Interactive Game Console using Python

1st Mitrajeet J. Golsangi
Department of Engineering Science and Humanity(DESH)
Vishwakrama Institute of Technology
Pune, India
mitrajeet.golsangi20@vit.edu

2nd Vedant M. Gokhale
Department of Engineering Science and Humanity(DESH)
Vishwakrama Institute of Technology
Pune, India
vedant.gokhale20@vit.edu

3rd Shriram P. Gole
Department of Engineering Science and Humanity(DESH)
Vishwakrama Institute of Technology
Pune, India
shriram.gole20@vit.edu

4th Aayush A. Gore
Department of Engineering Science and Humanity(DESH)
Vishwakrama Institute of Technology
Pune, India
aaysh.gore20@vit.edu

5th Ashutosh S. Gore
Department of Engineering Science and Humanity(DESH)
name of organization (of Aff.)
Pune, India
ashutosh.gore20@vit.edu

6th Kaushal K. Gore
Department of Engineering Science and Humanity(DESH)
name of organization (of Aff.)
Pune, India
kaushal.gore20@vit.edu

***Abstract*—The Paper focuses on the projects indivisuals components and stresses on the importance of why the team has chosen the Python libraries for the development if the games. It provides the readers with the information on each indivisuals game and their respective algorithm as to ensure a little more clarity on the working of the project. It also touches up on the points of future scope of the project and states the current limitation of the same.**

I. Introduction

A. What is PyGame?

PyGame is an open source (LGPL license) library used majorly to create graphical interfaces, more specifically games using Python. The library being based on Python is device and OS independent and gives the developer various tools for graphics development. Pygame adds functionality on top of the excellent SDL library. This allows you to create fully featured games and multimedia programs in the python language.

B. Features of PyGame

- Multi core CPUs can be used easily
- Uses optimized C and Assembly code for core functions
- Comes with many operating systems
- Truly portable
- Easy to use
- Many games have been published
- Does not require a GUI to use all functions
- Small amount of Coding required
- Modular approach

C. Why use PyGame?

PyGame is the easiest graphics development framework in python. The big advantages are that it allows you to handle graphics, collisions, player input and most of all, people are more familiar with pygame than other modules. The point about others using pygame counts, because it's not ”easy” and you'll likely have plenty of question. You'd still have to learn the coding stuff. You COULD right your own graphics and math and time and event handler modules... oh, and sound/music mixer. If you don't want to take the ”easy” route, you'll need to do all this all. Pygame has had 20 years and a lot of good coders working on it. The advantage is not having to do all that and instead write games.

II. About the Project

The project is broken into 7 major chunks of code

1) Fruit Ninja Game
2) Chess Game
3) Stone Paper Scissors Game
4) 2048 Game
5) Tic-Tac-Toe Game
6) Snake Game
7) Main Function

The project has six indivisuals games created by the six group members. The main function is the code which contains a GUI for incorporating and merging all the games in a single one. The main libraries used in implementing the games are PyGame, TKinter and Ursina. All games are cross platform compatiable being developed in Python. A breif introduction of the indivisual games are described in the below section and Algorithms for the same are given for a better understanding of the code workflow.

III. Indivisual Games

1) **Fruit Ninja:** The game consists of only one major file containing both the logic and GUI of the game. The game consists of a basic window with the random fruit generation logic.
2) **Chess Game:** The game has a main structure of 3 files The chessMain file contains the main PyGame code for the implementation of the GUI The chessEngine contains the board layouting and the logic of how the chess pieces move and some special moves like the castling and enpassant
3) **Stone Paper Scissors:** The game is made in tkinter and consits of a basic logic containing the games GUI and a basic if else statement for the move creation
4) **2048:** The game uses the random logic to generate the tiles and basic arithmetic operations for the addition of the tiles.
5) **Tic-Tack-Toe:** The game is made using the ursina engine it contains the base code for the games GUI and logic for assesing the game is decided using a switch case
6) **Snake Game:** The main file consists of the basic GUI for the game and the game ends when the snake hits the borders of the created game window
7) **Main File:** The main file consist of a GUI made in PyGame and contains the code for a basic mario style GUI wherein the user controls a character which navigates The user to the required game icon and then the respective game is launched

IV. Game Algorithms

A. Snake Game

1) Start
2) Initialize a PyGame window
3) Creation of the Snake and fruit
4) Changing the fruit position randomly if the snake passes the fruit co-ordinates
5) If snake is inside the window goto step 3
6) Else End

B. Chess Game

1) Start
2) Creating a Chess board
3) Placing Chess pieces on the created board
4) Getting the users moves
5) Validating the moves
6) If moves are valid displaying the moves
7) If two player goto step 4 for second user
8) Else creating best possible move
9) Displaying the move
10) Display which color has win or created a stalemate
11) End

C. Tic-Tac-Toe Game

1) Start.
2) Creating Game window and setting positions for board.
3) Creating player, cursor.
4) Creating a board (2D list (3*3)).
5) writing player altering logic.
6) Defining Winner Class and writing winning logic.
7) Display Winners.
8) Stop.

D. 2048 Game

1) Start
2) Importing required Libraries
3) Creating GUI Creating a 4*4 NumPy array of Zeros
4) Logic for random number generation
5) Logic for making moves and arithmetic operations
6) Creating background grid
7) Key Mapping
8) Logic for Game Over condition
9) Displaying everything on GUI
10) End

E. Stone Paper Scissors

1) Start
2) Using tkinter creating window
3) Randomly generating input of computer
4) Assessment the game
5) Update score
6) Display Winner
7) End

F. Fruit Ninja

1) Start
2) Initialise PyGame and create a window
3) Define colors
4) Creating fruit dictionary
5) Holding data of random fruit generation using dictionary
6) Draw fonts and player lives
7) Showing game over display and front display
8) Create gaming while loop
9) Calling all functions defined in upper code
10) Keep loop running up till the bombs cut 3 times
11) Game Over screen after loop termination
12) End

G. Main Function

1) Start
2) Importing required Libraries
3) Creating GUI
4) Defining individual game logo positions
5) Loading required images into variables
6) Rendering all images and the character
7) Logic for launching specific games
8) Key mapping for character
9) Game launch condition and end condition
10) End

V. Future Scope

The games are still in development and have a huge potential for the future scope. Some points include:

- To create even better GUI for all the games
- To increase the number of games
- To allow the users to add games and suggestion
- To link the game to internet for better accesibility
- To Deploy the games

VI. Limitations

As mentioned before the game still being in development the game has certain Limitations:

1) Low Resolution Games
2) Performance efficiency is reletively lower
3) The games are 2 dimentional

VII. Conclusion

- The Application is able to manage to provide multiple games on single platform.
- Gaming Platform using python can be used by every individual and useful for Logic development and for entertainment.
- The system has been developed with simple user interface, which is attractive.

VIII. Member Contributions

1) 11 - Vedant Gokhale - 2048 Game, Main Function
2) 12 - Shriram Gole - Fruit Ninja Game
3) 13 - Mitrajeet Golsangi - Chess Game
4) 14 - Aayush Gore - Tic-Tac-Toe
5) 15 - Ashutosh Gore - Snake Game
6) 16 - Kaushal Gore - Stone Paper Scissors Game