

SPACE INVADER

**Project report in partial fulfillment of the requirement for the award of the degree of
Bachelor of Technology
In
Computer Science**

Submitted By

Arya Nath	University Roll No. 12020009029009
Suprotim Karmakar	University Roll No. 12020009022174
Sayak Sarkar	University Roll No. 12020009022148
Bratin Das	University Roll No. 12020009022127
Sagor Mondal	University Roll No. 12020009001241
Abhijit Maity	University Roll No. 12020009023012
Mohan Kumar Jana	University Roll No. 12020009023009
Arigna Adhikari	University Roll No. 12020009022033

Under the guidance of

Prof. Dr. Maumita Chakraborty

&

Prof. Prasenjit Kumar Das

Department of **Computer Science & Technology**



UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA

University Area, Plot No. III – B/5, New Town, Action Area – III, Kolkata – 700160.

CERTIFICATE

This is to certify that the project titled **Space Invader** submitted by **Arya Nath**(University Roll No. 12020009029009),**Suprotim Karmakar**(University Roll No.12020009022174),**Sayak Sarkar**(University RollNo.12020009022148), **Bratin Das**(University Roll No. 12020009022127), **Sagor Mondal**(University Roll No. 12020009001241), **Abhijit Maity**(University Roll No. 12020009023012), **Mohan Kumar Jana** (University RollNo.- 12020009023009) and **Arigna Adhikari** (University Rollno. - 12020009022033) students of UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA, in partial fulfillment of requirement for the degree of Bachelor of Computer Science & Technology, is a bonafide work carried out by them under the supervision and guidance of **Prof. Dr. Maumita Chakraborty** & **Prof. Prasenjit Kumar Das** during 4thth Semester of academic session of 2020-2024. The content of this report has not been submitted to any other university or institute. I am glad to inform that the work is entirely original and its performance is found to be quite satisfactory.

Prof. Dr. Maumita Chakraborty
Project Guide
HOD,
Department of Computer Science and Technology
UEM, Kolkata

Prof. Prasenjit Kumar Das
Project Guide & Assistant Professor
Department of Computer Science & Technology
UEM, Kolkata

Prof. Dr. Maumita Chakraborty
HOD,
Department of CST & CSIT

Prof.Dr. Sukalyan Goswami
HOD, Department of CSE
UEM, Kolkata

ACKNOWLEDGEMENT

We would like to take this opportunity to thank everyone whose cooperation and encouragement throughout the ongoing course of this project remains invaluable to us.

We are sincerely grateful to our guide **Prof. Dr. Maumita Chakraborty** and **Prof. Prasenjit Kumar Das** of the Department of Computer Science & Technology, UEM, Kolkata, for his wisdom, guidance and inspiration that helped us to go through with this project and take it to where it stands now.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

Arya Nath

Suprotim Karmakar

Sayak Sarkar

Bratin Das

Sagor Mondal

Abhijit Maity

Mohan Kumar Jana

Arigna Adhikari

TABLE OF CONTENTS

ABSTRACT.....<<1>>

INTRODUCTION.....<<2>>

PROBLEM DEFINITION<<3>>

SOLUTION<<4>>

BASICS CONCEPT & TOOLS<<5-9>>

i) Python<<5>>

ii) Pygame.....<<6-8>>

iii) Elaboration of the code.....<<9>>

iv) Loop.....<<10>>

**SCREENSHOT OF OUR
PROJECT.....<<11>>**

CONCLUSION<<12>>

BIBLIOGRAPHY<<13>>

ABSTRACT

Our project “Space Invader” is a attempt to bring the joy and memories of the beloved game that has ruled over the arcade genre and also reigned over the hearts of children’s’ of multiple generations.

The ‘Space Invaders’ project is developed to overcome the unnecessary boredom that kids often face. An easy to develop game that requires no extra performance load from computer and really easy understandable game mechanics. A very quick mind refreshing game that doesn’t become an addiction to children. Users’ reflexes as well as presence of mind are tested while playing this game. No COMPLEX controls. The game gets over when the waves of aliens are destroyed or when the user gets defeated. Children of same age groups can play this game and compete amongst themselves to beat the high score.

INTRODUCTION

Space Invaders is a 1978 shoot 'em up arcade game developed by Tomohiro Nishikado. It was manufactured and sold by Taito in Japan, and licensed to the Midway division of Bally for overseas distribution.

The goal is to eliminate all of the aliens by shooting them. While the player has three lives, the game ends immediately if the invaders reach the bottom of the screen. The aliens attempt to destroy the player's cannon by firing projectiles. The laser cannon is partially protected by stationary defense bunkers which are gradually destroyed from the top by the aliens and, if the player fires when beneath one, the bottom.

As one of the earliest shooting games, *Space Invaders* set precedents and helped pave the way for future titles and for the shooting genre. *Space Invaders* popularized a more interactive style of gameplay, with the enemies responding to the player-controlled cannon's movement, and was the first video game to popularize the concept of achieving a high score, being the first to save the player's score. While earlier shooting games allowed the player to shoot at targets, *Space Invaders* was the first in which multiple enemies could fire back at the player, and in contrast to earlier arcade games which often had a timer, *Space Invaders* introduced the "concept of going round after round." It was also the first game where players were given multiple lives

PROBLEM DEFINITION

The aim is to destroy the rows of aliens as they move horizontally across the screen, moving faster and faster as you pass levels. Once you destroy one wave of aliens get ready to face another faster and more difficult wave of aliens. If you allow any space invaders to reach the bottom of the screen or enemies bullets consume your lives, the alien invasion has been successful and the game end.



SOLUTION

In this game, our player has to defeat waves of enemy before they reach the end of the line, to make the gameplay more effective it has multiple lives to fight with. It also first brought the idea of recording one's score, which made the game more competitive and interactive. One can now compete with their friends and known ones on who has the highest score. It also didn't have a timer like the arcade games of that time, it was based on the concept of round after round and each round or level was tougher than the previous one.



BASIC CONCEPT AND TOOLS

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.



Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Characteristics-

- 1) It supports functional and structured programming methods as well as OOP.
- 2) It can be used as a scripting language or can be compiled to byte-code for building large applications.
- 3) It provides very high-level dynamic data types and supports dynamic type checking.
- 4) It supports automatic garbage collection.
- 5) It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

PYGAME-

Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language.



Pygame is a Python wrapper for the SDL library, which stands for **Simple Direct Media Layer**. SDL provides cross-platform access to your system's underlying multimedia hardware components, such as sound, video, mouse, keyboard, and joystick. Pygame started life as a replacement for the stalled PySDL project. The cross-platform nature of both SDL and pygame means you can write games and rich multimedia Python programs for every platform that supports them.

Pygame is obviously strongly dependent on SDL and Python. It also links to and embeds several other smaller libraries. The font module relies on SDL_ttf, which is dependent on freetype. The mixer (and mixer.music) modules depend on SDL_mixer. The image module depends on SDL_image, which also can use libjpeg and libpng. The transform module has an embedded version of SDL_rotozoom for its own rotozoom function. The surfarray module requires the Python NumPy package for its multidimensional numeric arrays.

Initialization and Modules-

The pygame library is composed of a number of Python constructs, which include several different **modules**. These modules provide abstract access to specific hardware on your system, as well as uniform methods to work with that hardware. For example, `display` allows uniform access to your video display, while `joystick` allows abstract control of your joystick.

After importing the pygame library in the example above, the first thing you did was **initialize PyGame** using `pygame.init()`. This function **calls the separate `init()` functions** of all the included pygame modules. Since these modules are abstractions for specific hardware, this initialization step is required so that you can work with the same code on Linux, Windows, and Mac.

Displays and Surfaces-

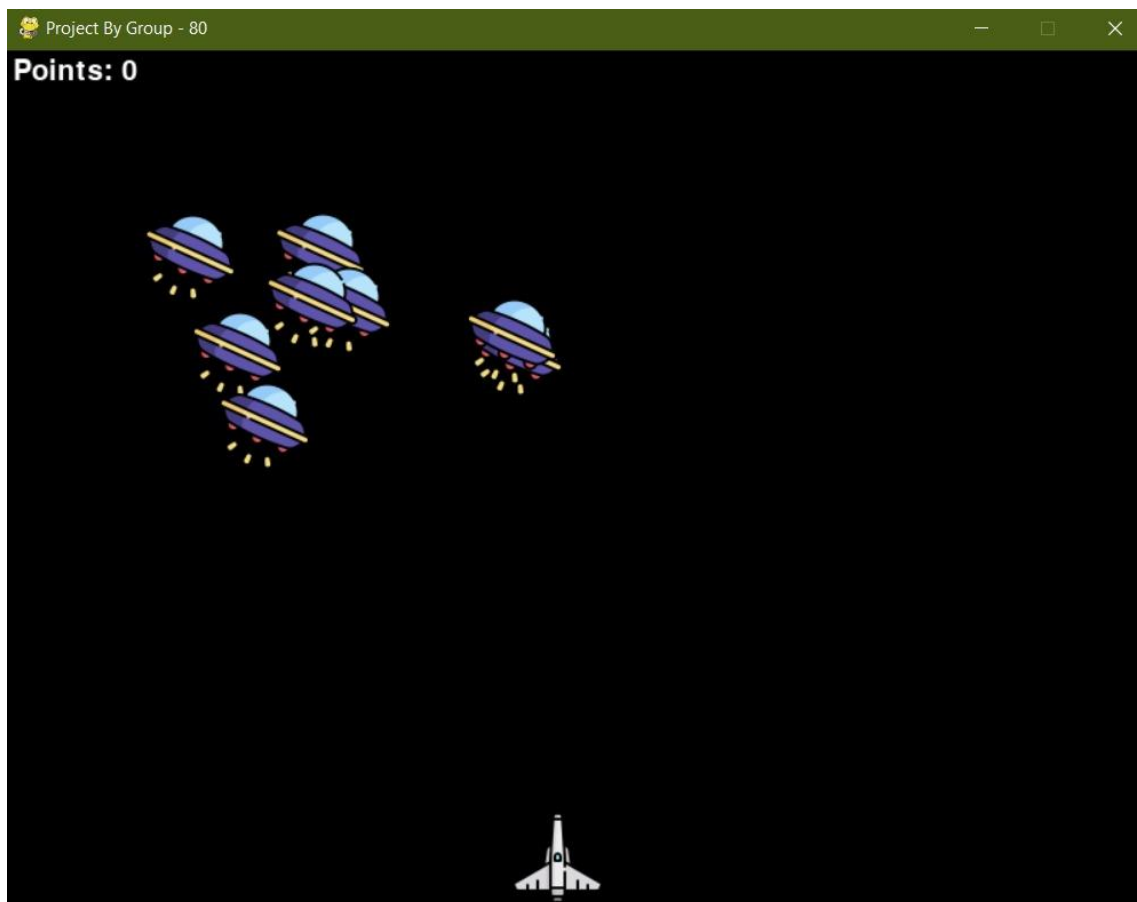
In addition to the modules, pygame also includes several Python **classes**, which encapsulate non-hardware dependent concepts. One of these is the **Surface** which, at its most basic, defines a rectangular area on which you can draw. Surface objects are used in many contexts in pygame. Later you'll see how to load an image into a Surface and display it on the screen.

In pygame, everything is viewed on a single user-created **display**, which can be a window or a full screen. The display is created using `set_mode()`, which returns a Surface representing the visible part of the window. It is this Surface that you pass into drawing functions like `pygame.draw.circle()`, and the contents of that Surface are pushed to the display when you call `pygame.display.flip()`.

Images and Rects-

Your basic pygame program drew a shape directly onto the display's Surface, but you can also work with images on the disk. The **image** module allows you to load and save images in a variety of popular formats. Images are loaded into Surface objects, which can then be manipulated and displayed in numerous ways.

As mentioned above, Surface objects are represented by rectangles, as are many other objects in pygame, such as images and windows. Rectangles are so heavily used that there is a special **Rect** class just to handle them. You'll be using Rect objects and images in your game to draw players and enemies, and to manage collisions between them.



Elaboration of the code:

Implementation of the code: -

Import the required module

Initialize the pygame

Create three functions:

IsCollision(): which tells us whether the collision has occurred or not?

game_over(): which returns true or false depending on what the code has decided if the game has ended.

Show_score(x,y): this shows the score on the screen

Create an infinite loop to execute the code continuously.

Elaboration of the functions being used in this code:

isCollision(): The value returned by this function is being stored inside the “collision” variable. The value returned by the function is either True or False based on the criteria set for collision inside the function.

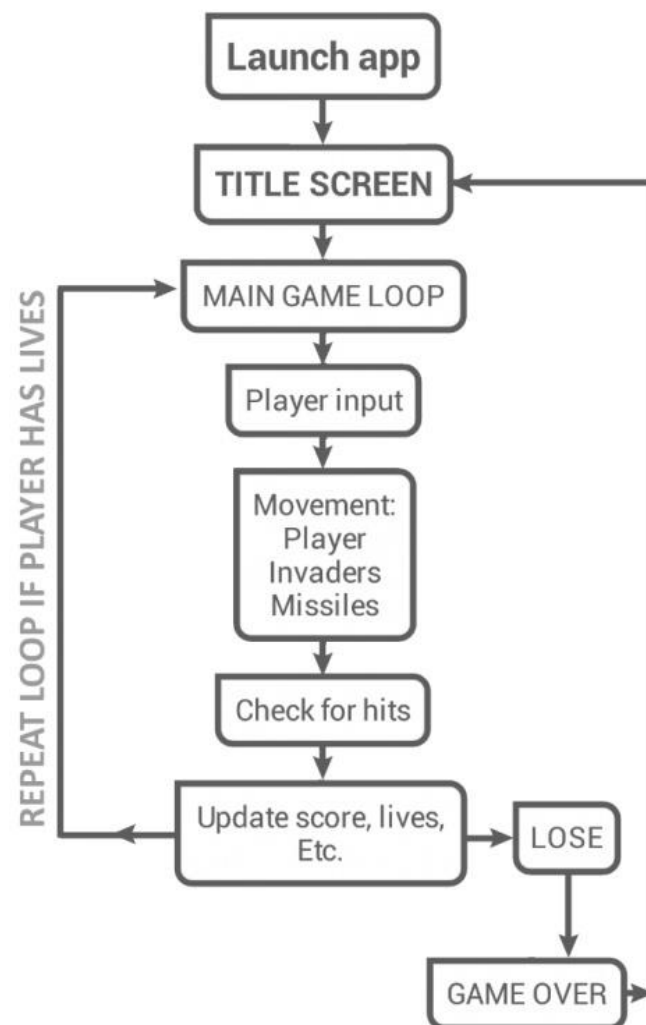
game_over(): Returns True or False based on what the code has decided after the game has ended.

show_score(x,y): Shows the score of the player on the screen.

player(x,y): This function defines a player, i.e , a user. This allows the user to get access to the control of the ship in the game.

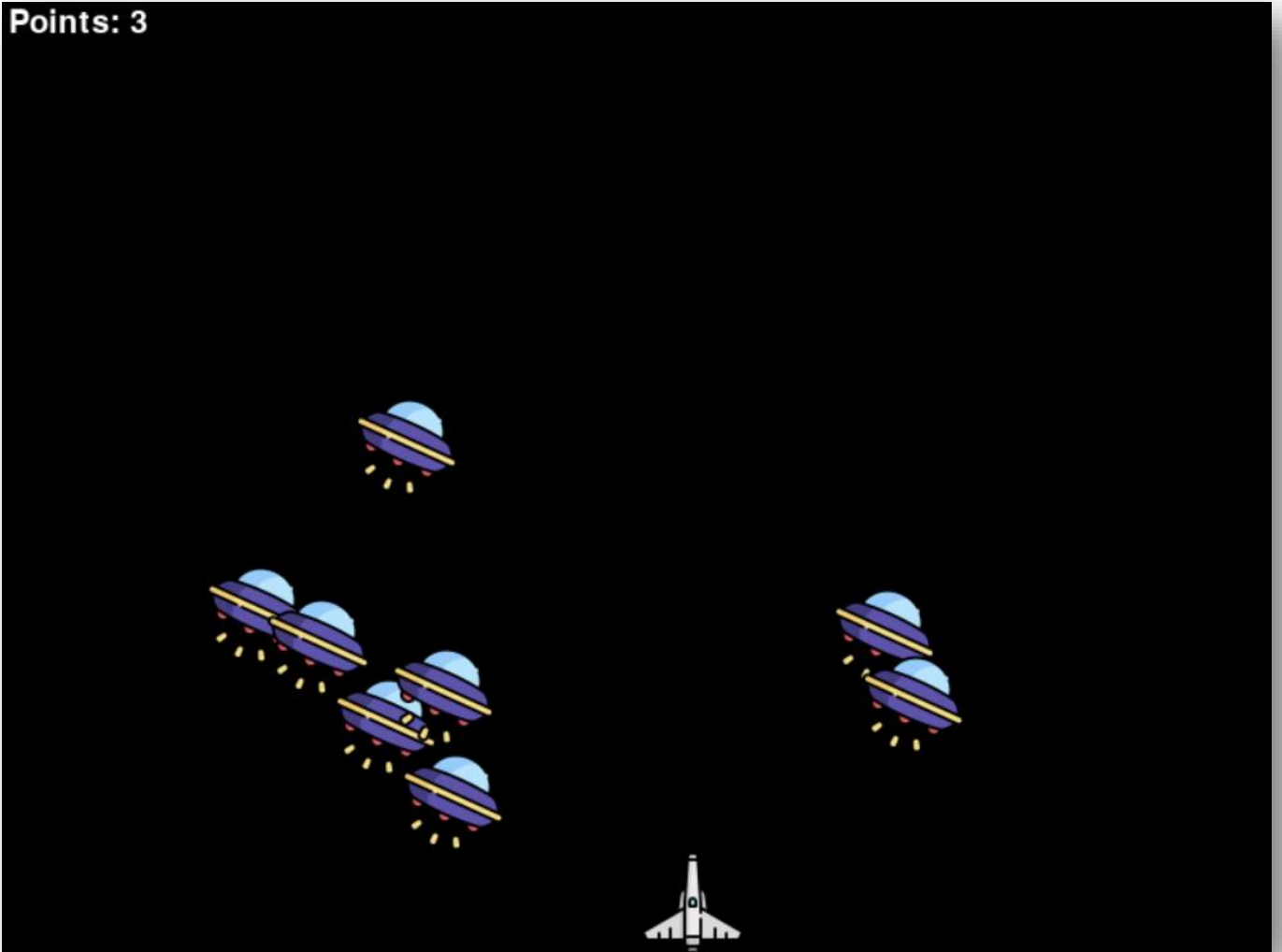
bullet(x,y); This function defines the bullets that comes out from the ship. This allows the user to shoot lasers towards the aliens, hence getting to play the game as it is meant to be played.

SPACE INVADER GAME'S LOOP-



SCREENSHOT

Points: 3



CONCLUSION

To conclude this, this is our humble and simple approach to rekindle our childhood and nostalgia, a game that has brought happiness to countless people. We have plans of making the rounds and gameplay more interactive and unique. This can be a great time-pass and also a way to relive one's childhood.

BIBLIOGRAPHY

[Python \(programming language\) - Wikipedia](#)

[Python Tutorial \(tutorialspoint.com\)](#)

[PyGame: A Primer on Game Programming in Python – Real Python](#)

[GitHub - pygame/pygame: pygame \(the library\) is a Free and Open Source python programming language library for making multimedia applications like games built on top of the excellent SDL library. C, Python, Native, OpenGL.](#)