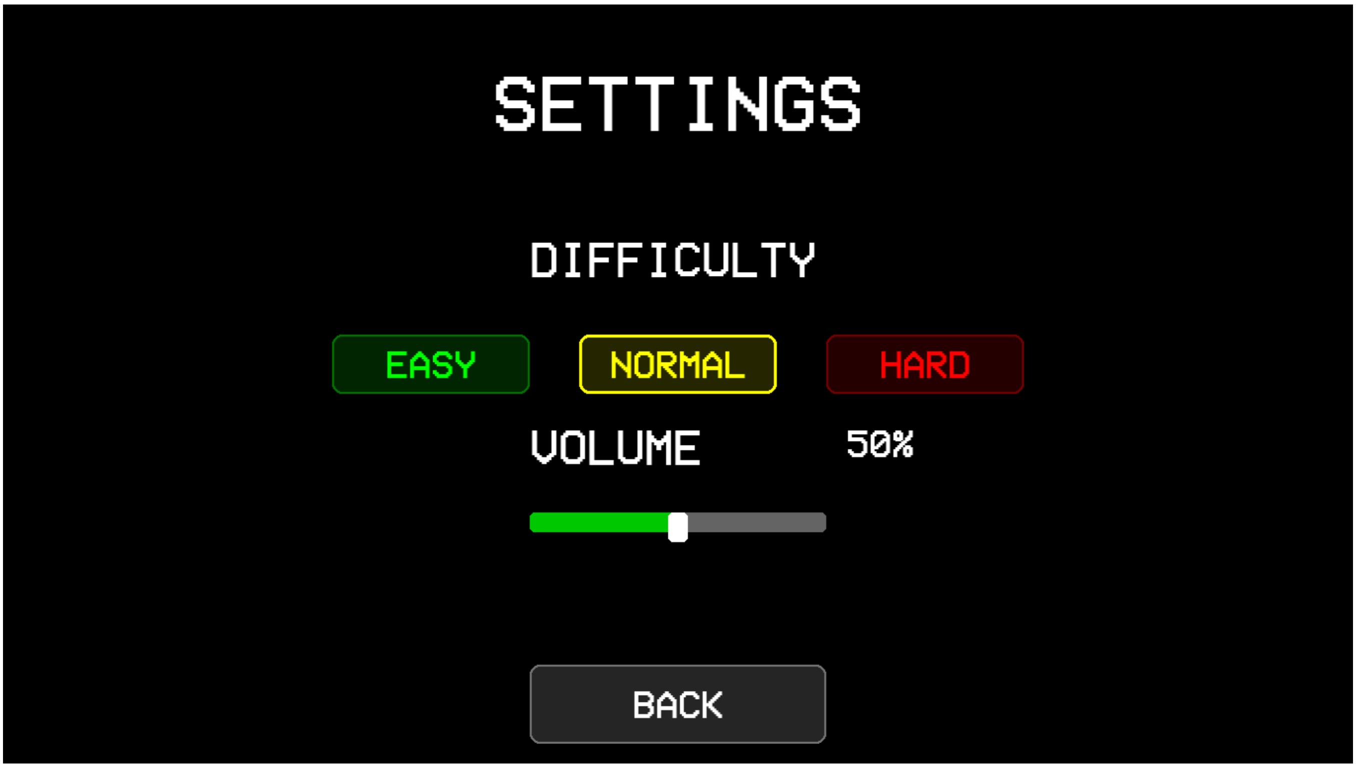
HOW TO PLAY:

- Use WASD or Arrow Keys to move your ship
- Press SPACE or Left Click to shoot lasers
- Destroy meteors to earn points
- Avoid getting hit by meteors
- You have 3 lives

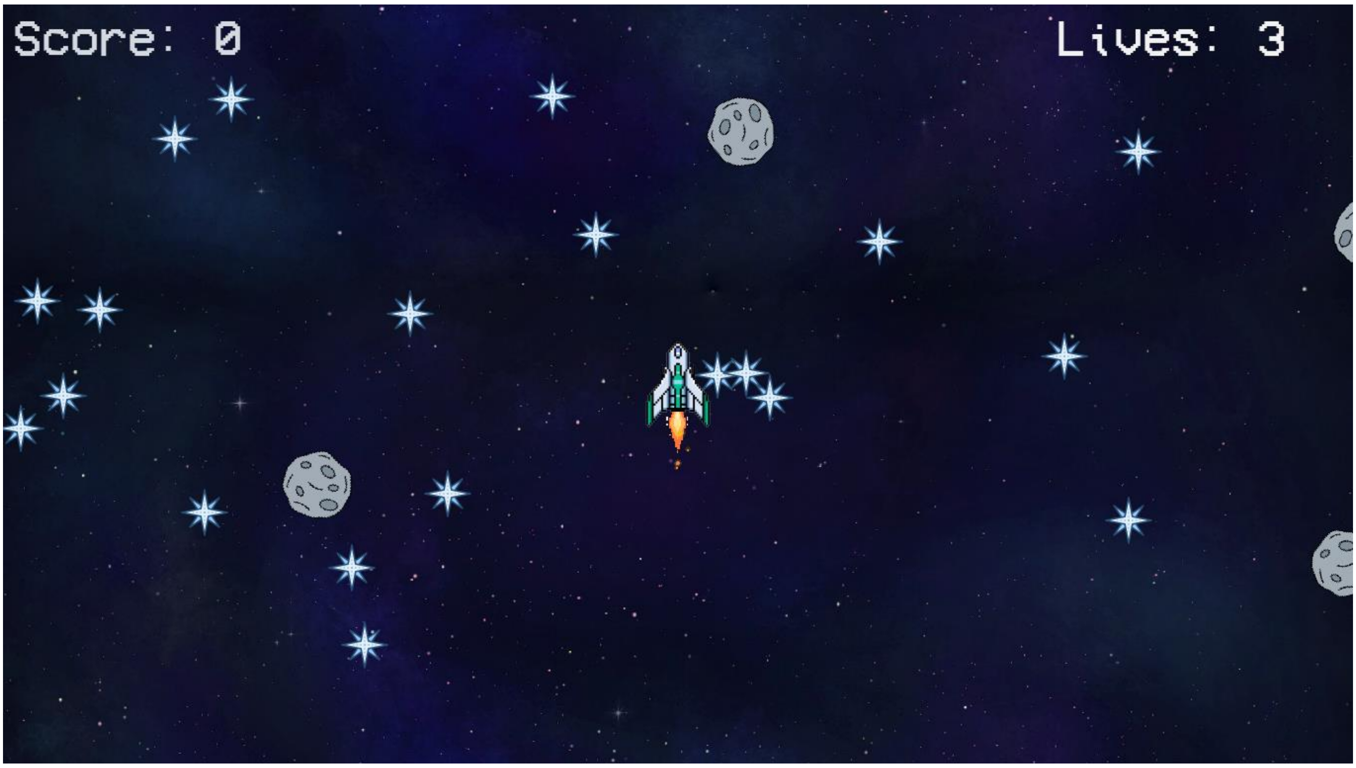
DIFFICULTY:

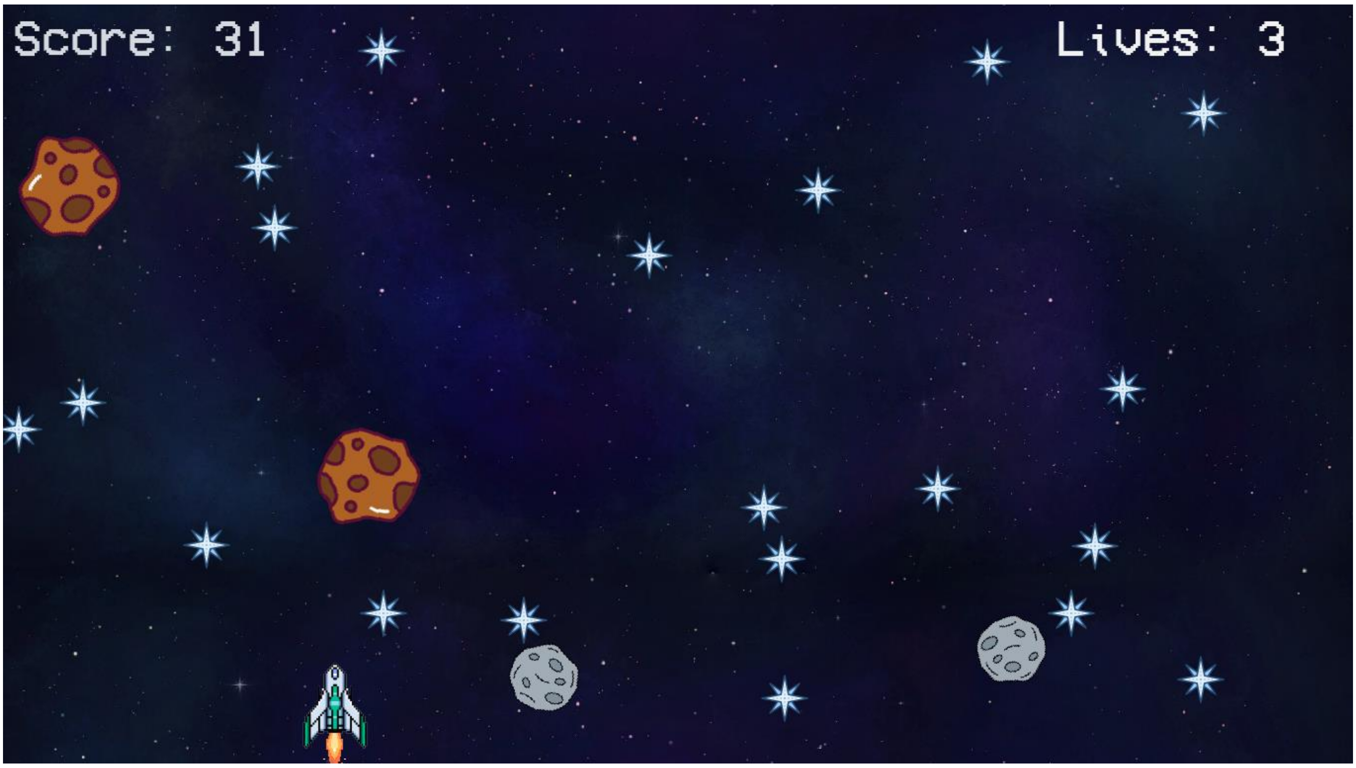
- Game gets harder as you score more points
- Select difficulty in Settings

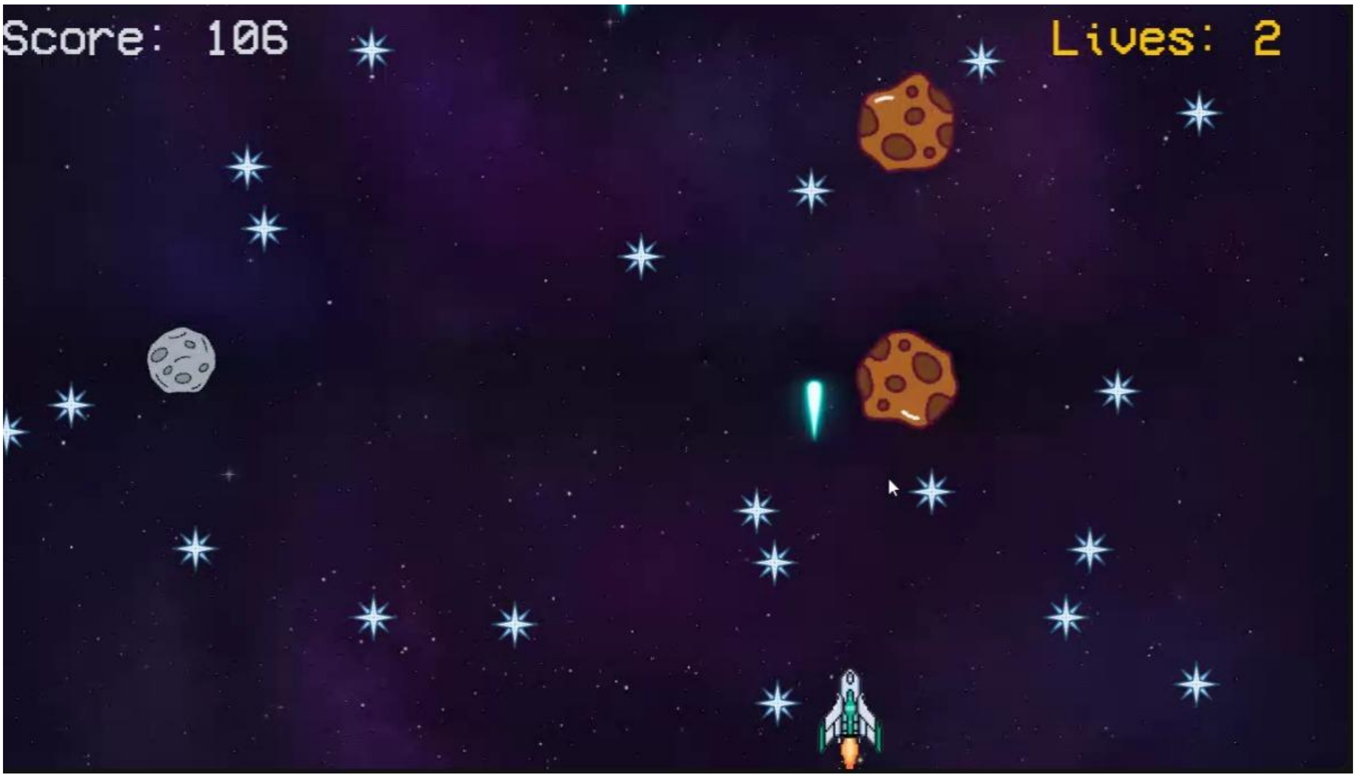
Press ESC or BACK to return to menu

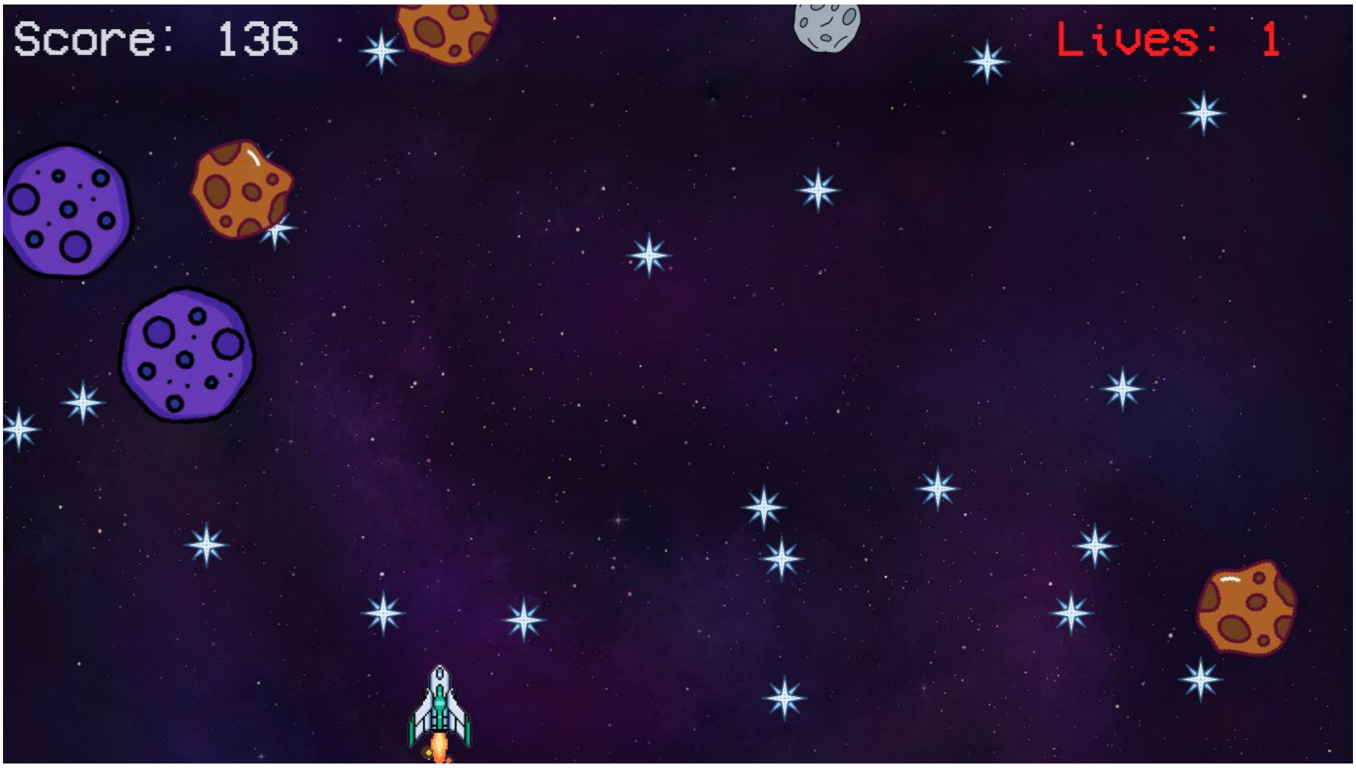


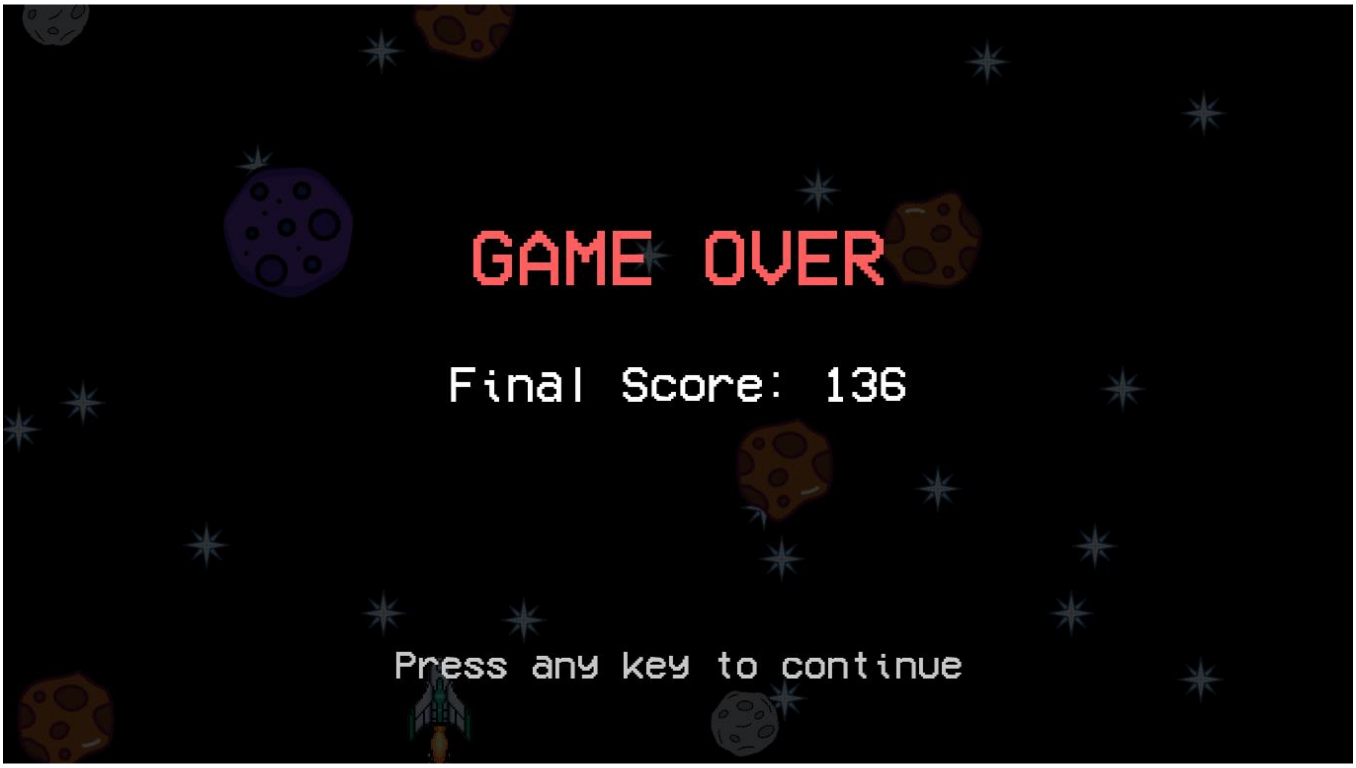
VOLUME 50%


Score: 0
Lives: 3


Score: 31
Lives: 3


Score: 106
Lives: 2


Score: 136
Lives: 1


GAME OVER

Final Score: 136

Press any key to continue

CONCLUSION AND FUTURE WORK

Conclusion:
"**Space Warz**" successfully demonstrates a fully functional arcade game with dynamic visuals, engaging mechanics, and customizable settings. It serves as a showcase of what can be achieved with Pygame and Python.

Link to download the game:- Click here

Future Work:

- Add more enemy types and bosses.
- Include power-ups and weapon upgrades.
- Implement multiplayer mode.
- Add achievements and unlockables.
- Deploy as a web-based game using tools like Pyodide or WebAssembly.
- Make user-friendly for mobile devices
- Build apk file and deploy to playstore