**Mini Project Report (KCS-354)**

**on**

**Python Game using Pygame Module**

**Submitted in partial fulfillment for award of**

**BACHELOR OF TECHNOLOGY**

**Degree in**

**COMPUTER SCIENCE & ENGINEERING**



**2022-23**

**Under the Guidance of: Submitted By:**

**Ms.Deepti Goel**  **Anurag Kumar Sahani (2200330100058)**

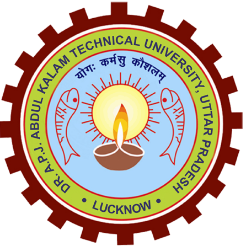
**Ms.Palak Masson Anuj Shubham Arya (2200330100056)**

**Assistant Profesor**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**RAJ KUMAR GOEL INSTITUTE OF TECHNOLOGY**

**DELHI-MEERUT ROAD, GHAZIABAD**

****

**Affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow**

**Synopsis:**

**Game Development with Python**

**Introduction:**

Python, a versatile and powerful programming language, has gained popularity in the field of game development due to its simplicity, readability, and extensive libraries. Leveraging frameworks and libraries, developers can create engaging and interactive games across various genres.

**Key Libraries and Frameworks:**

**Pygame:**

Pygame is a popular cross-platform set of Python modules designed for creating 2D games.

It provides modules for handling various aspects like graphics, sound, input, and more, making it suitable for both beginners and experienced developers.

**Pyglet:**

Pyglet is a cross-platform gaming library for Python that focuses on simplicity and ease of use.

It supports multimedia, windowing, and user input, making it suitable for creating games and multimedia applications.

**Godot Engine with Python:**

Godot Engine, a powerful open-source game engine, supports Python as one of its scripting languages.

Developers can leverage Godot's visual editor and use Python for scripting to create 2D and 3D games.

**Ren'Py for Visual Novels:**

Ren'Py is a visual novel engine that uses Python scripting. It is specialized for creating interactive stories and visual novels.

It simplifies the creation of narrative-driven games with its dedicated tools and scripting capabilities.

# TABLE OF CONTENTS

**CHAPTER NO. TITLE PAGE NO.**

**LIST OF TABLES** 5

**LIST OF FIGURES** 5

**1. INTRODUCTION** 6-8

Game Development process 7-8

**2. HARDWARE AND SOFTWARE REQUIREMENTS** 9

2.1 HARDWARE REQUIREMENTS 9

2.2 SOFTWARE REQUIREMENTS 9

**3. PROJECT MODULE DESIGN**  10-11

**4. PROJECT SNAPSHOTS**  12-14

**5. LIMITATIONS 22-23**

**6. FUTURE SCOPE** 24

**7. CONCLUSION**  25

**8. REFERENCES**  26

**LIST OF TABLES**

**CHAPTER NO. TABLE NO. TITLE PAGE NO.**

1 1 Firebase Authentication snap 12

LIST OF FIGURES

**CHAPTER NO. TITLE PAGE NO.**

1. Figures : Firebase Authentication 12-14

2 Figures : Output of Quiz App 15-19

4 Figures : Code for MainActivity,kt 22

**Game Development Process:**

**Design:**

Define the game concept, storyline, characters, and overall gameplay.

Create a design document outlining game mechanics, levels, and assets required.

**Implementation:**

Choose a suitable game development library or framework based on project requirements.

Write code in Python to implement game logic, graphics, sound, and user input.

Use object-oriented programming principles for organizing code and improving maintainability.

**Testing:**

Conduct thorough testing to identify and fix bugs or issues.

Test game mechanics, user interactions, and performance on various platforms.

**Deployment:**

Package the game for distribution on different platforms.

Utilize tools like PyInstaller or PyOxidizer to create standalone executables.

**Community and Iteration:**

Engage with the gaming community to gather feedback and make improvements.

Iterate on the game based on player feedback, fixing issues, and adding new features.

**CHAPTER-1**

**INTRODUCTION**

**Pygame is a cross-platform set of Python modules designed for writing video games. It provides a simple and easy-to-learn interface to handle various aspects of game development, such as graphics, sound, input devices, and more. Pygame is built on top of the Simple DirectMedia Layer (SDL) library, which is a low-level multimedia library commonly used in game development.**

Key Features:

1. Graphics:

• Pygame allows developers to create 2D games by providing a range of functions for drawing shapes, images, and handling sprites.

• The library supports hardware-accelerated rendering for improved performance.

2. Sound:

• Pygame includes modules for handling sound effects and music.

• Developers can load and play different sound formats, control volume, and manage multiple channels.

3. Input Handling:

• Pygame provides functions to handle user input, including keyboard, mouse, and joystick input.

• Input events can be easily captured, allowing developers to implement interactive gameplay.

4. Collision Detection:

• Pygame simplifies collision detection by providing functions to check for collisions between sprites and game objects.

• This feature is crucial for implementing game mechanics such as player-enemy interactions.

5. Event Handling:

• Pygame's event system allows developers to respond to various events like key presses, mouse movements, and window events.

• This enables the creation of responsive and interactive games.

6. Community and Resources:

• Pygame has an active community that contributes to tutorials, documentation, and additional libraries.

• Numerous resources are available, making it easier for developers to learn and troubleshoot.

## CHAPTER 2

## HARDWARE AND SOFTWARE REQUIREMENTS

Python:

Pygame is compatible with Python 2.7.x and Python 3.x versions. However, it is recommended to use Python 3 for new projects as Python 2 is no longer officially supported.

Operating System:

Pygame is designed to be cross-platform and can run on various operating systems, including Windows, macOS, and Linux.

SDL Library:

Since Pygame relies on the Simple DirectMedia Layer (SDL) library, the system must have the necessary SDL dependencies installed. Pygame will handle this automatically during installation.

Dependencies:

Pygame has minimal dependencies beyond Python and SDL. The specific dependencies may vary depending on the features used within Pygame.

Hardware Requirements:

Pygame is lightweight and doesn't have stringent hardware requirements. It can run on systems with modest hardware specifications.

## CHAPTER 3

PROJECT DESIGN

**Game Report: "Cats Day Out"**

**Game Overview:**

"Cats Day Out" is a visually engaging 2D side-scrolling game developed using the Pygame library. The game combines simple controls, vibrant graphics, and dynamic gameplay to create an enjoyable experience for players.

**Key Features:**

**Graphics and Design:**

The game features high-quality graphics, with well-designed characters, obstacles, and background elements.

The use of images for the player, obstacles, and coins adds a playful and charming aesthetic to the game.

The resizing functionality ensures a responsive and visually appealing experience on different screen sizes.

**Gameplay:**

The core gameplay involves controlling a cat character that moves horizontally, jumps, and collects coins while avoiding obstacles.

The jumping mechanic adds an element of skill, requiring players to time their jumps to navigate through obstacles successfully.

The inclusion of obstacles and coins creates a balanced and engaging challenge for players.

**Audio:**

The game includes sound effects, such as jump and coin collection sounds, enhancing the overall gaming experience.

The background music contributes to the game's atmosphere without being overwhelming, and its adjustable volume is a thoughtful feature.

**User Interface:**

The user interface is well-organized, featuring a score display, controls graphic, and developer logo.

The inclusion of a game over pop-up with a retry button provides a clear indication of the end of a session and encourages replayability.

**Introduction Logo:**

The addition of an introduction logo that appears for a set duration before the game starts adds a professional touch and builds anticipation.

**Coins Spawn Mechanism:**

The adjustment to the coin spawn mechanism allows multiple coins to be generated simultaneously, providing players with more opportunities to increase their score.

## CHAPTER 4

**PROJECT SNAPSHOTS**

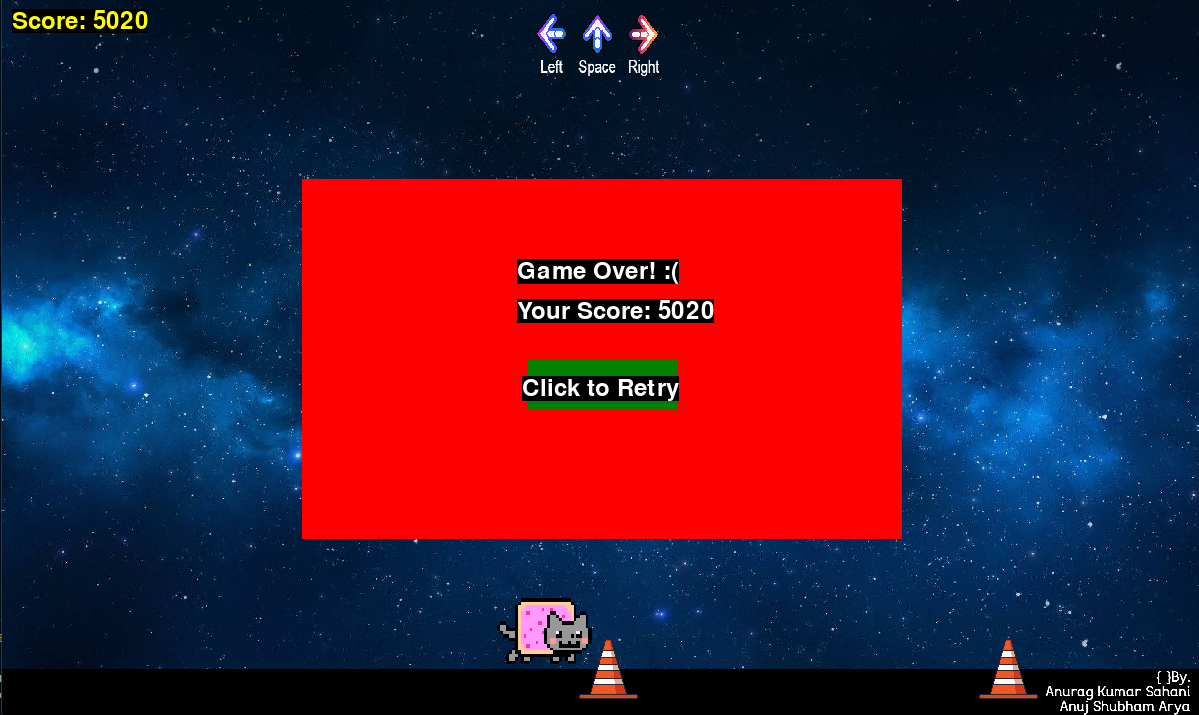
A black background with white text

Description automatically generated

Snapshot of game’s loading screen



snapshot of game



snapshot of game

A screen shot of a computer program

Description automatically generated

Snapshot of code

## CHAPTER 5

**LIMITATIONS**

Limited 3D Graphics Support:

Pygame is primarily designed for 2D game development. While there are some 3D graphics capabilities, they are limited compared to dedicated 3D game development libraries.

Performance Constraints:

Pygame may not be as performant as some other game development frameworks, especially for resource-intensive or graphics-heavy applications. For complex games or real-time simulations, more specialized engines may be preferable.

Lack of Built-in Physics Engine:

Pygame does not include a built-in physics engine. Game developers need to implement physics simulations manually or integrate third-party physics libraries.

Limited GUI Development:

Pygame is not the best choice for creating complex graphical user interfaces (GUIs). Other libraries and frameworks, such as Tkinter or PyQt, are better suited for this purpose.

Mobile Development Challenges:

While there are ways to deploy Pygame applications on mobile devices, it may not be as straightforward as using frameworks specifically designed for mobile development.

Limited Asset Management:

Pygame does not have built-in tools for complex asset management. Developers need to handle loading and managing assets, such as images and sounds, manually.

Community and Documentation:

While there is an active community around Pygame, it may not be as extensive as communities for some other game development engines. Additionally, the documentation may not cover every aspect in great detail.

Not a Full-Featured Game Engine:

Pygame is a library rather than a full-featured game engine. It provides essential components for game development, but developers may need to implement additional features themselves or use supplementary libraries.

Limited Built-in Effects:

Pygame has limited built-in support for graphical effects, such as shaders and particle systems. Developers might need to implement or integrate additional libraries for more advanced visual effects.

Cross-Platform Consistency:

Achieving consistent performance and behavior across different platforms might require additional effort, especially when dealing with platform-specific nuances.

**CONCLUSION**

"Cats Day Out" successfully combines fundamental gaming elements, including responsive controls, obstacle challenges, and dynamic gameplay, to create an entertaining and enjoyable 2D platformer. The incorporation of graphics, sound effects, and additional visual elements enhances the overall user experience. The game strikes a balance between simplicity and complexity, making it accessible to players of varying skill levels. With its engaging mechanics and well-implemented features, "Cats Day Out" provides a delightful gaming experience for those seeking a casual yet rewarding adventure.

By learning more about Pygame we have discovered

Python Language: Pygame is built on top of the Python programming language, which is known for its simplicity and readability. For beginners, Python provides an excellent entry point into programming, allowing them to focus on game development concepts rather than intricate syntax.

Ease of Learning: Pygame's straightforward API and clear documentation make it accessible to beginners. It simplifies many aspects of game development, enabling learners to quickly grasp fundamental concepts such as graphics rendering, event handling, and collision detection.

Rapid Prototyping: Pygame facilitates rapid prototyping, allowing developers to experiment with game ideas quickly. Beginners can see the immediate results of their code, making the learning process more engaging and iterative.

Community Support: Pygame has a vibrant and supportive community. Beginners can find a wealth of tutorials, forums, and resources to help them overcome challenges and gain insights into best practices in game development.

Cross-Platform Compatibility: Games developed using Pygame can be easily ported to different platforms, including Windows, macOS, and Linux. This cross-platform compatibility is advantageous for reaching a broader audience.

Versatility: Pygame is not limited to game development; it can also be used for multimedia applications, simulations, and interactive educational programs. This versatility allows beginners to explore a wide range of projects beyond traditional game development.

Graphics and Sound Integration: Pygame simplifies the integration of graphics and sound into games. Beginners can focus on creating visually appealing and immersive experiences without delving deeply into low-level graphics programming.

Object-Oriented Programming (OOP): Pygame encourages the use of object-oriented programming principles, which are fundamental to game development. Beginners can learn to structure their code in a way that promotes code organization, reusability, and maintainability.

Transition to More Advanced Frameworks: Learning Pygame provides a solid foundation for transitioning to more advanced game development frameworks and engines. The concepts learned in Pygame are transferable to other environments, enabling learners to tackle more complex projects in the future.

Motivation and Creativity: Creating games with Pygame allows beginners to express their creativity and see tangible results. The ability to build interactive and visually appealing projects can be a powerful motivator for individuals new to programming.