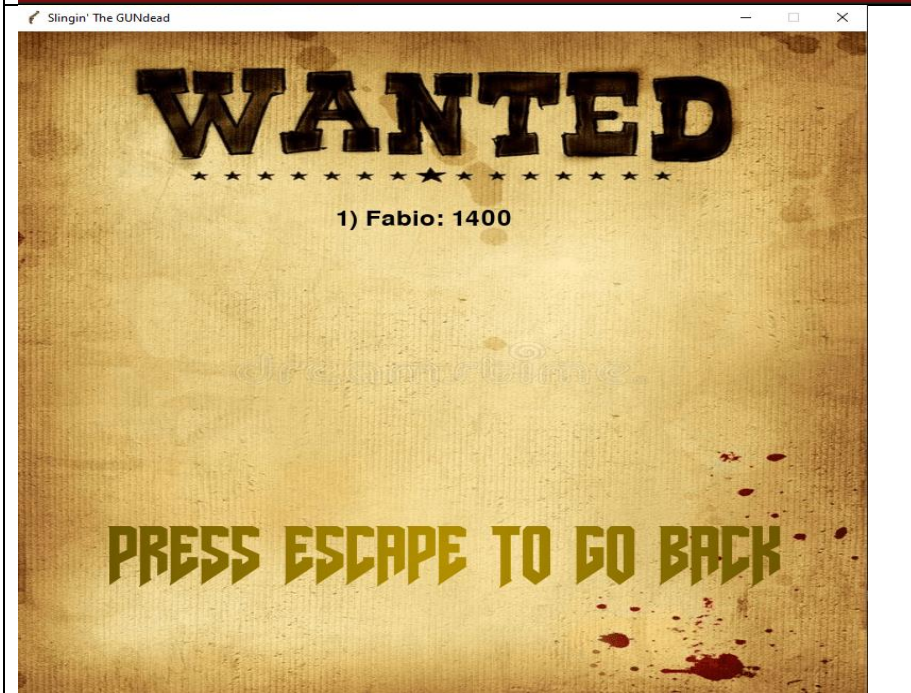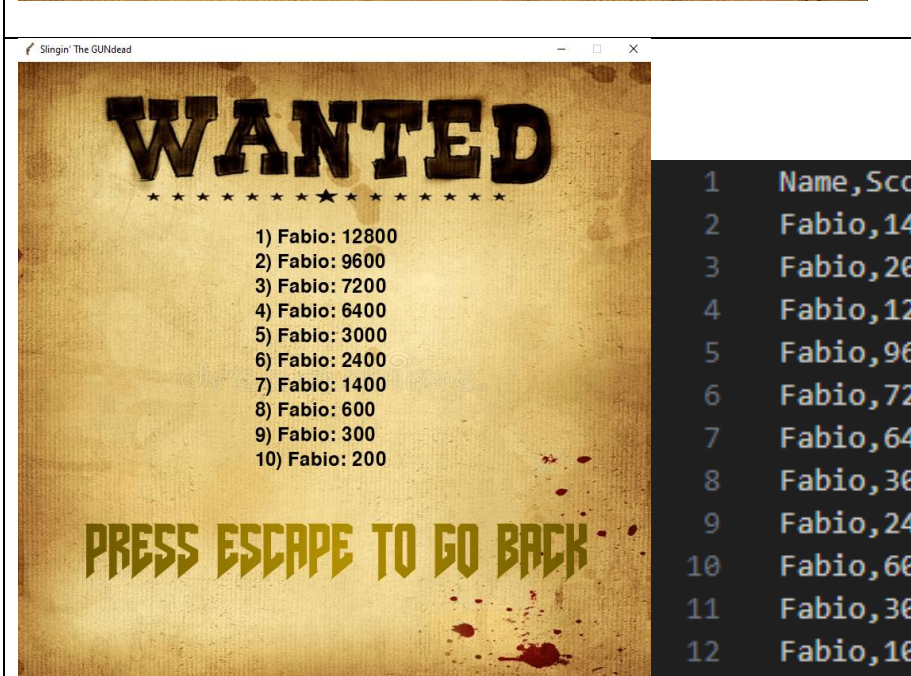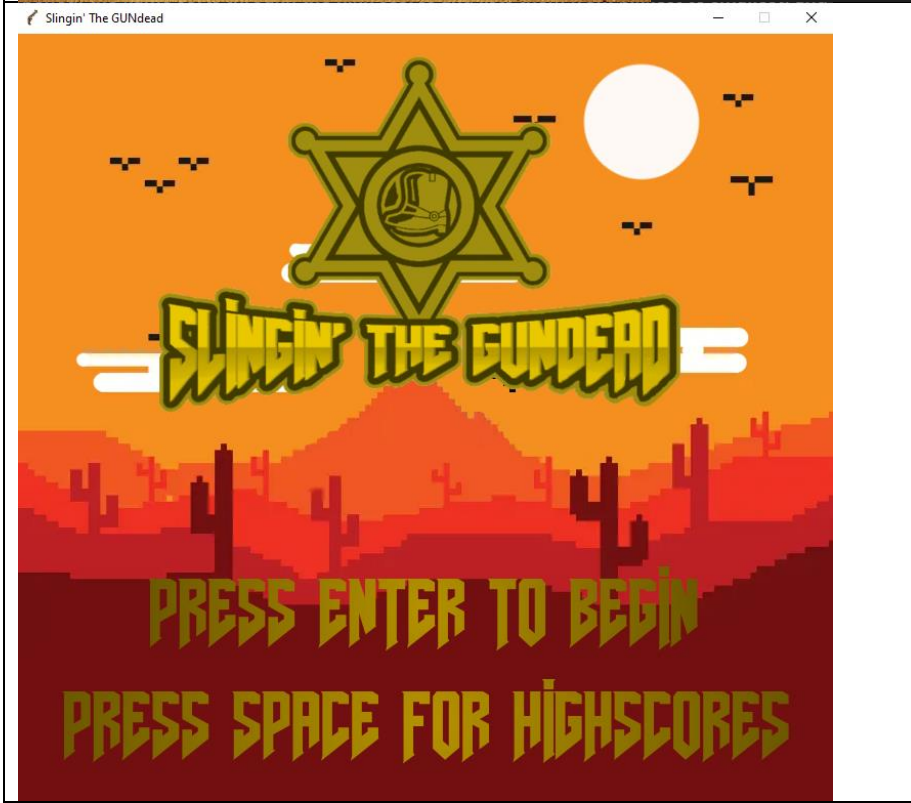<u>My Code Documentation</u>

What is the aim of my program?

- The aim is simply a game I created to have fun from an idea I had. I did not want to do the boring flight simulator project, so I chose to make a game, and from there, I decided to spend the past 2 months learning Pygame, and practicing each of my python modules to implement them into my game and make it a fully functioning game.
- The game starts at the main menu, where the player can either begin playing, or check the scoreboard for any high scores
- once the player decides to play the game, they are thrown into a 2d top-down shooter where your aim is to survive waves and kill as many zombies as possible.
- If the player dies, the game asks for user input, specifically the name, and once the name has been input, the game moves onto the game over screen, where it places your name and the score you achieved into the scoreboard CSV file.
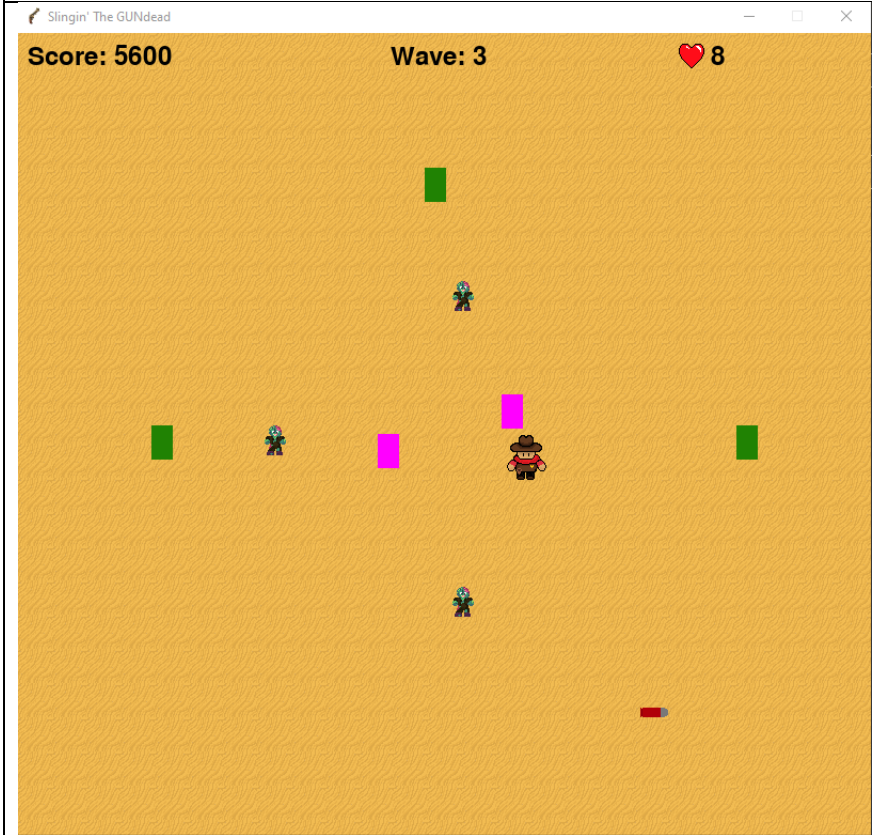
<u>Testing</u>

	<u>Starting the game:</u> Test result: - The game opens, background music plays, and the game awaits input from the player, with a choice of going to the high scores menu or jumping into the game. - The menu bar appears to have the name of the game, and the games' icon, indicating that they have been loaded correctly. Now I will progress into the high scores menu.
	<u>Pressing spacebar:</u> Test result: - This opens the leaderboard, with the background music still playing, where it will display the name of the person who played and the score they achieved. - Once 10 or more scores have been recorded, the intended function is that only the top 10 highest scores will be displayed. I will test this below.
	<u>Same menu with 11 scores:</u> Test result: - This demonstrated that once the leaderboard menu is opened, the game sorts through the high scores. - Once it is finished sorting them, it lists them from highest to lowest, and only lists the top 10, as intended. Now I will Press the escape key to test if it returns to the main menu.
	<u>Returning to the main menu:</u> Test result: - In the same instance of the game, whilst in the leaderboard menu, if the user presses the escape key, like indicated in yellow text, the game does return to the main menu. - It is still playing the music from when the game was launched. Now I will press enter to launch and test the game and its mechanics.
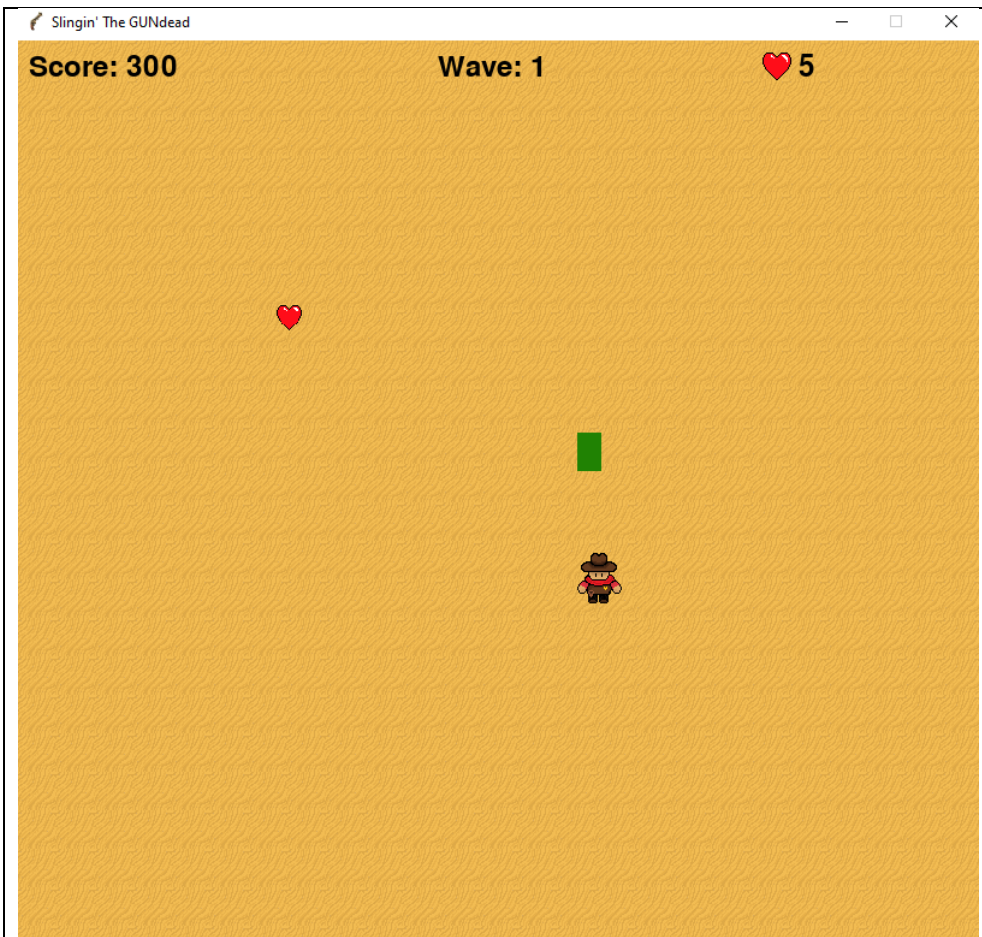
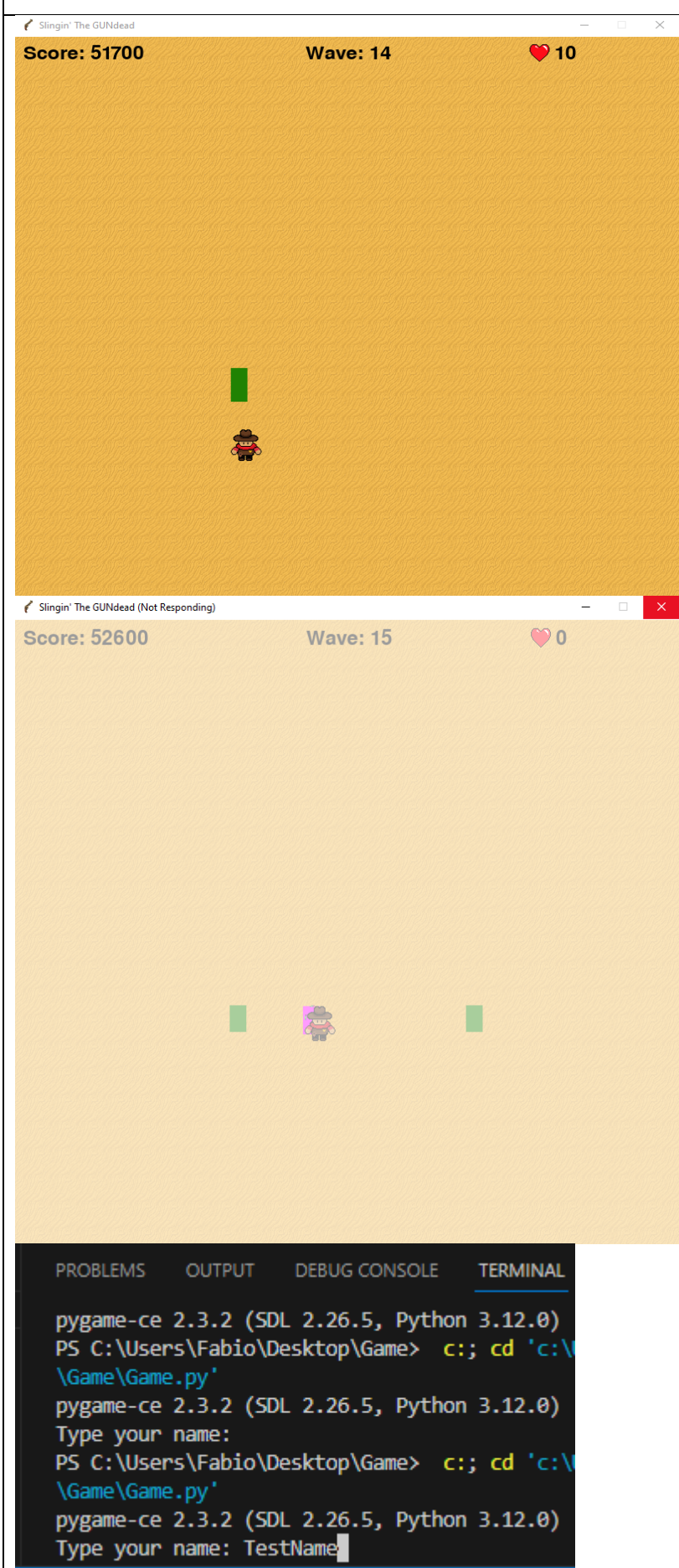|	
Slingin' The GUNdead — ☐ ✕ Score: 300 Wave: 1 ❤ 5	**The main game:** Test result: - Upon pressing the enter key, the menu music stops, and is replaced by the in-game music - the first wave of enemies spawn as intended chasing the player by checking the current player position and adding or subtracting based on said position. - Player movement is functional, using the WASD keys to move the player up, down, left and right. - Bullets fire, though accurately due to calculating the angle not being pixel perfect and instead gets a rough estimate. however, they fire and kill the enemies as expected. - When enemies die, they have a chance of spawning player HP, which when collided with the player sprite, disappears and adds 1 health to the players' current HP.
Slingin' The GUNdead — ☐ ✕ Score: 5600 Wave: 3 ❤ 8	**Main game progression:** Test result: - The score is increasing upon enemy deaths, as intended. - Player HP increases for each heart that the player picks up, being fully functional. - The wave number increases, demonstrating progression in the game, and for each wave, the number of enemies spawned increases, making the game more difficult. I will now try to set a new high score, to demonstrate that the high scores menu functions as it should upon reaching the game over screen and concluding the game loop.
Slingin' The GUNdead — ☐ ✕ Score: 51700 Wave: 14 ❤ 10 Slingin' The GUNdead (Not Responding) — ☐ ✕ Score: 52600 Wave: 15 ♡ 0 PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** pygame-ce 2.3.2 (SDL 2.26.5, Python 3.12.0) PS C:\Users\Fabio\Desktop\Game> c:; cd 'c:\ \Game\Game.py' pygame-ce 2.3.2 (SDL 2.26.5, Python 3.12.0) Type your name: PS C:\Users\Fabio\Desktop\Game> c:; cd 'c:\ \Game\Game.py' pygame-ce 2.3.2 (SDL 2.26.5, Python 3.12.0) Type your name: TestName	**Setting a new high score and inputting name:** Test result: - As demonstrated by the images, I set a new high score of 52600, and I purposefully reached 0 HP to end the game, as seen below in the second image. - When the player reaches 0 HP, the main game loop is set to false, so that it can move on to the game over screen, and this is proven by a prompt in the terminal. - The terminal asks to input a name, which I will leave as "TestName," so that I can validate whether it has saved properly in the high score menu. - Upon pressing the enter key on the name prompt, the game should continue into the game over screen. I will go ahead and test this now.

	Game over screen
	Test results: - Upon pressing the enter key, the game broke off the main game loop, and entered the game over screen. - In the background of the game over loop, it appends and writes the name of the player and the score to the CSV file, which we will check to see if it has saved correctly. - In this screen there is not much to do, as it is the end of the game, and the only accepted input is the enter key to exit the game. I will now reopen the game, to test whether the score has been appended into the scoreboard, and displays as the top score, comparing to the scores shown on the third image.
	New score appended
	Test results: - When The game was reopened, I clicked the space bar to open the high score menu. - Upon the menu being loaded, I immediately see the new high score I set during the test "TestName" listed, with its correct score achieved. This demonstrates that the code works as is, and has no major errors that impede progression, and runs perfectly.

Self-critique:

What worked:

- The game worked as intended, from player movement to enemy movement, enemy spawning, enemy killing, getting extra hearts upon enemy death, checking collision between player/enemy and enemy/bullet and doing the appropriate thing once either of those two happen.
- The game reads and writes correctly off of the CSV file and appends new data to it each loop to store the new score correctly.
- The program caught some errors when closing prematurely, excepting them and passing them as they did not affect the main game loop, only causing errors when closing in the main menu or the high score menu

What did not work/what I could have done better:

- I wanted to loop the game over, rather than have to close it completely, and reopen it to play again. I know why this happened, but I was simply unable to fix it. The error was that once the player reached 0 HP, and the menu moved on to the game over screen, and back to the main menu, the player HP was still at 0, and the game loop had not reset, potentially due to none of the variables being reset once the main game loop reinitiated, and this could be fixed by resetting all the values from the player and enemy classes and my variables back to their original set state, however I could not figure out how to do this, so instead I chose to simply close the game upon completion.
- If I had more time, I would like to learn how to make my code more efficient, and function properly as I wanted it to, as I am missing out on some features that I wanted to add such as powerups which would wipe enemies on screen, or make the player shoot faster and also more enemies including a ranged enemy that shoots bullets back at me, however I simply ran out of time due to bad management and simply not knowing how to program either of those two things, but with time I may have been able to do it.

Overall, I had a great time working on my project, and I am pleased with what I have achieved, and have proven to myself that I am more than capable at programming than I had originally thought. At the start, I was unable to even write a single piece of code, but from having HURRAH moments where I just figure out a fix to an issue that I was having for a long time made me feel accomplished, and not feel like I had impostor syndrome and could actually write a program, granted, with some help at the start when learning how to use pygame, which I will reference each link below here, however there are more links in the folder "Things that taught me pygame"

https://www.youtube.com/watch?v=7QX8y6R9Hi8&t=1s - by CodeNULL

https://www.youtube.com/watch?v=DHgj5jhMJKg - by Code Russ

https://www.youtube.com/watch?v=XKHEtdqhLK8 - by Bro Code

https://www.youtube.com/watch?v=ygopBQ7D3ig - by LeMaster Tech

https://realpython.com/python-csv/ - Reading and writing to CSV files

https://www.youtube.com/watch?v=AY9MnQ4x3zk - by Clear Code