Synopsis : PSDL Mini Project
Title: Productivity Game

1) Group Member details-

Number of members : 4
1)UCE2022556 - Ishita Lele
2)UCE2022571 - Neha Chatterjee
3)UCE2022572 - Vedika Nehe
4)UCE2022573 - Sanika Nikam

2) Problem Statement- Creating a productivity booster game, which allows users to enter their to-do list , and on completion of tasks, the users earn coins, using which they can play a fun game.

3) Keywords- pygame, tkinter, gamification, productivity, importlib, openpyxl

4) Abstract-

Our project aims to combine the idea of productivity and task management with the concept of gamification. We tried to create a reward based system which encourages users to finish their tasks and as a break and motivation , play a fun runner game.
The application starts with a main menu, which includes buttons to the task list, the game and an option to exit it.
The task list is built using the user-friendly interface tools provided by the tkinter library in Python. Upon completion of a task the user gets coins. These coins can be used to play the fun runner game. This game serves the purpose of motivating the users to complete their tasks and also is a good break from work.
The game is built using the pygame library. We have also used the importlib library to handle the circular imports in the application. The sys library is also used to exit the application.
Apart from that, to store the high score and number of coins we have used the file handling concept to read and write updated values into text files in the backend. We have used the openpyxl library to store the tasklist in a .xlsx file in the backend.

5) Module-wise description-

Module 1 : The main menu-
Libraries used - Pygame, importlib
User defined modules imported- button class, coins class, demo(game), doing(task list)
Methods created -

- get_font() : used to set the font style and size
- play(): to call the demo module and play the game
- tasks() : to call the doing module and use the tasklist

Description - Importlib is used to call other modules during runtime. It has been used to call the main functions of demo and doing(which is the task list module). The pygame library is used to create the interface involving the text displayed and the buttons.

Module 2 : The to-do list-

Libraries used - Tkinter , messagebox from tkinter, importlib, openpyxl

User defined modules imported - coins class

Methods created -

- doing_main()-creates a window and all its components and contains the main loop
- new_task() - add a task
- deleteTask() - used to remove a task from the list after its completion. Will also display total coins earned upon completion of a task.
- goBack() - go back to the main menu

Description - This module is used to create an interface between the user and their to-do list. The to-do list is stored in an Excel file in the back-end, and using this interface, the user can add and remove tasks. When the user wishes to close the tasklist, they can click on the “Return to menu” button, which will destroy the tasklist window.

Module 3 : The game-

Libraries used - Pygame, importlib

Methods created - game_main() - contains entire game , load_high_score(), save_high_score(), display_score(), obstacle_movement(), player_animation(), collisions()

Description - The game is primarily built by making surfaces and rectangles of the images of- the background, foreground, the player, the obstacles(snail and fly). The score is incremented by using a simple counter variable and the high score is stored in a text file. After each play the score is compared with the file and stored if greater than the high score. Every time we press the game button in the main class, we check if the player has 100 or more coins. Only then the game starts. After completion of the game, 100 coins are deducted from the coins text file. The collisions are detected using the inbuilt functions of the library.

Module 4: The Button class-

Methods created:

- __init__(): to initialise all variables of the class
- update(): to apply underlying image to button
- checkforInput: to check if the coordinates of the mouse are within the bounds of the button’s rectangle. If it is, then it returns the boolean value “True”.
- changeColour(): to change the colour of the text on the button when the mouse hovers over it

The Button class is stored in a user defined button.py module. It’s __init__() function is used to initialise all the requisite variables. The button is created such that it has an underlying image, and the text written on top of it. If no underlying image is provided, the text is displayed atop a

transparent surface. On hovering the mouse over a button object of this class, the text changes colour to show that it has been detected.

Module 5: The coins Tracker-
In this module, we have 2 methods- coins_increase() and coins_decrease()

- coins_increase() gets activated once we finish a task in the task-list module.The user gets 100 coins and this also gets written in a text file in the backend.
- Similarly, the coins_decrease() method gets activated once we click on play game and the user has 100 or more coins. The coins are decremented by 100 and this also gets stored in the backend.

In both these cases, a message box made by using the tkinter library informs the user of the changes in their coins.

6) Technology selected- Pygame, importlib, Tkinter, openpyxl libraries

Technology Features covered -
Creating a game loop, displaying animation, storing high score using file handling on the user device, displaying a dynamic to-do list, and storing tasks in an Excel file on the user device.

7) References-

https://www.pygame.org/docs/
https://docs.python.org/3/library/tk.html
https://docs.python.org/3/library/importlib.html
websites referenced - stackoverflow , quora , geeksforgeeks
Various youtube video tutorials referenced