ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 4| April 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

MY SPACE GAME

Ms. Muskan Praveen[1] , Gaurav Pal[2], Aanchal[3] , Yash Tayal[4]
[1]*Assistant Professor, Department of Computer Science and Engineering, Shri Ram Group of Colleges, Dr. A.P.J. Abdul Kalam Technical University*
[2]*Student, Department of Computer Science and Engineering, Shri Ram Group of Colleges, Dr. A.P.J. Abdul Kalam Technical University*
[3]*Student, Department of Computer Science and Engineering, Shri Ram Group of Colleges, Dr. A.P.J. Abdul Kalam Technical University*
[4]*Student, Department of Computer Science and Engineering, Shri Ram Group of Colleges, Dr. A.P.J. Abdul Kalam Technical University*

ABSTRACT
Introduction This project involves designing and developing a 2-dimensional game. The aim of the game involves controlling the spaceship, maneuvering in different directions and attacking other players, which would help in earning points. Multiple enemies will be randomly positioned within the game, and the collision detection will be used to determine whether a bullet attacks an enemy. Moreover, sound effects and background music are incorporated into this game. The primary aim of the project includes learning game development basics such as events handling, real-time computing, and object interactions within the game. This project has been done in collaboration with team members. Different aspects of the game have been designed and developed by different people.
KEYWORDS : *Python, Pygame, 2D Game, Space Shooter, Game Development, Collision Detection, Event Handling, Score System.*

INTRODUCTION

Nowadays, video games have become very popular and are widely used for entertainment as well as learning purposes. Game development is an important field in computer science because it helps in improving programming skills, logical thinking, and problem-solving ability. It also gives practical knowledge of how real-time systems work.

In this project, we have developed a 2D space shooting game using Python and the Pygame library. The main objective of this project is to create an interactive game where the player controls a spaceship and shoots enemies to gain score. The player can move left and right using keyboard controls, and bullets are fired to destroy incoming enemies.

The game includes several important features such as player movement, multiple enemy generation with random positions, bullet firing system, collision detection, score tracking, and sound effects. Collision detection is used to check when a bullet hits an enemy, and based on this, the score is increased. Background music and sound effects are also added to make the game more engaging and realistic.

While developing this project, we faced some challenges such as implementing accurate collision detection, managing multiple enemies at the same time, and controlling the speed of objects. By solving these problems, we improved our understanding of game logic and programming concepts.

This project was developed as a team effort, where each member contributed to different parts such as coding, designing, and testing. Overall, this project helped us to understand the basics of game development and improved our coding, logical thinking, and problem-solving skills.

METHODOLOGY

In this project we took things one step at a time to make a 2D space shooting game using Python and the Pygame library. First we figured out what the game needed to be fun like how the player would move how the enemies would work, how shooting would happen and how scoring would work. Then we made a plan for the game. Split up the work among the team.

We started making the game by setting up the game window, background and sound system with Pygame. Next we made the player part, where the player can move left and right with the keyboard. The players position and speed are controlled by variables.

After that we made the enemy part, where lots of enemies show up in places on the screen. The enemies move side to side. Bounce off the edges of the screen. They also slowly move down which makes the game harder.
Then we made the bullet system, where the player can shoot bullets with the space key. The bullets move up. Disappear when they go off the screen. We used a system to control when the player can shoot, with two states: "ready" and "fire".

We used a math formula to detect when a bullet hits an enemy. When that happens the enemy moves back to the start. The player gets points. We also added effects for shooting and hitting enemies to make the game more fun.

Finally we added a way to show the players score and a game over screen when the enemies get close. The game is always

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 4| April 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

updating everything like the player, enemies, bullets and score to make it feel like its happening in time.

We worked together as a team to finish the project with each person helping with coding finding mistakes and testing the game. The 2D space shooting game was made with Python and the Pygame library. We used the Pygame library to make the game window, background and sound system. The player can. Shoot in the 2D space shooting game. The enemies, in the 2D space shooting game move. Try to hit the player

SYSTEM DESIGN

The space shooting game is built using an approach. This means different parts of the game are designed separately and then combined.
The main parts of the game are:
* The player module
* The enemy module
* The bullet system
* Collision detection
* The score system

The game starts by setting up the Pygame window, background and sound system. Then the player module is activated. This allows the user to control the spaceship using keyboard inputs. The enemy module creates enemies at random positions and controls how they move across the screen.

The bullet system lets the player fire bullets. These bullets move upward. Interact with enemies. When a bullet hits an enemy collision detection is used to check for a hit. If a hit is detected the enemy is. The score is updated.
The game loop keeps running to update all parts of the game in time. This includes:

* The players position
* Enemy movement
* Bullet motion
* Score display

The game ends when enemies reach a position on the screen.
The system is designed so all components work together smoothly. This provides an real-time gaming experience with the space shooting game. The game has a player module, enemy module and bullet system. The player module and enemy module work, with the bullet system.

IMPLEMENTATION

In this project we made a space shooting game using Python and the Pygame library. We started by setting up Pygame and making a game window that's a certain size. We added a background picture. Sound to make the game more fun.

The player part of the game is a picture of a spaceship. You can move the spaceship left and right using the keyboard. We use numbers to keep track of where it is. We can make the spaceship move faster or slower by changing a number.

We made the enemy part of the game using enemy objects that we store in lists. Each enemy starts at a spot on the screen when the game begins. The enemies move across the screen. Bounce off the edges. They also move down a bit at a time to make the game harder.

We added a bullet system so the player can shoot the enemies. When you press the space key a bullet comes out of the spaceship. The bullet moves up the screen. Disappears when it goes off the top. We use a system to decide when you can shoot another bullet.
We use a math formula to see if a bullet hits an enemy. If it does the enemy moves back to the start we play a sound. You get points.

The score part of the game shows your points on the screen using text. We also added a game over part, which happens when the enemies get to a spot, on the screen.

The whole game runs in a loop that keeps updating everything in real time. We use code to see what you are doing like pressing keys or closing the window. The space shooting game and the Pygame library work together to make this happen. The game loop updates the player and the enemies and the bullets and the score all the time.

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 4| April 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188



Fig.1- Gameplay Screen Of The Space Shooting Game

RESULTS

The space shooting game made with Python and the Pygame library is working great. It runs really smoothly on the computer. Gives the player a fun experience. Everything in the game is working like it should.

The player can use the keyboard to control the spaceship and move it around. The spaceship moves easily to the left and right. The game responds well to what the player does so it is easy to play. The player can also shoot bullets at the enemies. It works really well.

When the game starts many enemies appear on the screen at places. They move around smoothly and consistently. When they hit the edge of the screen they change direction. Start moving down. This makes the game harder for the player.

The game can detect when a bullet hits an enemy. When this happens the enemy moves to a random place a sound is played and the player gets points. This shows that the game is working correctly when things interact with each other.

The score is also working well. Is shown on the screen all the time. It updates away when an enemy is destroyed, which keeps the player interested.

The game has effects like music, bullet sounds and explosion sounds. These things make the game more fun and realistic.

The game ends when enemies get too close to the player. When this happens, the game. Shows a message that says the game is over. This gives the game an ending.

The space shooting game is working well and does everything it is supposed to do. The project shows that important things like handling events objects interacting, detecting collisions and updating in time are all working. The space shooting game and the Pygame library are a combination, for making a fun game. The game is an example of what can be made with Python and the Pygame library.

CONCLUSION

We created a two-dimensional shooting space game. The game was created using Python with the Pygame libraries, and it enabled us to create a structure in which we could learn the fundamentals of developing games and apply those fundamental concepts into a real-world application. All of the basic game functionalities were implemented successfully, including the ability to move the player, spawn enemies, shoot bullets, detect if a bullet hit an enemy (collision), maintain a score, and use sound effects. Through this project, we developed an understanding of important programming concepts, including events, loops, and conditional statements used in programming. Additionally, we developed an understanding on how the components of a game interact with one another to develop a gaming experience that is both smooth and enjoyable. During the development of the game, we overcame many challenges, including how to control multiple enemies, the accuracy of the collision detection, and the speed of the objects. Each time we solved a problem, we became better at logical thinking and debugging, as well as developing our ability to work as a team. The project was a collaborative effort; each person in the team worked on one or more of the following: coding, graphic design, or testing the game. As a

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 4| April 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

result of working together as a team, we improved our ability to communicate and coordinate. In conclusion, this project has been a valuable experience in terms of gaining technical knowledge and having the opportunity to gain actual experience in the development of code and gaming through Python. It also helped develop problem-solving and technical skill development, which are all transferable skills.

Future Scope

The project can benefit from future improvement options that involve creating new features and enhancements to the game. Adding multiple levels that are progressively more challenging can enhance gameplay and make it more enjoyable. Improving the game's visual appearance will help to make it more visually appealing and realistic for players.

Adding features like player health/lives will enhance overall gameplay experience. Looking at more extensive development opportunities (e.g., sounds/music) will create a better user experience.

Expanding the project through the introduction of multiplayer will allow for more player interaction/connectedness to occur. Developing on more sophisticated game engines will allow for optimal design/performance capabilities.

In summary, as you can see, multiple opportunities to enhance this project provide great potential to develop an even more complex, engaging and user friendly gaming experience.

Future Scope

1. *Python Software Foundation, Python Documentation. Available at: https://docs.python.org*
2. *Pygame Community, Pygame Documentation. Available at: https://www.pygame.org/docs/*
3. *YouTube Tutorials on Python Game Development*

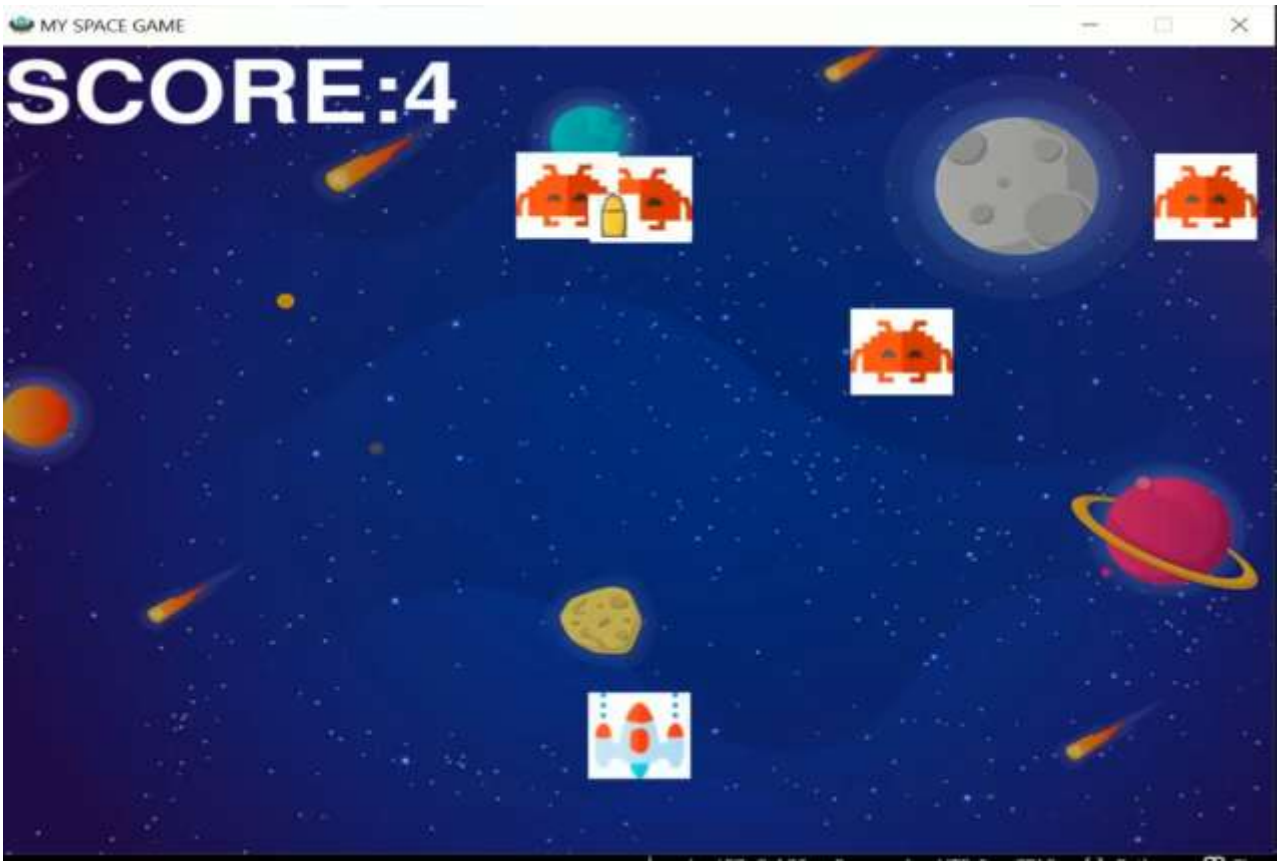


Fig.2- Collision detection between player bullet and Enemy Spaces

ISSN (Online): 2455-3662
EPRA International Journal of Multidisciplinary Research (IJMR) - Peer Reviewed Journal
Volume: 12| Issue: 4| April 2026|| Journal DOI: 10.36713/epra2013 || SJIF Impact Factor 2026: 8.753 || ISI Value: 1.188

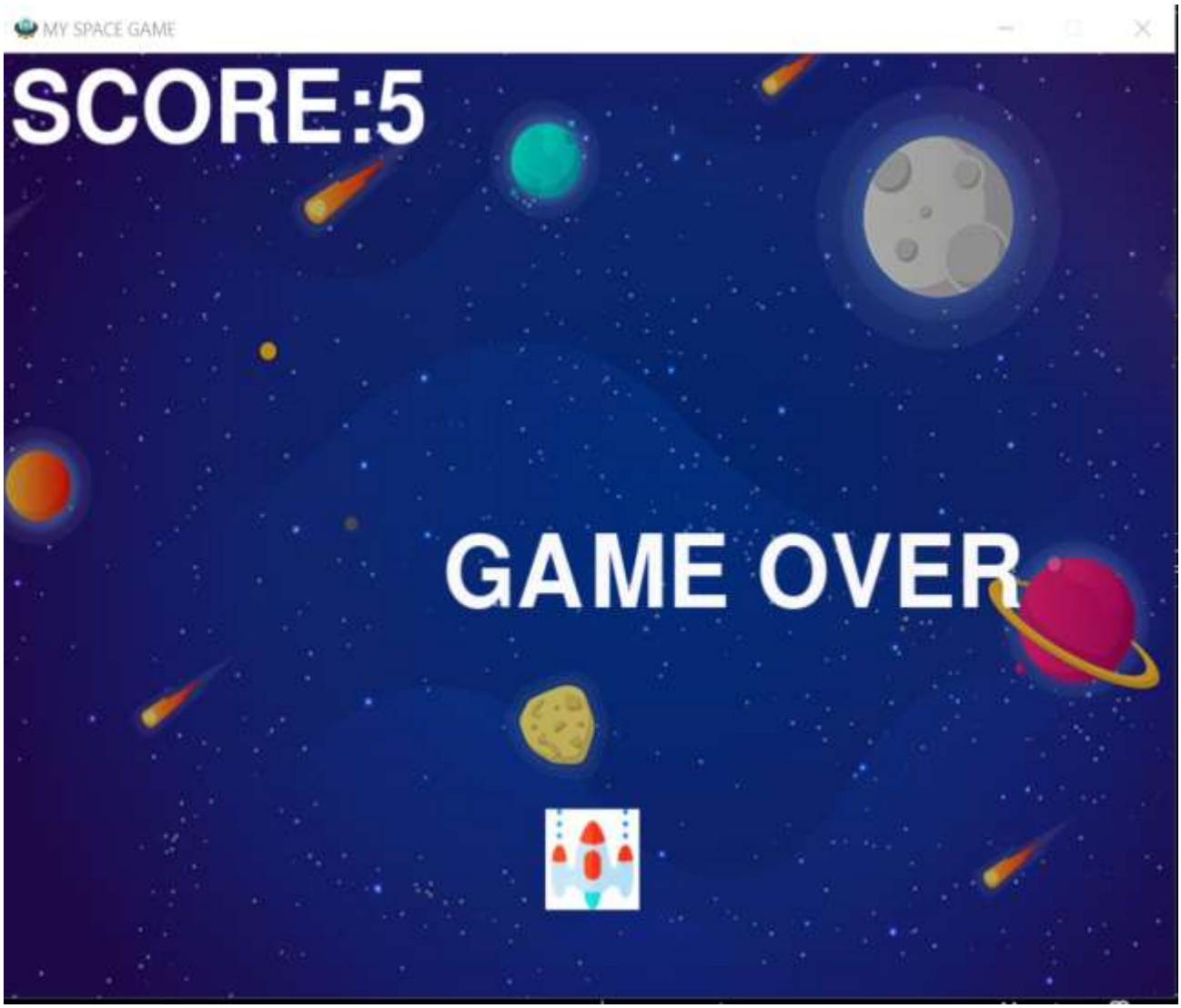


Fig.3-Game Over Interface Showing end of Gameplay when enemy reaches The player