1. **INTRODUCTION**

Pygame is a cross-platform set of Python modules which is used to create video games. It consists of computer graphics and sound libraries designed to be used with the Python programming language. Pygame was officially written by Pete Shinners to replace PySDL. Pygame is suitable to create client-side applications that can be potentially wrapped in a standalone executable.

The concept of shooting games existed before video games. Mechanical target shooting games first appeared in England's amusement arcades around the turn of the 20th century, before appearing in America by the 1920s. Shooting gallery games eventually evolved into more sophisticated target shooting electro-mechanical games (EM games) such as Sega's influential Periscope (1965). Shooting video games have roots in EM shooting games. While earlier shooting games allowed the player to shoot at targets, Space Shooter game was the first where multiple enemies fired back at the player. It also introduced the idea of giving the player multiple lives and popularized the concept of achieving a high score.

The space shooter is a very attractive game for any age peoples. The spaceship moves around the galaxy and protect galaxy from various asteroids. The game begins by pressing enter key. Asteroids starts attacking the spaceship. The spaceship shoots and destroy the asteroids. The score is being increased automatically. A hit from an asteroid causes the player’s score to decrease. While smaller asteroids deals lesser damage, larger asteroids will deal a heavier blow to the spaceship. The game ends when the player’s spaceship was destroyed by the asteroids. In the existing system the keys in the keyboard are not used, so handling the game is so hard.

* 1. **PROBLEM DEFINITION**
* Easy to handle.
* Increases mind ability

**2. SYSTEM ANALYSIS**

System analysis is a general term that can refer to stored process for identifying and solving problem. Analysis implies the process of breaking something down into parts so that the whole may be understood. The definition of system analysis not only the process of analysis but also that of synthesis, which is the process of putting parts together to form a new whole.

**2.1 EXISTING SYSTEM**

Previously, there was no facility to see the users details presented in Space Shooter Game. And most of them sign up options, but to make easy gaming options, there was no sign-up option for the users. There was no provision of adding extra gaming section for feature reference by the admin to make their website highly demanded in coming future.

**2.1.1 DRAWBACKS**

* Sign in to play
* Slow work processing
* Not user-friendly environment
* Sound effect are not clear

* 1. **PROPOSED SYSTEM**

In this new gaming environment, no sign-up options and fees required to play the games. We have tried to code smart decision-making algorithms to make system more intelligent and can learn the move from the human beings. At same time, while playing most of us faces difficulties like consuming more time and other features. To overcome from all these problems a new system has been proposed.

**2.2.1** **ADVANTAGES OF PROPOSED SYSTEM**

* Easy to play
* No sign in options
* Improves concentration and problem solving skills
* Gives relaxation
* Sound effects are good and clear

**2.3 PROJECT DESCRIPTION**

The “**Space Shooter Game**” is simple game project which allow players or users to play the space shooter game. The users have the choice to select the game as per their choice and play at free cost with full features. The good or excellent picture quality and its optimized code will help users to get processing work faster and without any delay.

The purpose of this project is to design and build a **Space Shooter Game** that will give user the experience of solving various tricky Space Shooter through gaming.

This was achieved through several steps. In background we have used programming language assembly, instructions like condition, loops and call procedures what we have learnt is used.

**3. SYSTEM REQUIREMENTS**

**3.1 HARDWARE REQUIREMENTS**

Processor : Intel Pentium IV

RAM Capacity : 10GB

Hard Disk : 500GB

Mouse : Optical Mouse

Keyboard : 108 keys

Monitor : Color Monitor

**3.2 SOFTWARE REQUIREMENTS**

Operating system : Windows 10

Language : Python

Software : Visual studio code

**3.3 SOFTWARE SPECIFICATION**

**3.3.1 PYTHON**

**Python is the most versatile language, and it makes its presence almost in every field including Web-development,**[**Machine Learning**](https://www.javatpoint.com/machine-learning)**,**[**Artificial Intelligence**](https://www.javatpoint.com/artificial-intelligence-tutorial)**, GUI Application as well as Game Development. It provides a built-in library called**[**pygame,**](https://www.javatpoint.com/pygame)**which used to develop the game. Once we understand the basic concepts of Python programming, we can use the pygame library to make games with attractive graphics, suitable animation and sound.**

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers to write clear, logical code for small and large scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.

**Python libraries**

**A Python library is a collection of related modules. It contains bundles of code that can be used repeatedly in different programs. It makes Python Programming simpler and convenient for the programmer**. Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary-precision decimals, manipulating regular expressions, and unit testing. Some parts of the standard library are covered by specifications—for example, the Web Server Gateway Interface (WSGI) implementation ws giref follows PEP 333—but most are specified by their code, internal documentation, and test suites. However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations. As of September 2021, the Python Package Index (PyPI), the official repository for third-party Python software contains over 329,000 packages with a wide range of functionality.

# ****Python Features****

**Python provides many useful features which make it popular and valuable from the other programming languages. It supports object-oriented programming, procedural programming approaches and provides dynamic memory allocation.**

**The features are**

### **Easy to Learn and Use**

### **Interpreted Language**

### **Cross-platform Language**

### **Free and Open Source**

### **Dynamic Memory Allocation**

**3.3.2 Pygame**

Python is the most popular programming language or nothing wrong to say that it is the next-generation programming language. In every emerging field in computer science, Python makes its presence actively. Python has vast libraries for various fields such as **Machine Learning** (Numpy, Pandas, Matplotlib), **Artificial intelligence** (Pytorch, TensorFlow), and **Game development** (Pygame,Pyglet).

* **Pygame** is a cross-platform set of Python modules which is used to create video games.
* It consists of computer graphics and sound libraries designed to be used with the Python programming language.
* **Pygame** was officially written by Pete **Shinners** to replace **PySDL**.
* **Pygame** is suitable to create client-side applications that can be potentially wrapped in a standalone executable.

## ****Pygame Surface****

## **The pygame Surface is used to display any image. The Surface has a pre-defined resolution and pixel format. The Surface color is by default black. Its size is defined by passing the size argument.Surfaces can have the number of extra attributes like alpha planes, color keys, source rectangle clipping, etc. The blit routines will attempt to use hardware acceleration when possible; otherwise, they will use highly enhanced software blitting methods.**

## ****Pygame Text and Font****

**Pygame also provides facilities to render the font and text. We can load fonts from the system by using the pygame.font.SysFont() function. Pygame comes with the built-in default font which can be accessed by passing the font name or none. There are many functions to help to work with the font.The font objects are created with pygame.font.Font().The actual font objects do most of the works done with fonts. Font objects are generally used to render the text into new surface objects.**

## ****Pygame Blit****

## **The pygame blit is the process to render the game object onto the surface, and this process is called blitting. When we create the game object, we need to render it. If we don't render the game objects and run the program, then it will give the black window as an output.Blitting is one of the slowest operations in any game so, we need to be careful to not to blit much onto the screen in every frame. The primary function used in blitting is blit( ).**

## ****Pygame Sprite****

**A pygame sprite is a two-dimensional image that is part of the large graphical scene. Usually, a sprite will be some object in the scene. One of the most advantages of working with sprites is the ability to work with them in groups. We can easily move and draw all the sprites with the one command if they are in the group. The Sprite module contains the various simple classes to be used within the games. It is optional to use Sprite classes and different group classes when using pygame.**

**4. SYSTEM DESIGN**

**4.1 Site map**

**Launch app**

**TITLE SCREEN**

**MAIN GAME LOOP**

**Player Input**

**Movement: Player Invaders Missiles**

**Check for hits**

**LOSE**

**Update score, lives, Etc**

**GAME OVER**

**4.2 DATA FLOW DIAGRAM**

A data flow diagram is a graphical representation that depicts information flow and thetransforms that are applied as data move from input to output. A data flow diagram may be used to represent a system or software at any level of abstraction. DFD’s can be partitioned into levels that represent increasing information flow and functional details. Analysis model help us to understand the relationship between different components in the system design. Analysis model shows user more clearly, how a system will function. This is the first technical representation of a system.

The analysis is modeling must achieve three primary objectives:

* To establish a basis for creation of software design
* To describe what the user requires
* To define a set of requirement that can be validated once the software is built Data

Flow Diagram.

A Data Flow Diagram is a graphical technique that depicts information flow and transforms that are applied as data move from input to output. The DFD is used to represent increasing information flow and functional details. A level 0 DFD, also called a fundamental system model or a Context Model, represents the entire software elements as a bubble with input and output indicated by incoming and outgoing arrows respectively. Each of the processes represented at Level 1 are sub functions of overall system depicted in the Context Model. Any processes, which are complex in Level 1, will be further represented into sub functions in the next Level, i.e. in level 2.

Basic DFD symbols:

* Arrow
* Circles
* Open End Box
* Squares

|  |  |  |
| --- | --- | --- |
| NAME | SYMBOLS | DESCRIPTION |
| Arrow |  | Data flow |
| Circle |  | Process |
| Rectangle |  | Source (or) Destination |
| Open end box |  | Data storage |

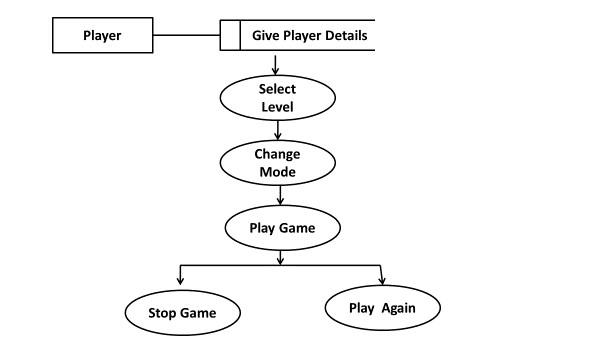
## FLOW DIAGRAM

**Level-0**

Player

Database stores the score

##### Level-2



**5. SYSTEM TESTING**

**5.1 TEST PLAN**

Testing is an important stage in system development life cycle (SDLC). Softwaretesting is a critical element of software quality assurance and represents the ultimate view ofspecification, design and coding. Testing begins at the module level and work towards theintegration of entire computer based system. Nothing is complete without testing, as it is avital success of the system.

There are three ways to test a program:

* For Correctness
* For Implementation efficiency
* For Computational Complexity

Tests for correctness are supposed to verify that a program does exactly what it was supposed to do. This is much more difficult than it may first appear, especially for large programs. Tests for implementation efficiency attempt to find ways to make a correct program run faster and use less storage. It is a code refining process, which re-examines the implementation phase of algorithm development. Tests for computational complexity amount to an exceptional analysis of complexity of an algorithm or an experimental comparison of two or more algorithms, which solve the same problem. The next step involved is the Design and Development phase that practically helps to build the project.

The methods that are applied during the development phase are:

* Software Design
* Code Generation
* Software Testing

The “Linear Sequential Model” or “Classic Life Cycle” or the “Waterfall Model” develops project. This is a sequential approach to software development that begins at the system level and progresses through analysis, design, coding and testing.

**TYPES OF TESTS**

**UNIT TESTING**

**Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.**

**INTEGRATION TESTING**

**Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.**

**SYSTEM TESTING**

**System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.**

**WHITE BOX TESTING**

**White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.**

**BLACK BOX TESTING**

**Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.**

**6. SYSTEM IMPLEMENTATION**

Implementation is the final stage of the project. System Implementation is the major part of the project in which theoretical design is turned into working system. System Implementation is the process of converting a new or a revised system design into an operational one. System implementation includes planning, investigation of scenario, constraints and implementation of the project. Initially, the task is to plan the concept includes execution methods, time taken to complete the task. Project concept is completely based on playing game.

The implementation plan consists of the following sample steps

* Detection and correcting errors
* Making necessary changes
* Installation of hardware and software utilities.

**7. CONCLUSION**

The main objective of the project **“Space Shooter Game”** was to design and develop a shooter game and learn the maximum amount of things while doing. The different graphic elements of the game were designed. As a result the game is composed of one single perfectly playable level which includes many different objects and obstacles to overcome. Overall the game logic and the objects were designed and subsequently implemented. The obtained game can be played at different operating systems and it can be easily ported to any of the current generation game consoles.

Finally, I would like to highlight that most of the things done throughout this project were learned while developing it. I had very good experience in the whole process game development.

**8. FUTURE ENHANCEMENT**

In this project, I have done all the necessary to achieve the main targets by designing and developing my own game. However, I already have a few ideas to extend this work and improve the game:

* To create more levels with different game-play options.
* To make a different kind of reward logic, like including a multiplier to calculate the final score.
* To make different types of ships that can be used as the Player.

**9. BIBLIOGRAPHY**

**Reference Books**

1. [Sloan Kelly](https://www.google.com/search?sa=X&rlz=1C1RLNS_enIN995IN995&sxsrf=APq-WBvMfrD0SiQNlmXdvAdYR8RbXnpCXQ:1650438343409&q=Sloan+Kelly&stick=H4sIAAAAAAAAAOPgE-LVT9c3NEzOTTaOz7XIUoJyTYtNc-Nz8rRkspOt9JPy87P1y4syS0pS8-LL84uyrRJLSzLyixaxcgfn5CfmKXin5uRU7mBl3MXOxMEAAOvwL11UAAAA&ved=2ahUKEwiIy7r5iaL3AhU_TGwGHarxCuwQmxMoAHoECEEQAg), “Python, PyGame and Raspberry Pi Game Development”, Galgoties Publication, 23 December 2016.
2. [Will McGugan](https://www.google.com/search?sa=X&rlz=1C1RLNS_enIN995IN995&sxsrf=APq-WBuc4IAuFCFdsDuPRanIrGVPGO_bKw:1650438348525&q=Will+McGugan&stick=H4sIAAAAAAAAAOPgE-LRT9c3NErKTbcsMs9R4gXxDJPMC81Nc8pytGSyk630k_Lzs_XLizJLSlLz4svzi7KtEktLMvKLFrHyhGfm5Cj4JruXpifm7WBl3MXOxMEIAHh2xPJUAAAA&ved=2ahUKEwjD6_L7iaL3AhXRRmwGHV7vBYwQmxMoAHoECFAQAg), [Harrison Kinsley](https://www.google.com/search?sa=X&rlz=1C1RLNS_enIN995IN995&sxsrf=APq-WBuc4IAuFCFdsDuPRanIrGVPGO_bKw:1650438348525&q=Harrison+Kinsley&stick=H4sIAAAAAAAAAOPgE-LRT9c3NErKTbcsMs9R4gXxDNPNDQyN09MrtGSyk630k_Lzs_XLizJLSlLz4svzi7KtEktLMvKLFrEKeCQWFWUW5-cpeGfmFeekVu5gZdzFzsTBCAAo5kavWAAAAA&ved=2ahUKEwjD6_L7iaL3AhXRRmwGHV7vBYwQmxMoAXoECFAQAw), “Beginning Python Games Development”, Publication,, 21 June 2015.
3. Shoot Em Up, Wikia, Shmup genre history, [Online] Available: shmup.wikia.com/wiki/Golden\_Age, last accessed: may 2017.
4. Unity Technologies, Unity - Game Engine, [Online] Available: https://unity3d.com/es, last accessed: june 2017.

#### Reference Websites

1. [**https://www.pygame.org/docs/**](https://www.pygame.org/docs/)**.html**
2. [**https://www.geeksforgeeks.org/introduction-to-pygame/.html**](https://www.geeksforgeeks.org/introduction-to-pygame/.html)
3. **https://www.w3schools.com/graphics/game\_score.asp**