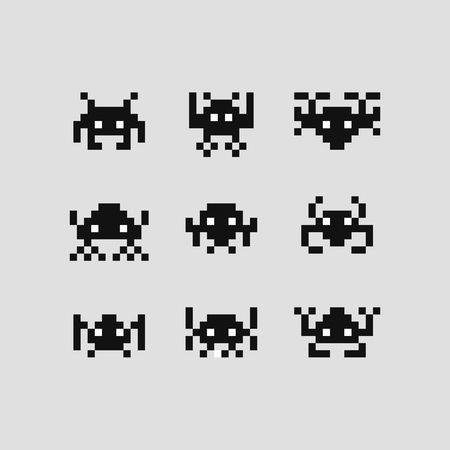
**Space Invaders**



**By Thomas William Burnage**

Contents

[Theory 3](#_Toc18097338)

[Why am I doing this? 3](#_Toc18097339)

[The History Of Space Invaders 3](#_Toc18097340)

[What is the gameplay like? 4](#_Toc18097341)

[My plan for my version of the game 5](#_Toc18097342)

[How will make the game 5](#_Toc18097343)

[How will I display my progress when I code my game? 6](#_Toc18097344)

[Practical 6](#_Toc18097345)

[How to install pygame 6](#_Toc18097346)

[1st Display of Code (25/06/2019): The basic Structure: 7](#_Toc18097347)

[The Code for the program so far: 7](#_Toc18097348)

[2nd Display of Code (30/06/2019): 9](#_Toc18097349)

[The Alien Features: 9](#_Toc18097350)

[The Code for the program so far: 10](#_Toc18097351)

[3rd Display of Code (02/07/19): 14](#_Toc18097352)

[The Life system: 14](#_Toc18097353)

[4th Display of Code (30/08/2019): 14](#_Toc18097354)

[The Scoring System: 14](#_Toc18097355)

[5th Display of Code: 15](#_Toc18097356)

[The Title Screen: 15](#_Toc18097357)

[Design: 15](#_Toc18097358)

[Making the aliens and tanks skins: 15](#_Toc18097359)

[Overview: 16](#_Toc18097360)

[Bibliography 17](#_Toc18097361)

# Theory

## Why am I doing this?

Ever since I was little, I have played video games and loved the idea of potentially making one when I was older. Recently I have been enjoying and playing retro games such as Galaga and Pacman and the one that struck me out the most was Space Invaders for how the controls are so simple yet the game becomes progressively harder over a set amount of time.

Not only this but I have also been reading a book called “Python Crash Course” by Eric Matthes and at chapter 12, it leads you and teaches you how to program a Space Invaders clone and I thought it would be a great time and opportunity to have a shot at making this beloved retro game.

This will be the first game I have made that isn’t text based so it will challenge me and help me develop my knowledge and understanding of how python can function. It will also introduce me to new modules with the new effects and applications.

## The History Of Space Invaders

Due to Space Invaders being the main motivation of this project, I feel it is justified to understand its history and the effect it had on society as a whole.

It was developed by Tomohiro Nishikado and apparently, he took inspiration from the book **“The War Of The Worlds”** by H. G. Wells and he was also inspired by **Star Wars** (N\A, 2019). During the original planning, instead of having the player destroying/killing aliens they had planned to have human soldiers instead. Although due to Taito’s intervention, they changed it to the alien ships that we all know and love today.

The Space Invaders game was released in 1978 by Taito (a Japanese company) and it became an immediate hit. Two years after its release in Japan, it was licensed for use in the United States During this time, it was released on coin-operated arcade machines and home gaming systems.

It can be easily said that Space Invaders was a revolutionary game due to it introducing many new concepts into the world of gaming. Two of the many new concepts that was introduced were:

*- A fully animated interlude*

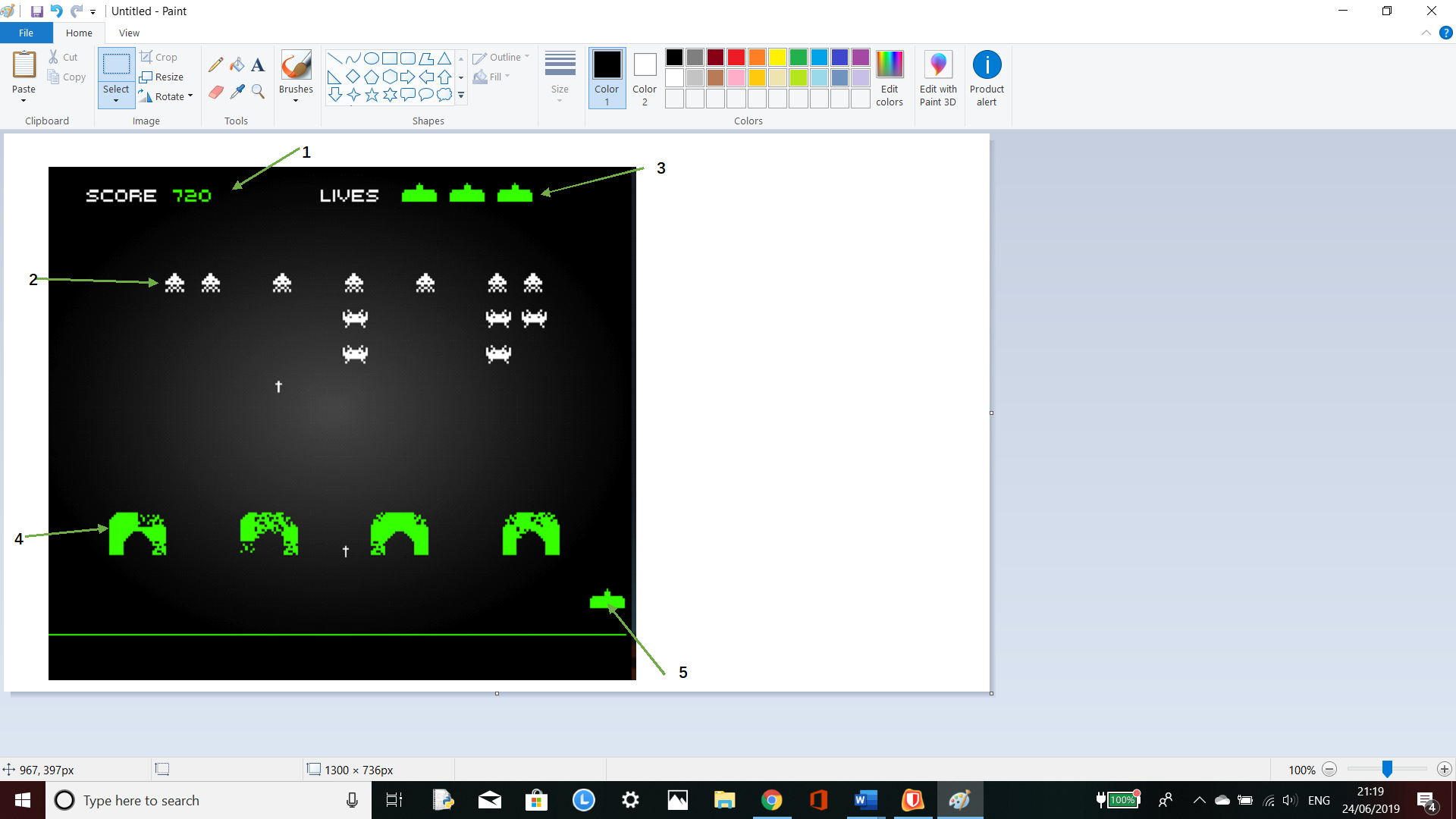
*-An original take on background music*

The music strikes me the most because the speed of it is increased or decreased depending on how close to the ground the aliens were to the bottom of the screen (Earth).

According to the classic games website, Space Invaders managed to accumulate more than 500 million dollars in revenue.

## What is the gameplay like?

This is gameplay image is sourced from <http://www.classicgaming.cc/classics/space-invaders/play-space-invaders> and is from a clone of Space Invaders, it is not the original version so some features may vary.



1. This is the score that the player collects by destroying the spaceships. The score added depends on the ship that was destroyed.
2. This is one of the five different alien spaceships, the spaceships on the bottom row is worth the least and it progresses in the intensity of the score as you move up the rows.
3. These are the amount of lives the player has, after being hit by a bullet by a spaceship, you will lose a life.
4. These are shields that protect the player from enemy fire, these are gradually destroyed when hit by a bullet. Once they are gone, they are gone for the rest of the game.
5. This is the player, the player can only move the sprite left or right and fire.

## My plan for my version of the game

For my version of this iconic game, I would like to adapt/ modify and add new features into the gameplay of it. Some of the features like the user only being able to move horizontally and only being able to hide behind temporary shields are good so I will not alter them.

Although, the features I will alter are as follows:

*-The life system*

The features I will like to add in are:

*- Power ups (Turret & Oily Tracks)*

*-Title Screen*

*-Customizable Skins for aliens and tank*

When I program these specific features, I will go into more detail about them.

## How will make the game

I will create my version of the game using Python 3 by taking advantage of a program called pygame. This is a program that contains a set of cross platform Python modules that are created with the idea of making it easier to write video games.

Although pygame isn’t initially installed on Python, it is saved in a Local Script file on Python so I will need to install it into my version of Python.

## How will I display my progress when I code my game?

As I write the program, there will be multiple things that I could talk about and many errors that I may encounter although it is not practical to document every error. This is due to errors being very specific to the programmer and this program will more than likely going to be over 1000s lines long in total.

However, in certain points I will display my code at certain points and I will state the major changes to my programs. These will be the following moments when I will display my code:

*-When I have completed the User controlled features (being able to move the tank, being able to shoot from the tank) aka the basic structure of my game\*

*-When I have created the features for the aliens.*

*-When I have made the life system*

*-When I have made the scoring system*

*-When I have made the Title Screen*

# Practical

## How to install pygame

In order for the game to run on Python 3, the person will need to install Python 3. Then go into the files and find the location of Python 3 ‘s Scripts files. After that they will then need to open up the command prompt and insert the pathway used to go into the Local file with the letters ‘ cd ‘ just before it.

Example:

*cd C:\Users\ExampleUsername\AppData\Local\Programs\Python\Python36-32\Scripts*

Then press enter.

After that they will need to write the following code:

*pip install pygame*

There should be a download undergoing where a bar is slowly going up, the user needs to wait until that is completed. Once it is completed, the user will need to open the Python 3 terminal (the one we just installed pygame onto) and check that it is installed. To do this they need to write:

*>>> import pygame*

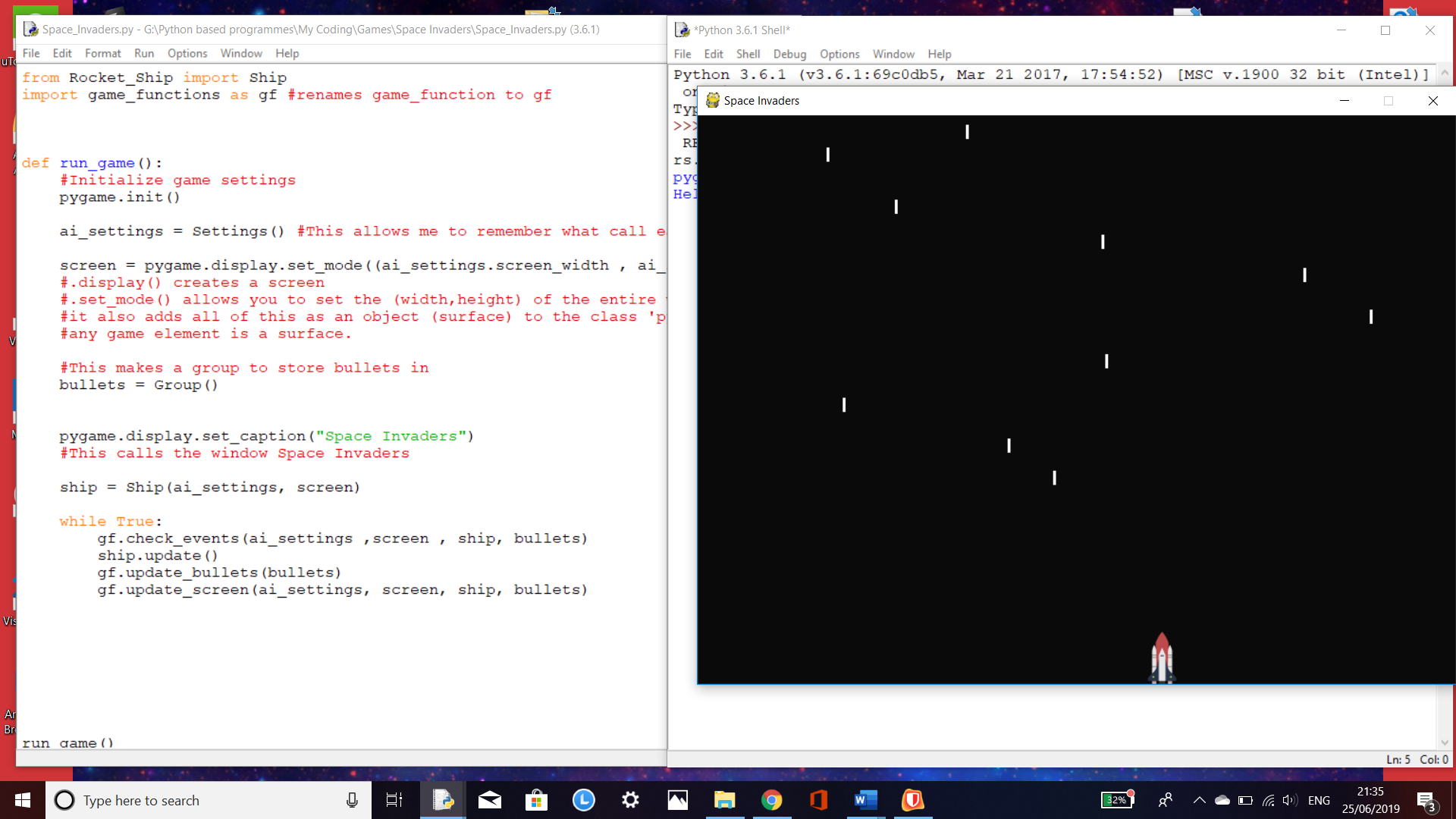
The response should be:  
>>>

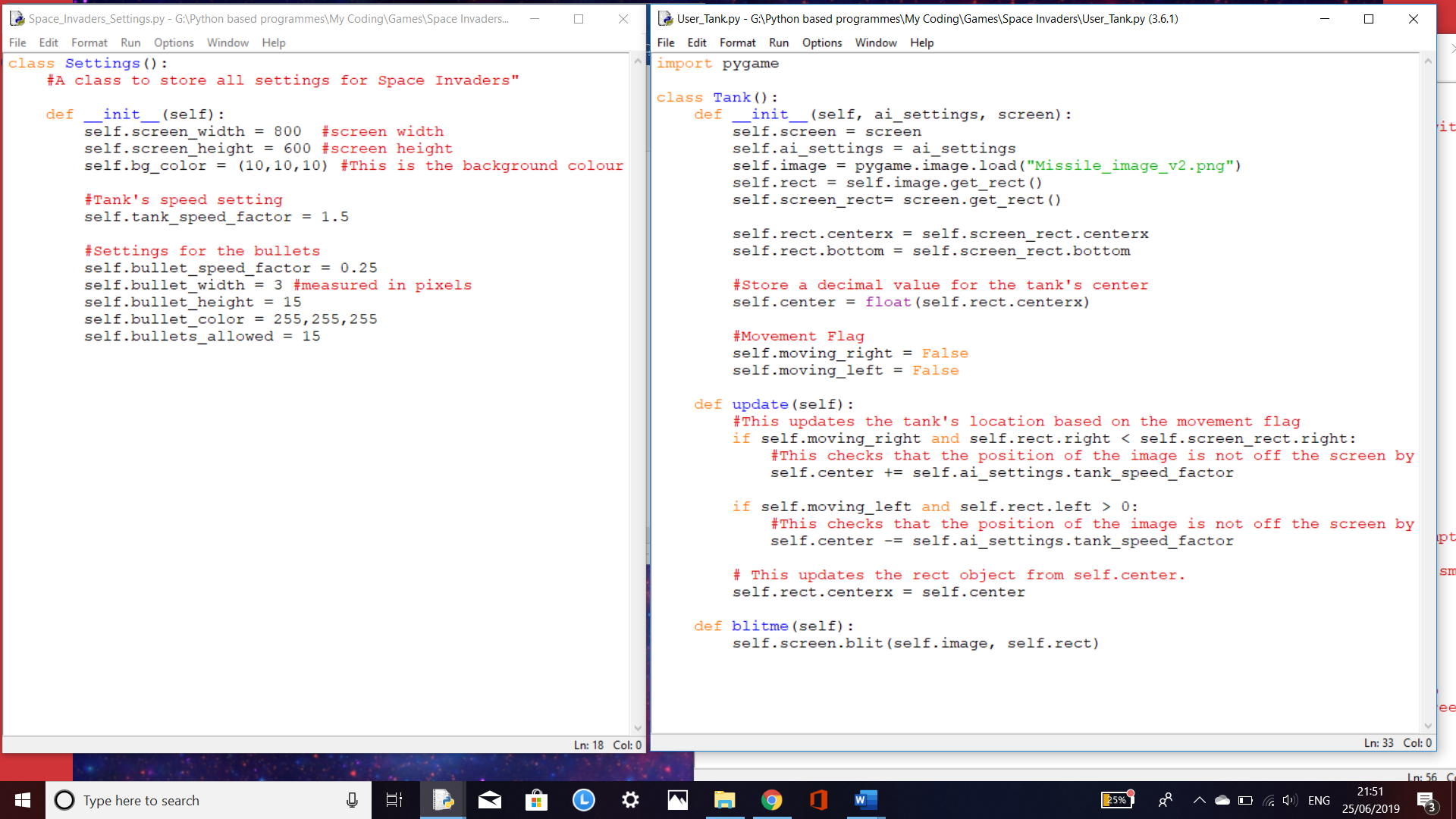
This means that pygame has been correctly installed onto your version of Python. Now it ready.

### 1st Display of Code (25/06/2019): The basic Structure:

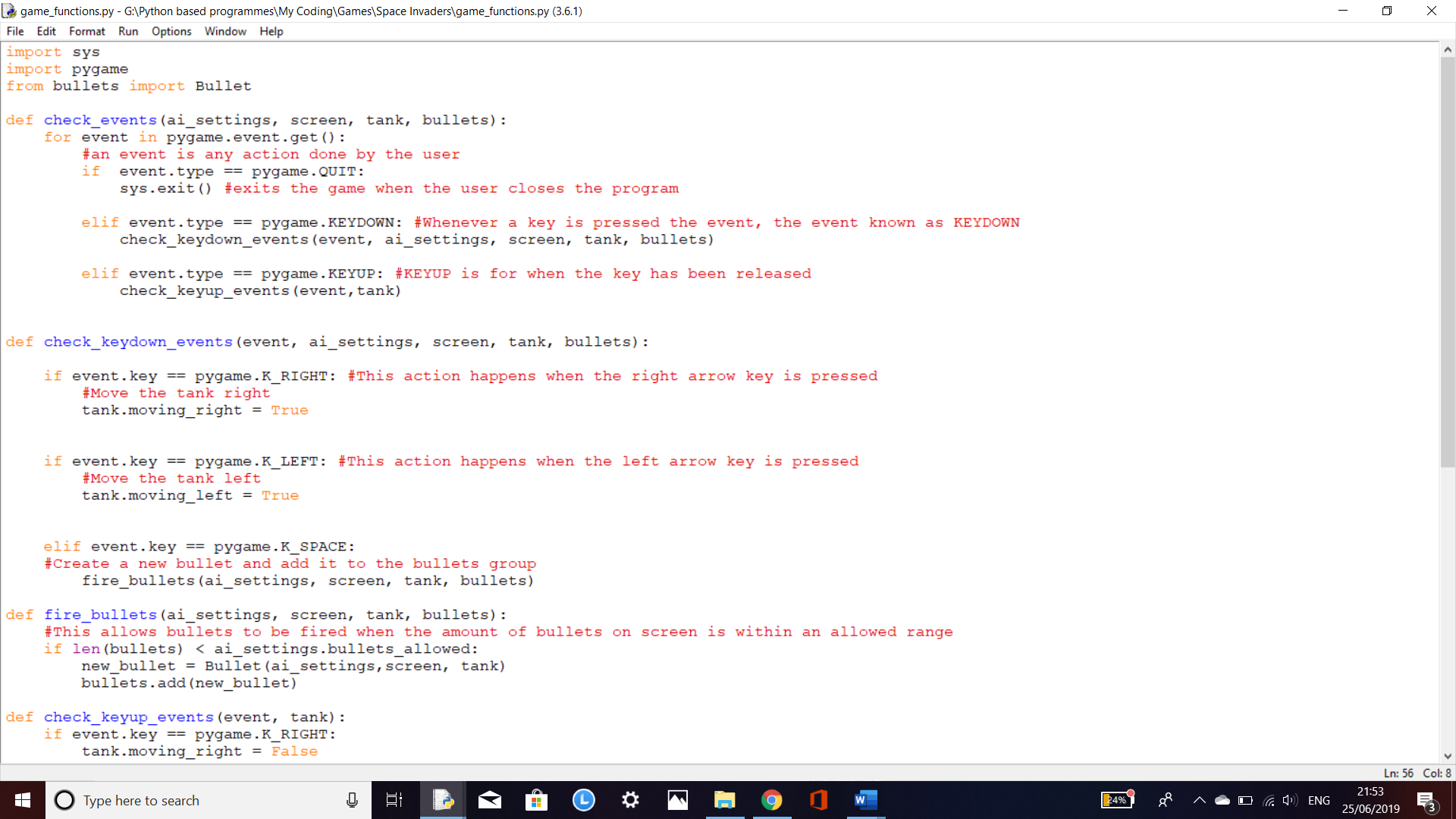
At this current point, I have made the basic bullet and movement systems and I have also managed to insert a basic placeholder image as the character that the user controls. The main program is shown in the following image next to the game running.

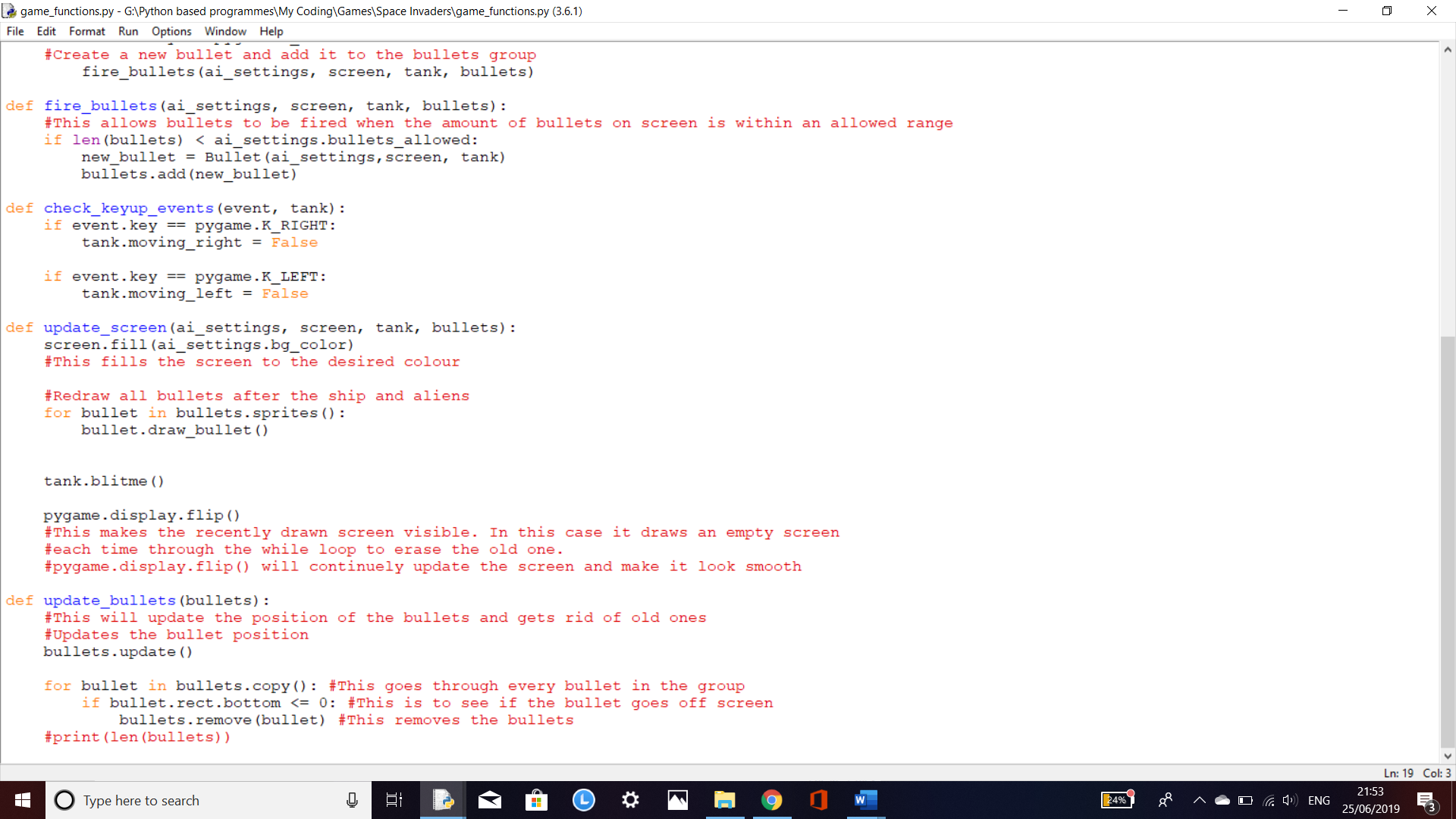
### The Code for the program so far:











This is all of the code I have gotten at this current time. I have commented on each part I felt may need some explanation. I have used classes and functions as well as pygame so far in my code. At this current time, the code heavily relies on inheritance so this means that all of the code I write must be perfectly linked and written in order for a whole part of code to work.

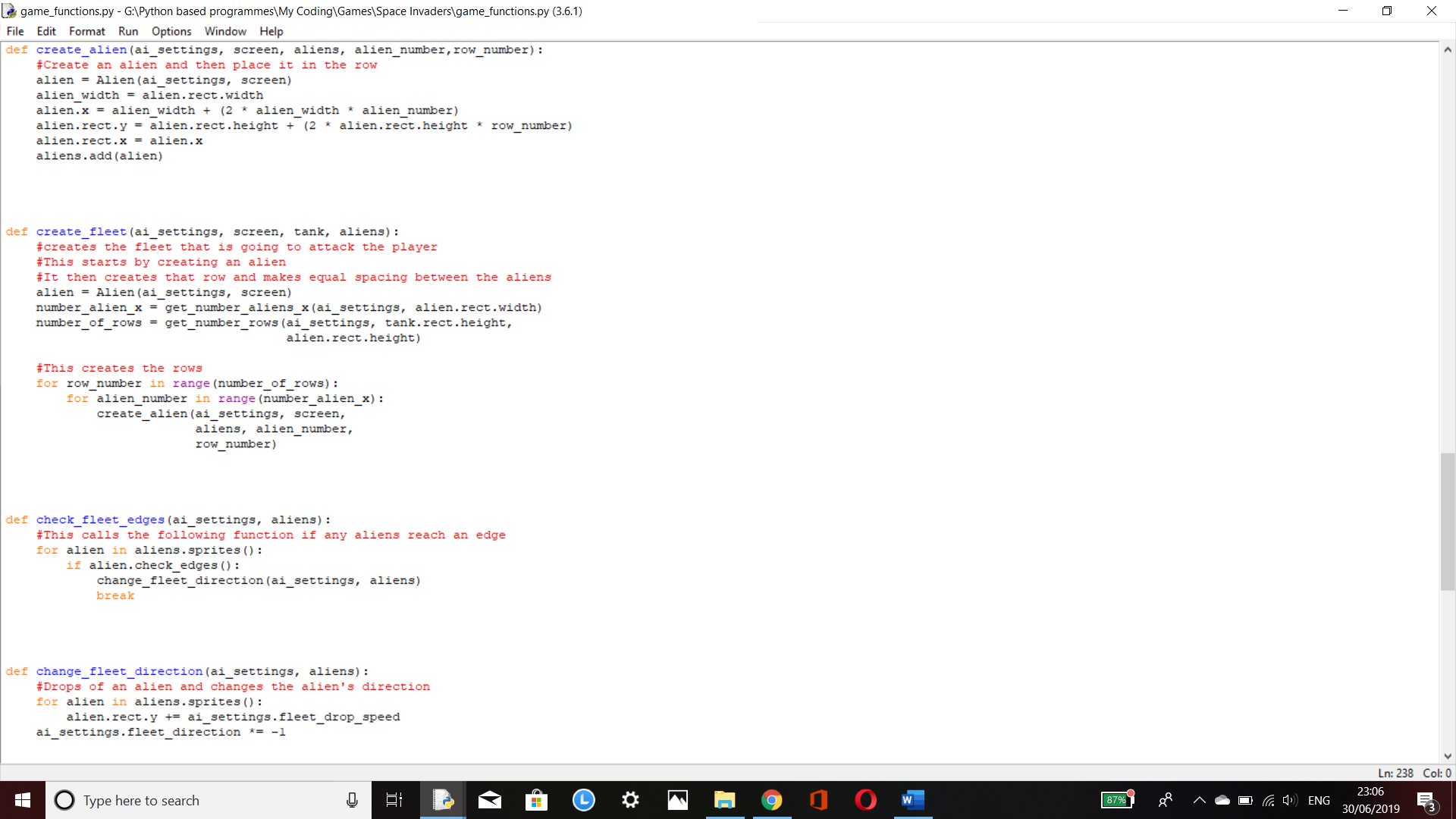
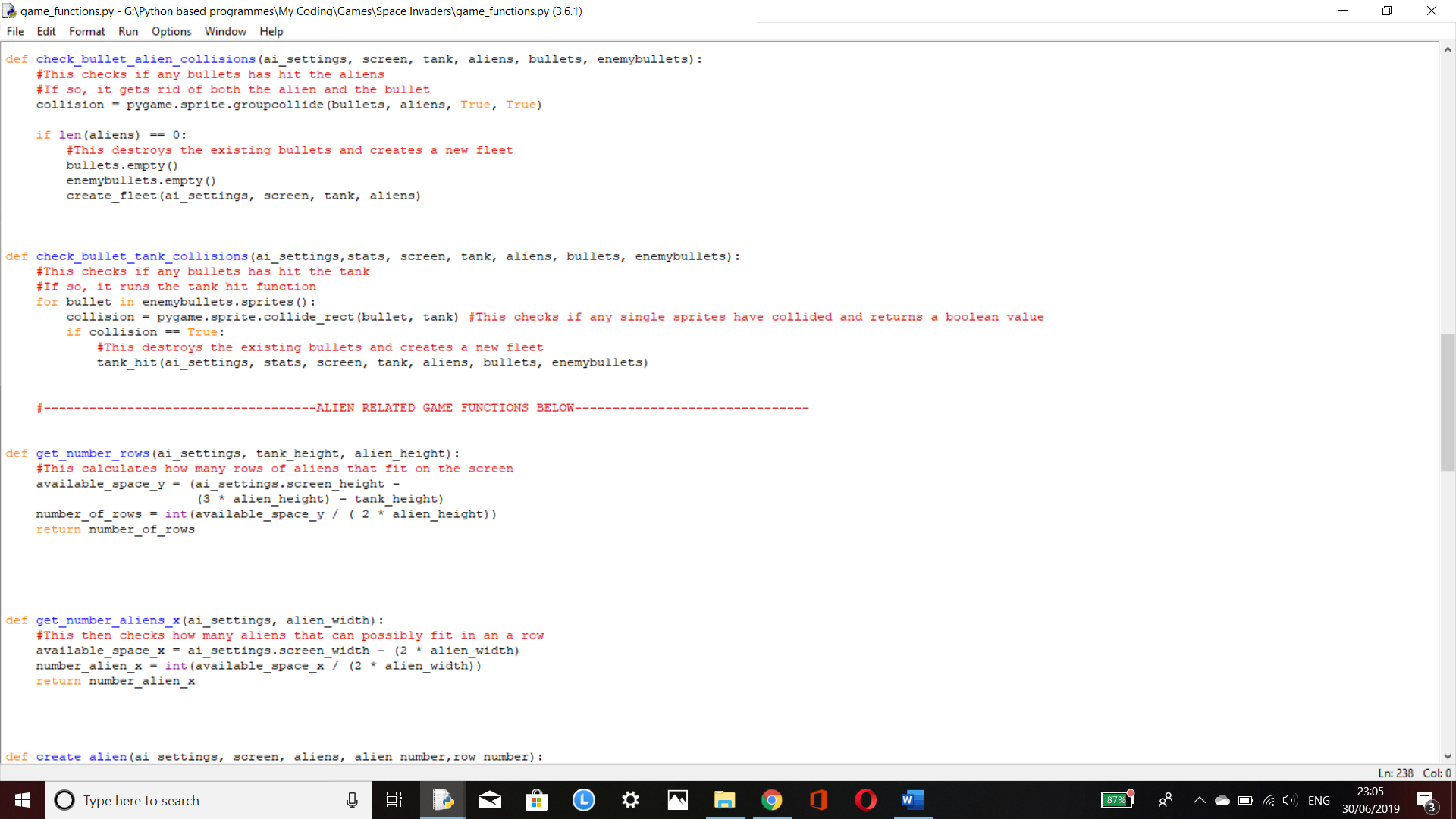
## 2nd Display of Code (30/06/2019):

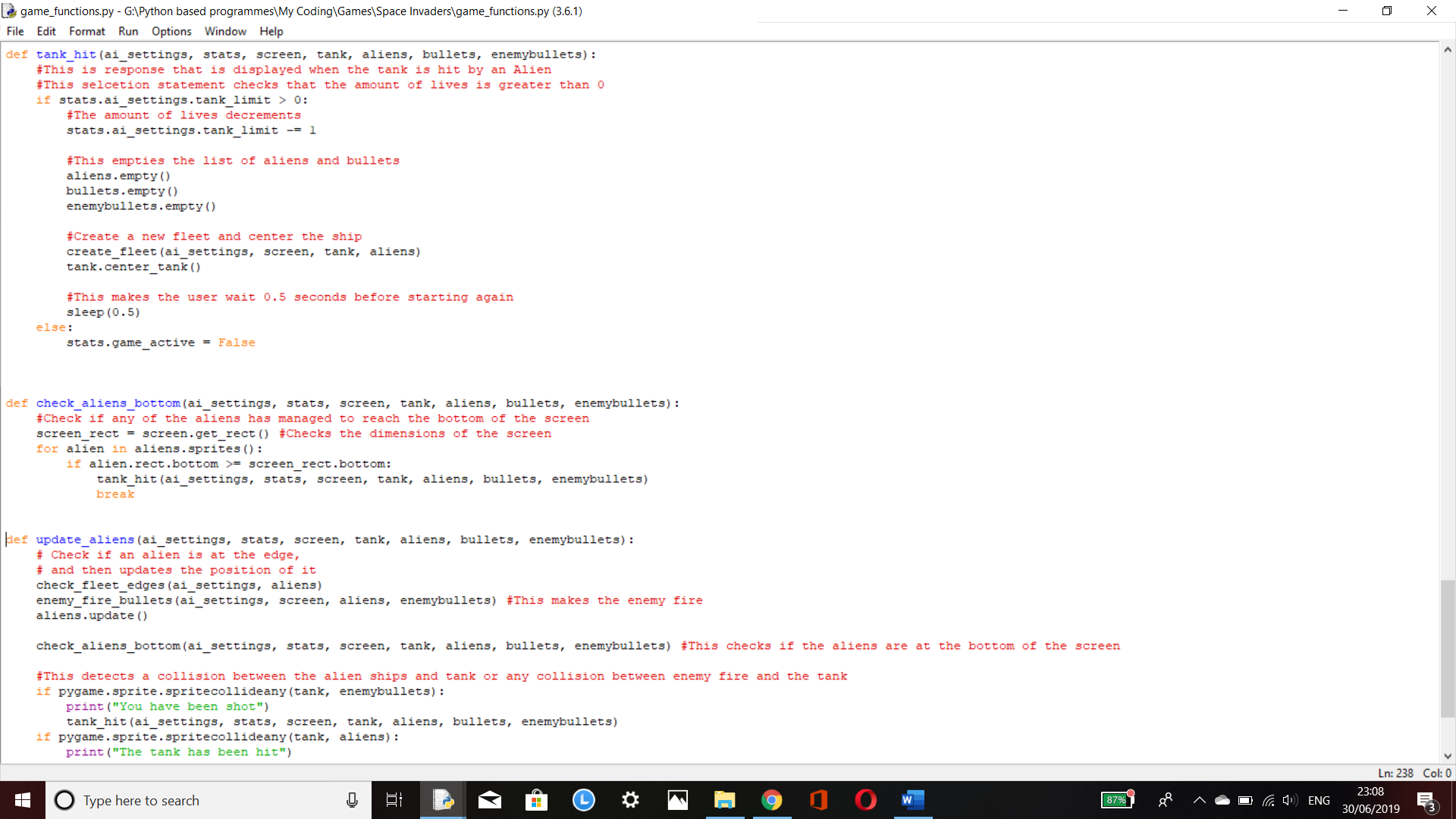
### The Alien Features:

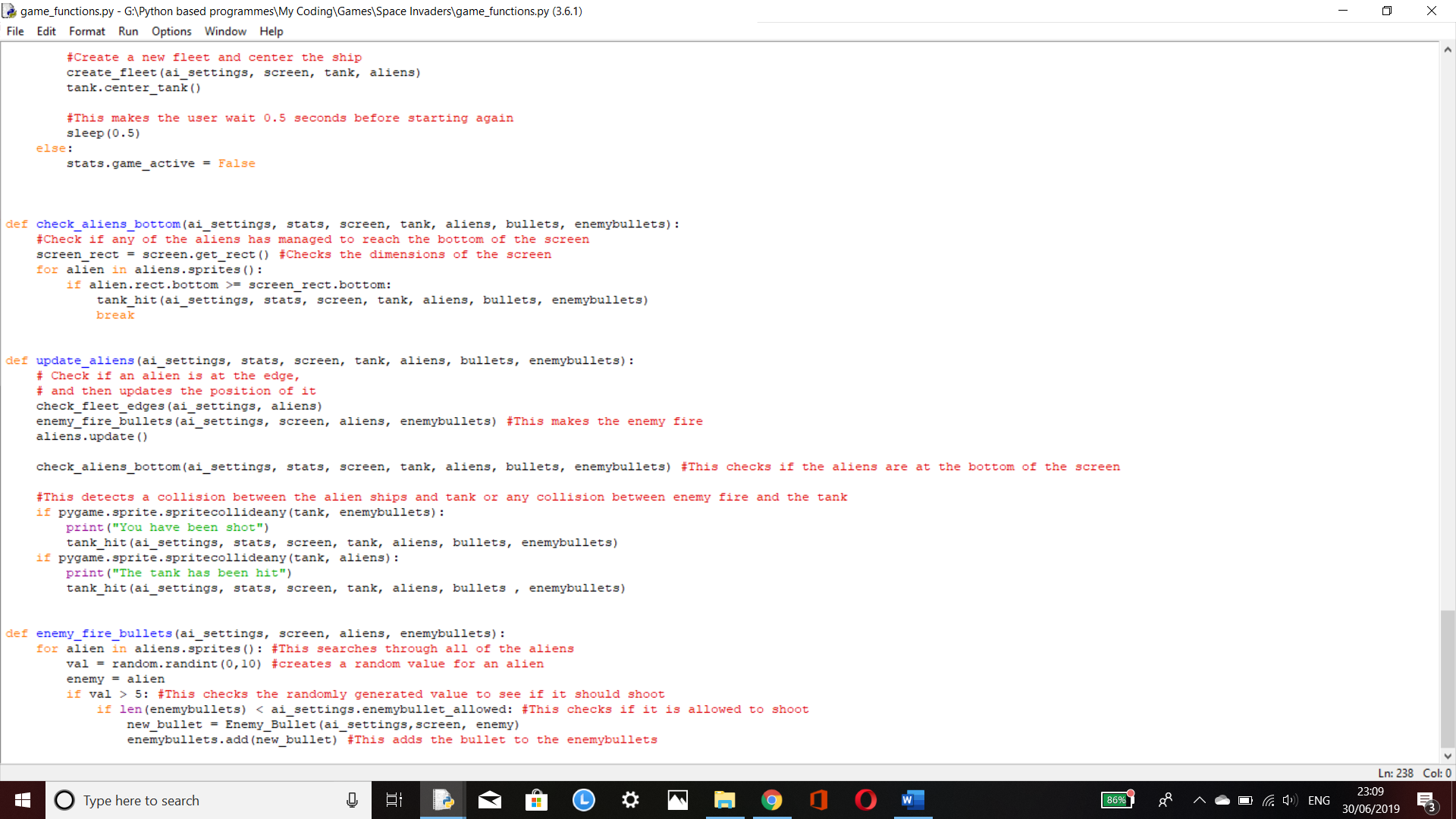
Programming these features for the aliens have been quite difficult; so far, I have done most of the features for the Aliens. Throughout the programming of this section I have tested and tried different things such as making the fleet move in a method similar to the old ‘Snake’ game. Although, I was unsuccessful with this.

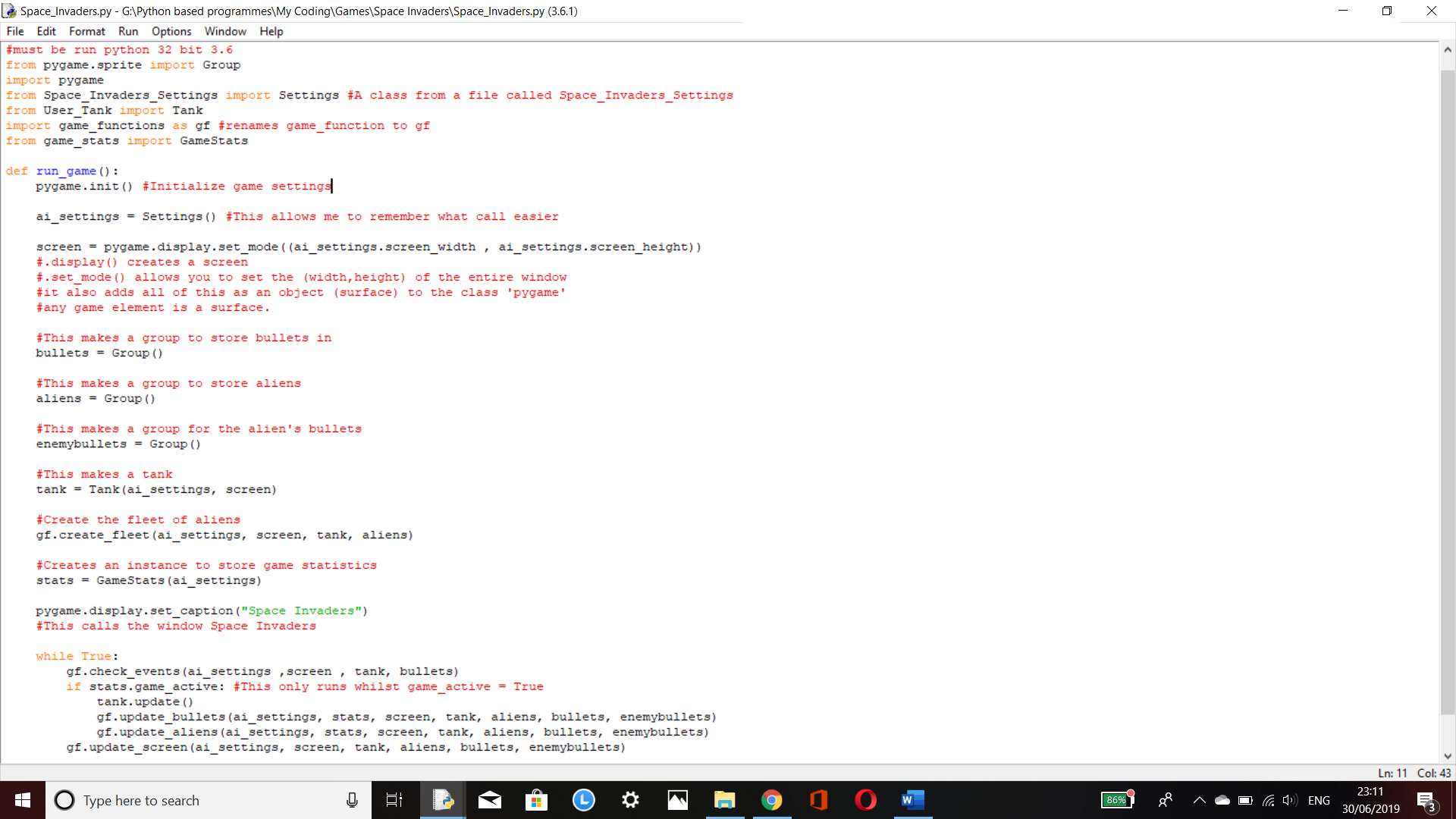
One feature that differs from the book, that I programmed by myself, is a feature that allows random aliens to fire bullets. I managed to achieve this by using code similar to the code that allow the user able to fire bullets. The thing that differs from it is that I used a random module to control which alien fires. I also made a hit detection for the bullets colliding with the player.

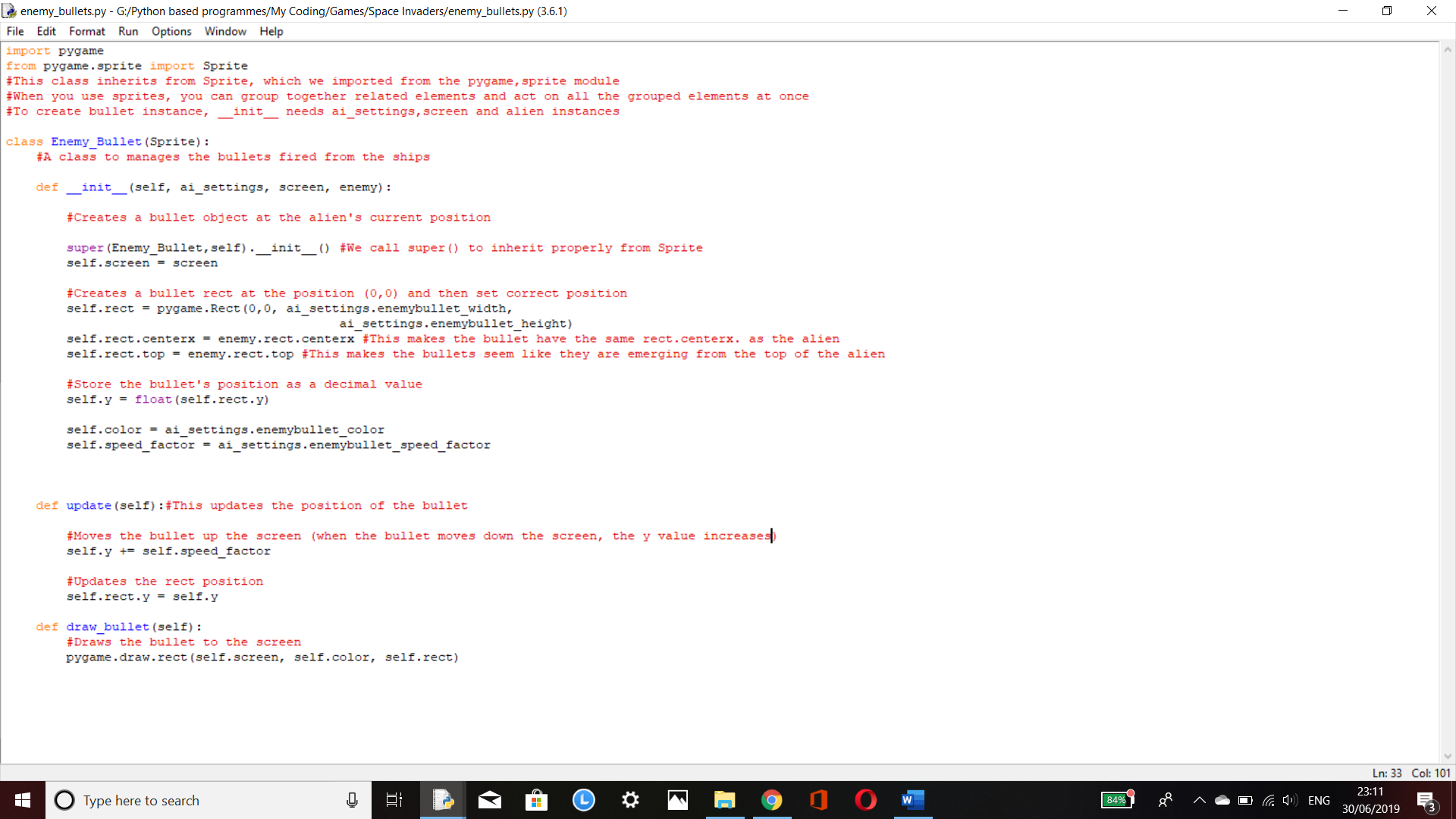
### The Code for the program so far:

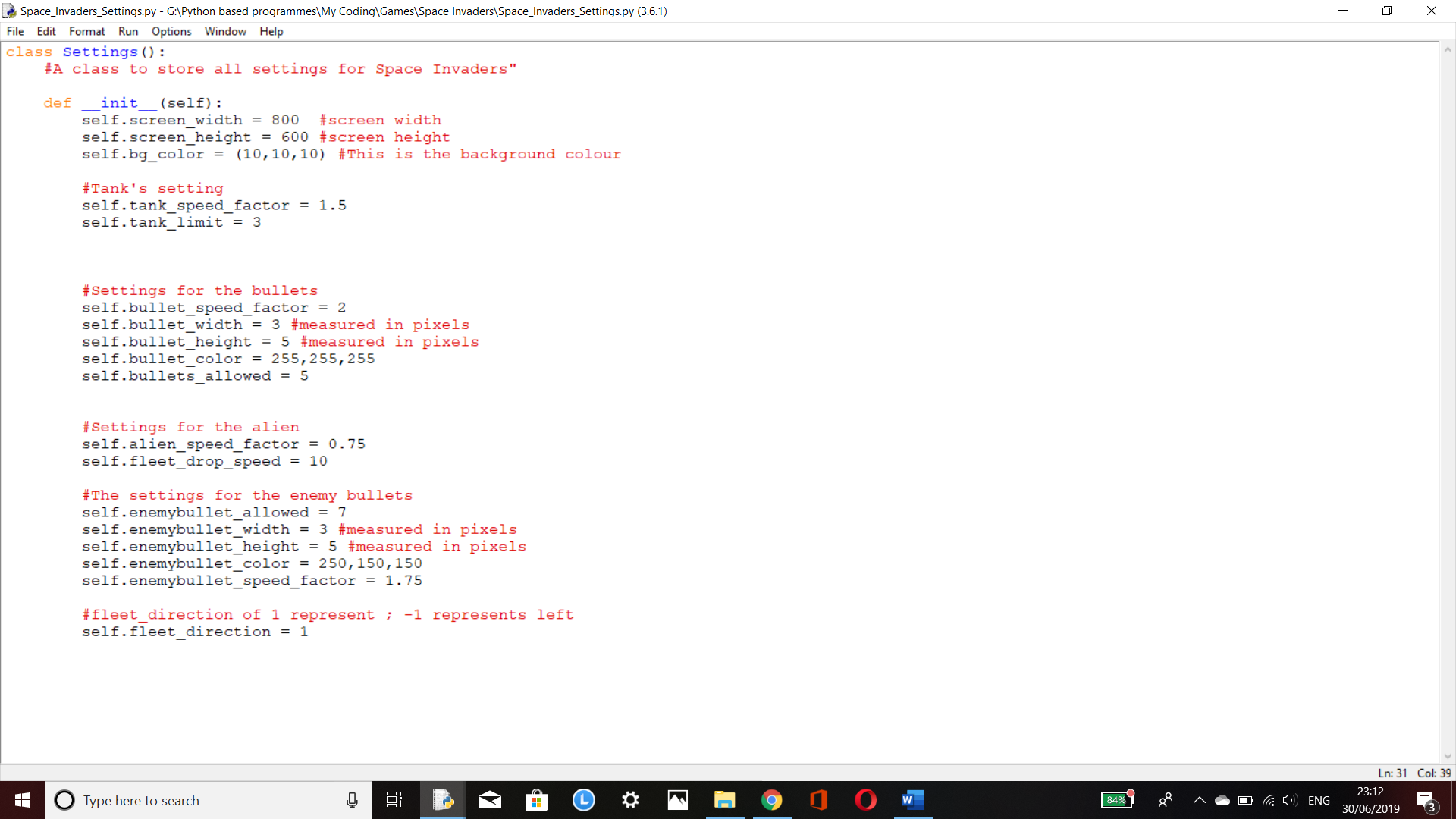


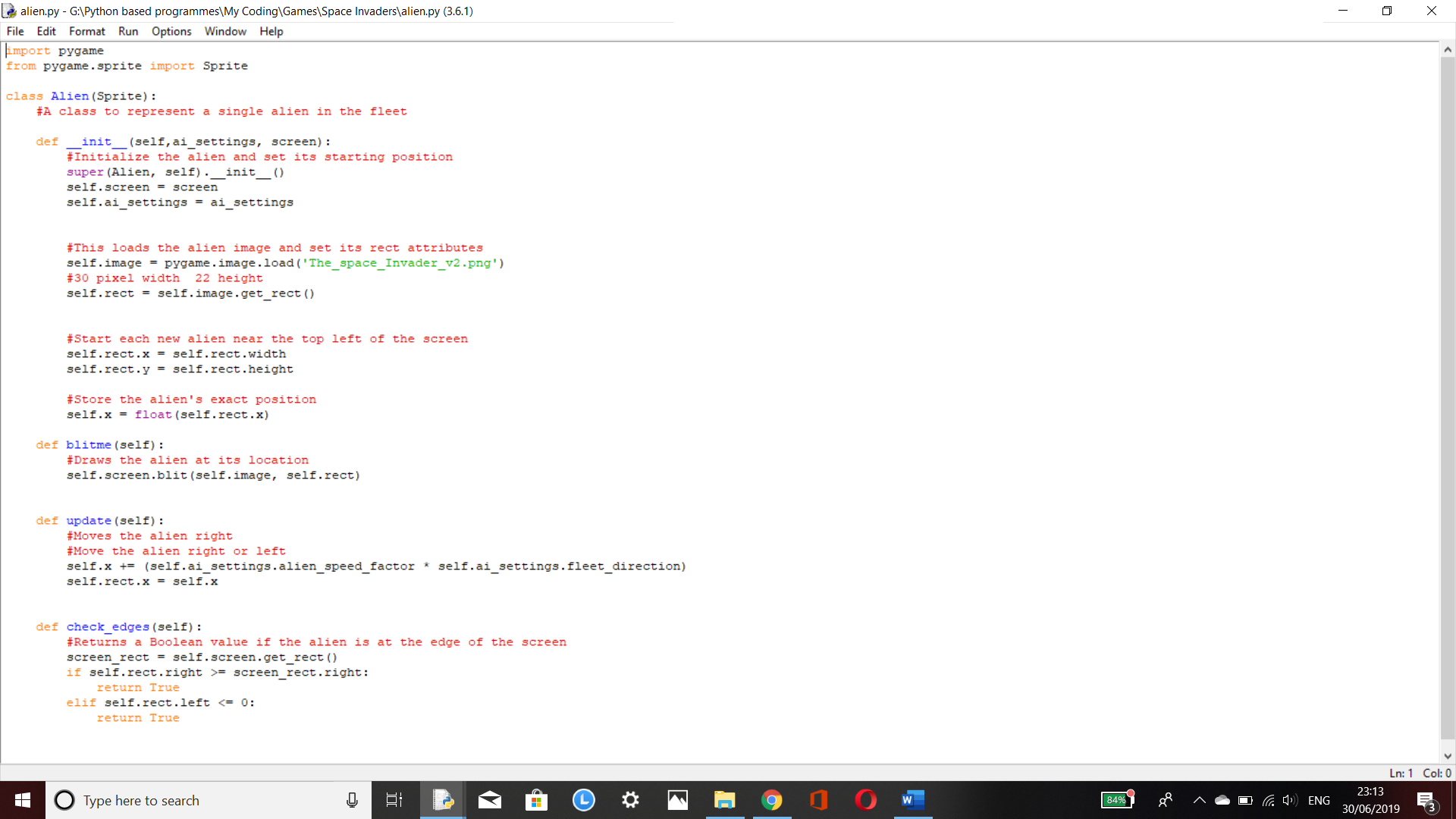


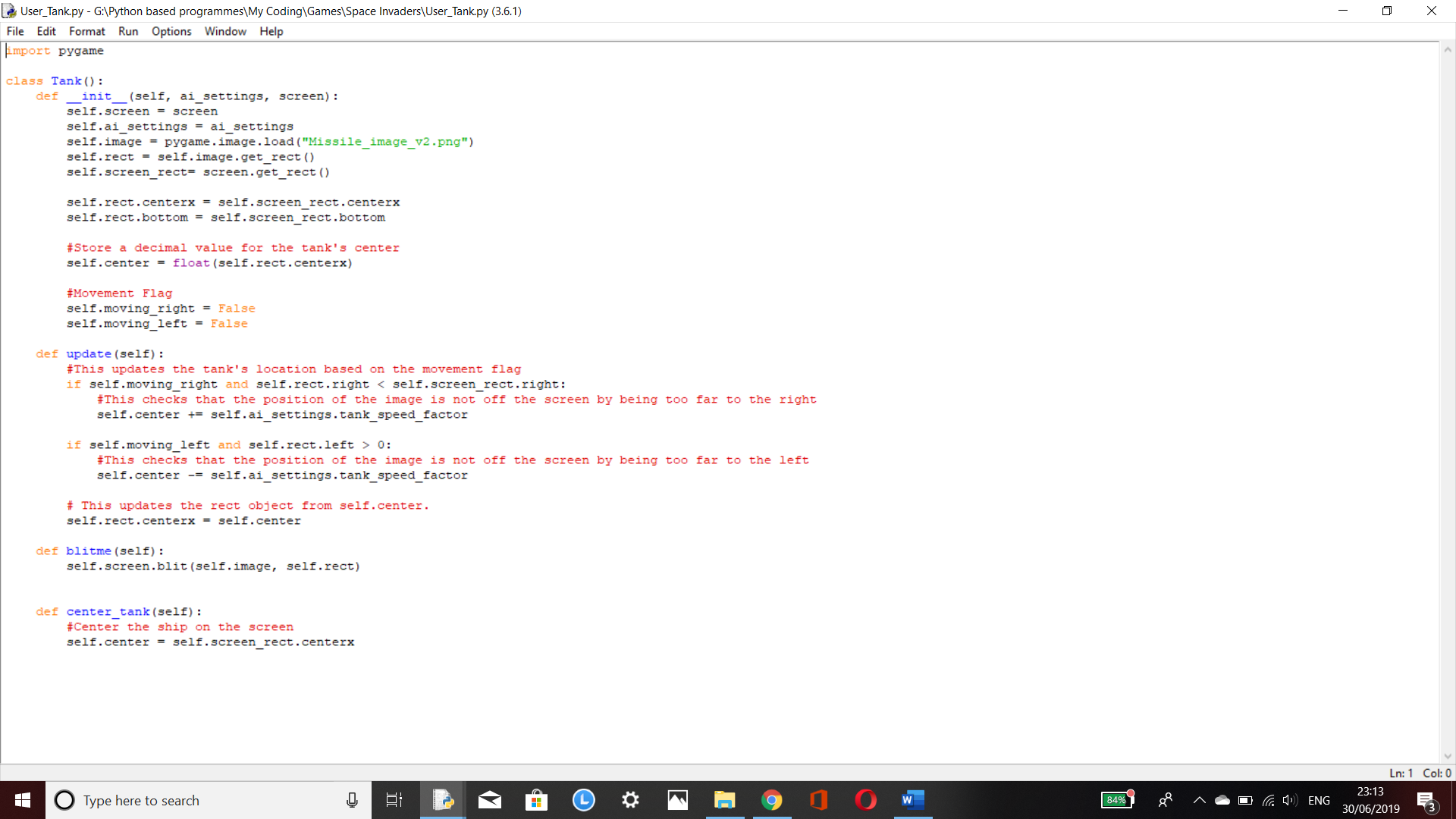


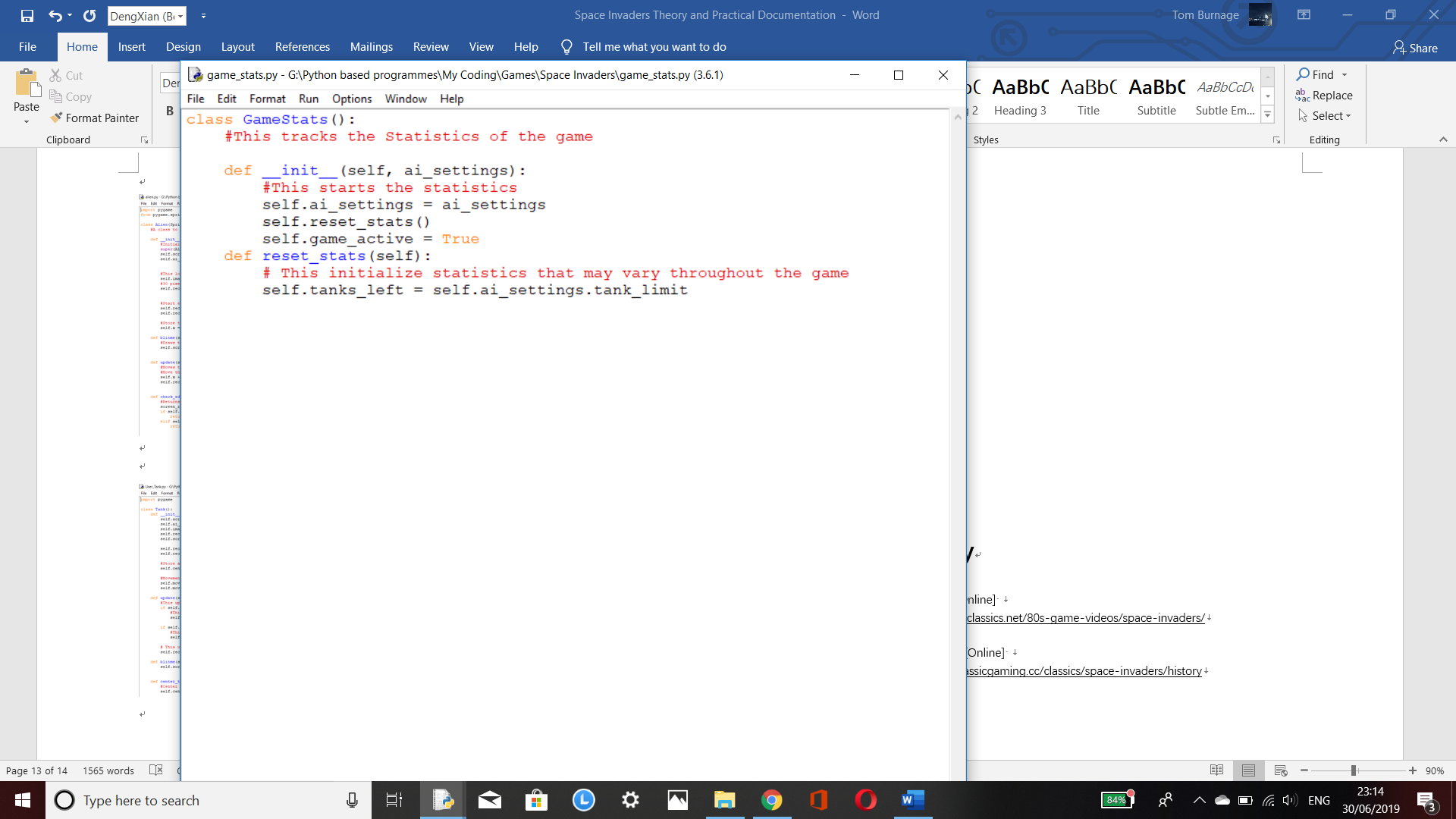


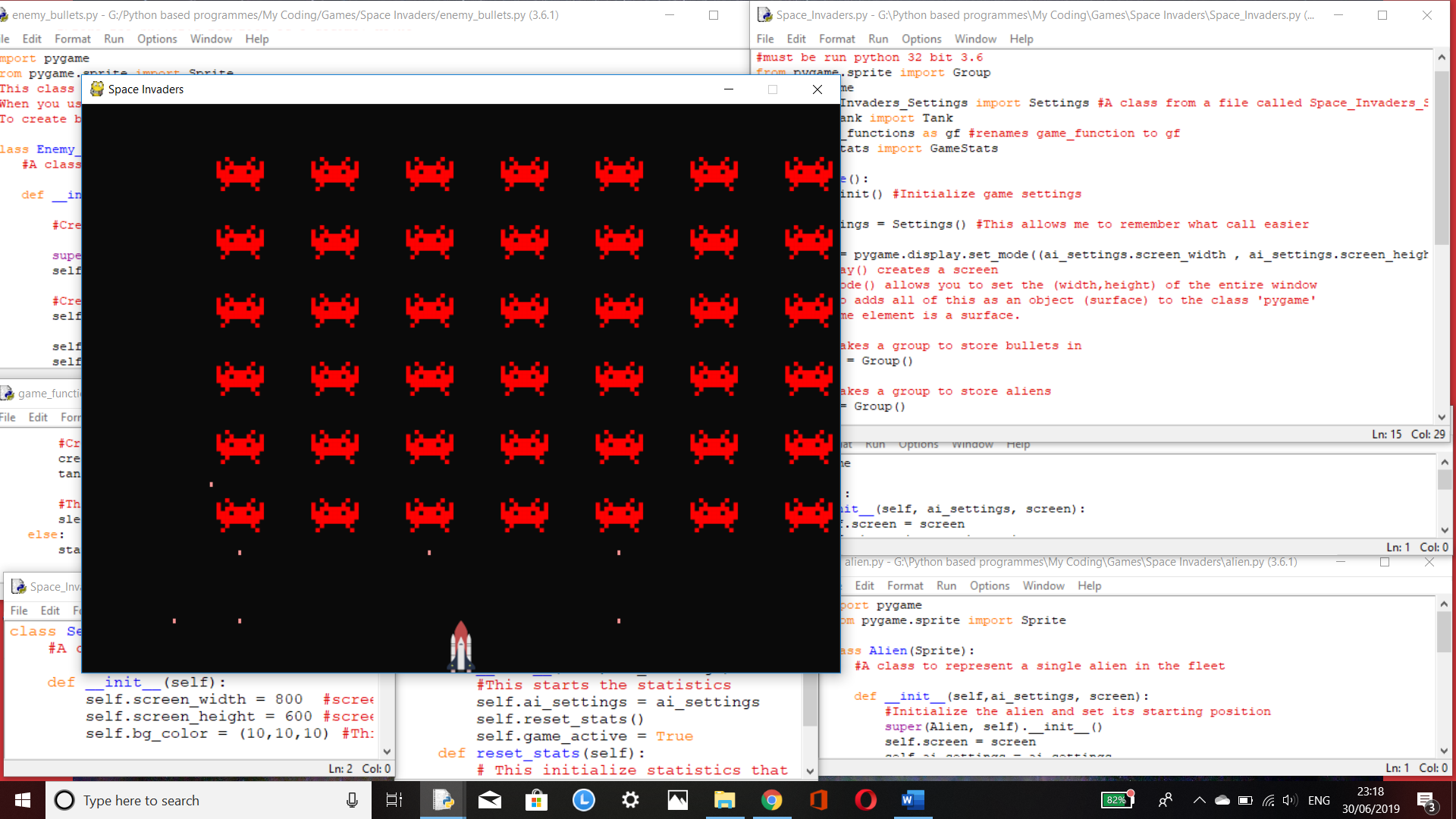










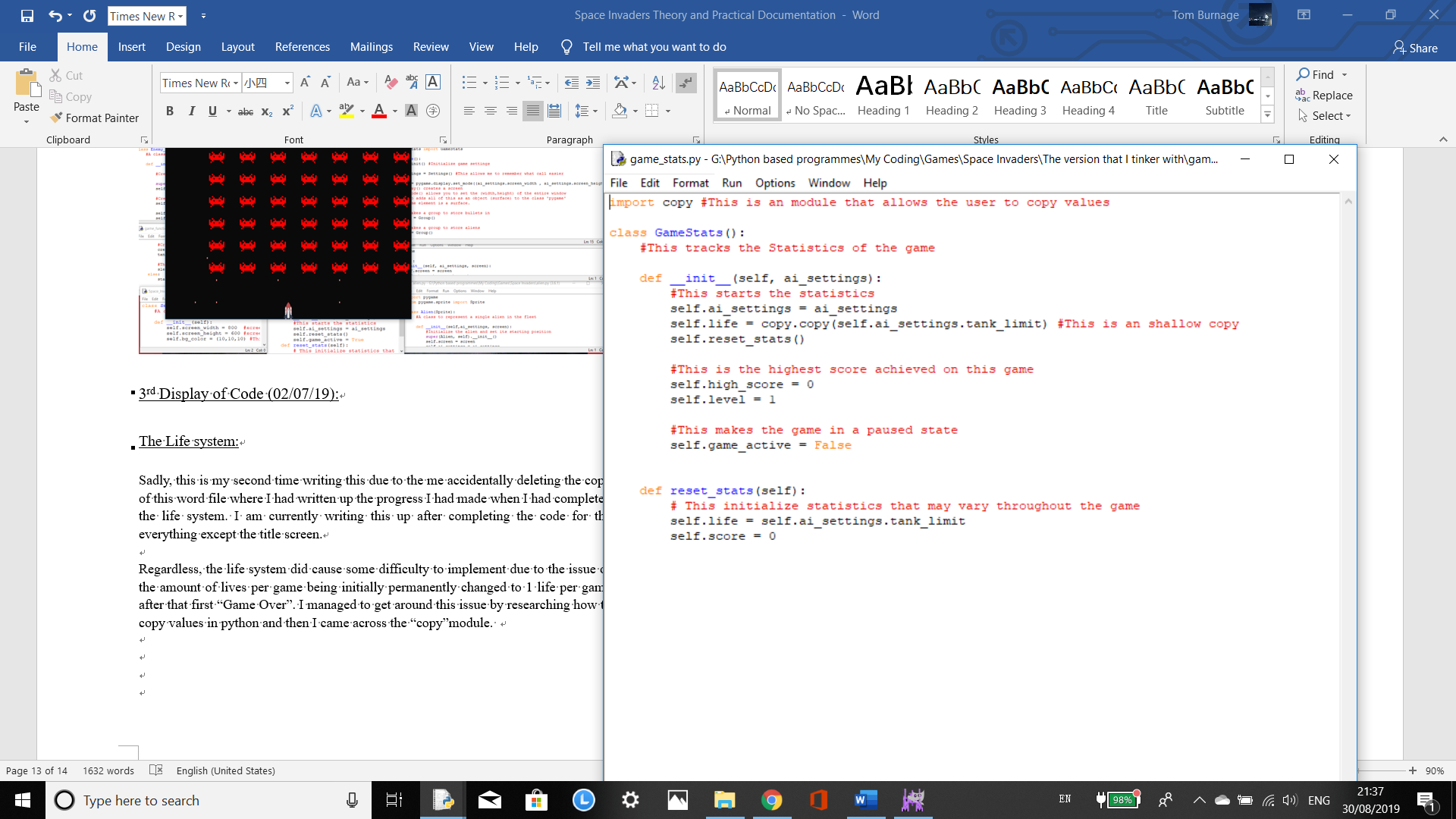


## 3rd Display of Code (02/07/19):

### The Life system:

Sadly, this is my second time writing this due to the me accidentally deleting the copy of this word file where I had written up the progress I had made when I had completed the life system. I am currently writing this up after completing the code for the everything except the title screen (30/08/2019); due to this predicament, I will not display all the screenshots of the programs of code.

Regardless, the life system did cause some difficulty to implement due to the issue of the amount of lives per game being initially permanently changed to 1 life per game after that first “Game Over”. I managed to get around this issue by researching how to copy values in python and coming across the “copy” module which allowed me to pass the amount of lives by value.



## 4th Display of Code (30/08/2019):

### The Scoring System:

The scoring system was annoying to implement due to my original plan was to store the scores achieved in an external file along with a player’s name in the format of [ score, player name] and have it updated when new data has been added to the file. I managed to use the tkinter module to create a basic window to allow the player to their name. However, I kept encountering an a few errors such as TypeError whenever I tried retrieving the information from the file (both when I tried using the json module and the basic python version of using files). When I managed to retrieve the information as well as sort it, I encountered a problem that I couldn’t solve even when I spent multiple hours trying multiple methods and researched it. This problem was that the file system wouldn’t let me write the list of multiple arrays to file.

You may be thinking that I could simply use the format of score/Player-Name but for some reason that I couldn’t quite get the sorting method for that one to work. So, I had to do a different method.

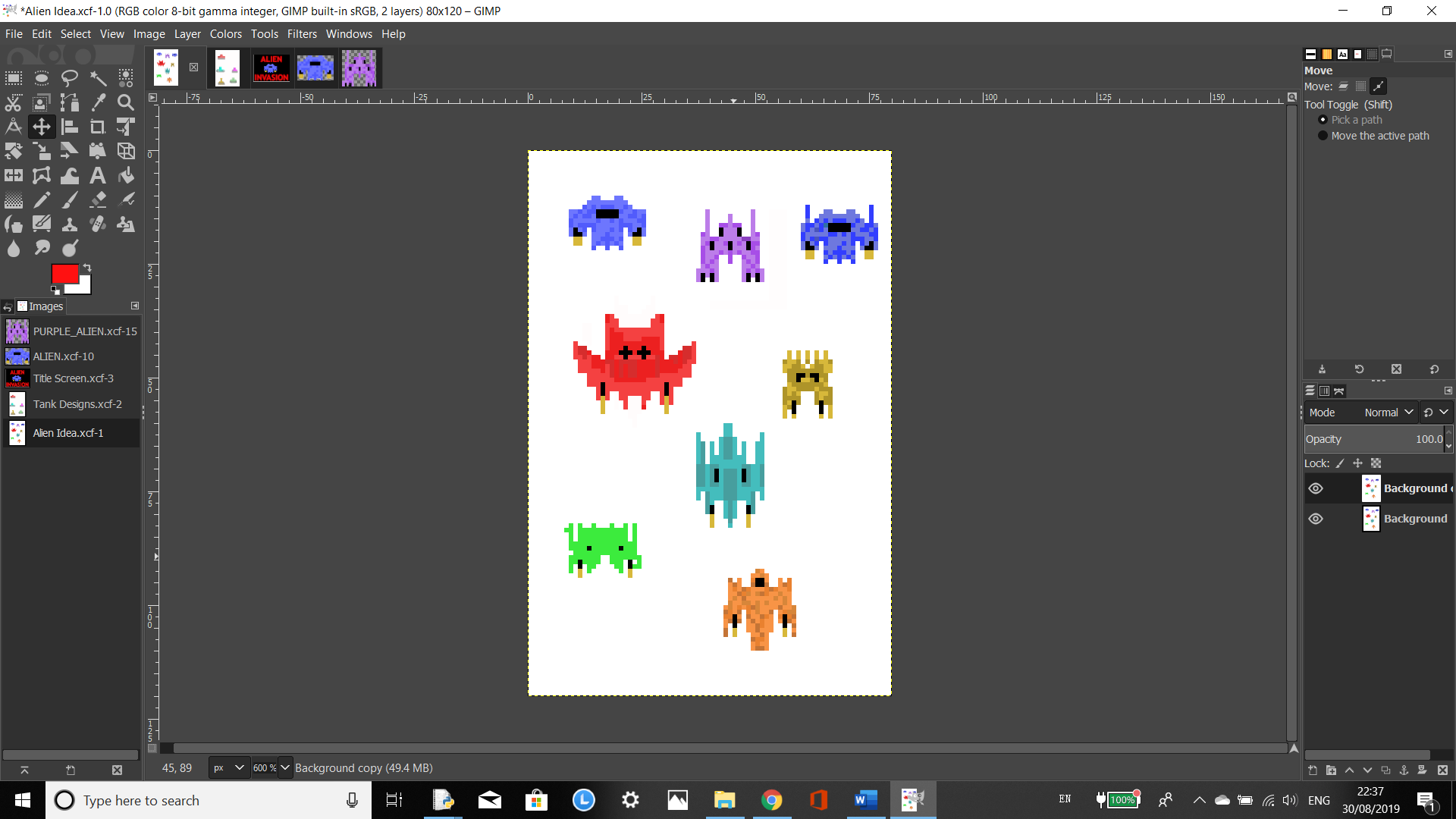
The method I resulted in using was relatively simple in comparison to my original idea but I managed to make it work. The method involved having both the high score and player’s score being stored on the same program file and have it updated when it is required. This was easy to program although the drawback is that the high score Is volatile and when the game is closed, the high score gets reset as well. The other drawback is that no player name is being recorded so people will just have to remember who achieved the highest score.

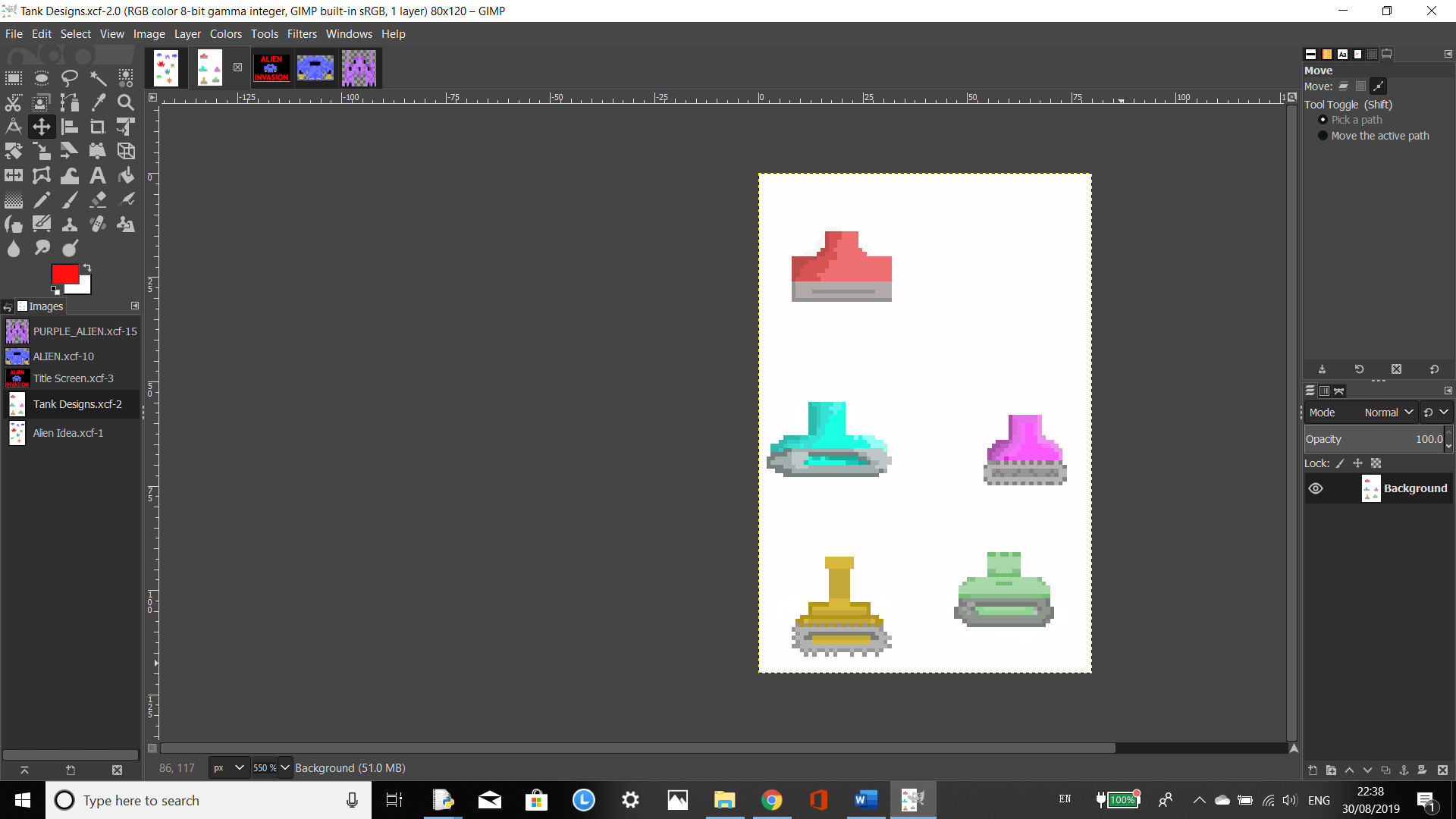
# Design:

## Making the aliens and tanks skins:

This was the fun part because I had freedom to make my own retro designs, I decided to use GIMP (it is a free program similar to photoshop) to design my aliens and tanks. My reason for doing this is to remove the possibility of crossing over any ownership issues over intellectual property. The image of the alien and rocket that I used whilst programming the game were placeholders and when I searched for copyright free images of their respected objects, they came up.

I based my aliens and tanks off of my own imagination so if any of the designs are similar to other designs online, it is purely coincidental, and it was not my intention.







Title Screen:

This was the design of the title screen and I was intending on implementing this however I couldn’t seem to get it to work.

# Overview:

I really enjoyed to doing this project due to the sense of accomplishment when I managed to use multiple programs together at once to make this game. There were a lot of problems that I encountered whilst doing this however I managed to solve most of them,

If I was to do this again, I would change the scoring system to my original design (when I figure out how to make it work). This would remove the issue of a volatile high score with no clear player who achieved it. It would also allow players to find out where they are skill-wise compared to players who had played the game before them.

The code that I used for this project was based off of the alien invasion project from the book “Python Crash Course by Eric Mattes”, however there has been several differences between my code and the one in that book such as:

-Random aliens shoot back at the player

-Different way of sending the information of the life data

-Self created Tank and Alien

I also attempted to add these features but was unsuccessful:

-Score storage with player name input

-Power ups such as oily tracks (makes the user to go faster for a set amount of time) and turret (deploys a little turret that only shoots 5 shots up every 5 seconds. Requires 4 shots to destroy.)

-Multiple alien skins shown at once whilst playing.

-Choice between the Alien Skins and Tank Skins

-Title screen

# Bibliography

N/A, N/A. *arcadeclassic.* [Online]   
Available at: https://arcadeclassics.net/80s-game-videos/space-invaders/  
[Accessed 24 June 2019].

N\A, 2019. *classicgaming.* [Online]   
Available at: http://www.classicgaming.cc/classics/space-invaders/history  
[Accessed 24 June 2019].