***Python Project***

***Game Development***

The Game has been constructed using three different python files, few images , few backgroundmusic. The latter is stored in its respective folder . That path is defined using the feature import of python use the feature of path from OS. Having all the code in the file is a boring job. The file becomes lengthy and is much more difficult to debut the single code . Importing required files in other files in other files eases our job. The code contains comments wherever required so that even a noob coder can understand the code .

First of all We are going to explain about the game:

***GAME NAME:*** SneGaVa

***Caption:*** Kill to Survive

***Controls:***

UP ARROW = To Jump

SPACE KEY = To Fire Bullets

LEFT ARROW = To Move in Left Direction

RIGHT ARROW = To Move in Right Direction

ENTER KEY = To Start the game in Start page

***Rules:***

We need to click Enter key to start the game immediately after opening the game.

If we hit the enemy with bullets the score in incremented by 2.

If hero touches the enemy or Traps, then the game is completed.

Finally, it is an Infinite Loop game.

Coming to the files , there are three of them , namely “Proj.py”,”main.py”,”enemy.py”.Explanation for them is as follows:

***Proj.py***:

This is the main part of the file. First in this file, we imported pygame, main.py, enemy.py and time. Then we initialized the PYGAME library. Then we added Logo , Resolution of the game , Name of the game , Background for the game , game over background , and Background music’s for Bullet sound and music while game is running, Trap images are also added.

Next we defined a class named Shooting to draw the bullets which contains coordinates, radius of the bullets, colour.

Then we defined a New function called display\_window. This is the main function because it contains all the content we need to display on the screen. Other than this function, other Classes or Functions just do Internal Part of the Game. In this function we gave if Game is not over it should display background, traps, bullets, hero, enemy etc. If game is over it displays the Game over Background.

Then we called the main\_character, Enemy with coordinates and width, height.

Then we gave a new page that is start page, which contains rules, we need to click Enter to start the game. For this, we called a While loop namely while Start.

If we start the game then it moves into the next loop called namely while alive. First we defined FramesPerSecond to set how much fast the game must play. Then we gave for loop to quit the game. We gave some conditions for bullets so that bullets will fire with some gap and the bullets won’t be continuous. Next we gave for loop for bullets, this for loop make the bullets hit enemy and adds score. For this, we defined invisible red boxes in main.py and enemy.py because we don’t know the exact coordinates of every part of the images of hero and Enemy, if bullets touch the enemy box the score is added by 2. After collision the bullets will be vanished using new inbuilt function pop.

Then we defined controls. First control we gave is to shoot bullets when we hit SPACE button. The next controls are Left and Right Arrows and in the same conditions we gave even the boundaries so that the hero won’t go beyond the screen

The next one is Up Arrow to Jump. The else condition in the jumping is most important and complicated part for jumping, it’s not just moving up, it should be able to move in a projectile direction.

The next and last conditions are given because when hero touches the enemy or Traps the game should be over. The first condition given if the red boxes of enemy collide with red box of hero then hero dies and game is completed. The next conditions are if the Hero touches any of the traps, then also game should be over.

Then we called the Function display\_window at last.

Then the last line of the program is to QUIT the game

***main.py:***

This file contains all the content of **main\_character(HERO).**

First of all we loaded sprites of moving right and left sprites in a list as there are many photos and we can access the list very easily.

Then we defined a class called main\_character.

First we initialised what all parameters needed coordinates, width, height, speed, move count, jump, box, etc. Then we defined a function called movement which consists the movement of the Main character (Hero).

***enemy.py:***

This file contains all the content of enemy.

First we loaded the enemy sprites of moving right and left in a list.

Then we defined a class called Enemy.

First we initialised what all parameters needed coordinates, width, height, speed, box, boundary, path etc. Then we defined a function called movement which consists the movement of Enemy.

Then we defined a new function move because the enemy should move itself automatically.

Here ends the explanation of the code.

***References:***

Code was done with some help from a YouTube channel “Tech With Tim”. The videos subsequently had there code uploaded in GitHub, reading and understanding the code helped us a lot. Images were taken from “Google”. Main Character sprites were taken from google and altered in GIMP and enemy sprites, back ground music were taken from GitHub of “Tech with Tim” and others were directly found.

***TEAM MEMBERS :***

GANESH BARMA - IMT2018024

SNEHITH SAI - IMT2018062

GNANA VAMSI PEDDU - IMT2018506