**American International University- Bangladesh**

**(AIUB)**

**Faculty of Engineering**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Course Name:** | | | Programming In Python | | | | | | | |
| **Semester:** | | | Spring (2021-2022) | | **Section:** | | B | | | |
| **Course Code:** | | | CSC4162 | | **Faculty:** | | Akinul Islam Jony | | | |
| **Submission Date:** | | | 13-3-2022 | | Comment: | | | | | |
| **Group Name:** |  | |  | | | |
| **No** | **Name** | | | | **ID** | | | **Program** |
| 1 | Haque Md Imdadul | | | | 19-40619-1 | | | BSc[CSE] |
| 2 | Sujoy Chandra Das | | | | 19-40100-1 | | | BSc[CSE] |
| 3 | MD TOUHIDUZZAMAN SAGOR | | | | 19-39951-1 | | | BSc[CSE] |
| 4 | Misu MD Murad Alahi | | | | 19-40467-1 | | | BSc[CSE] |

# 1.0 Problem statement

Space Invaders is an old Atari game and one of the first games that were made 40 years ago. Even though it can sound like “Back to the future” where it seems like no one is playing it nowadays but with the knowledge of Artificial Intelligence, we can bring back this game for the older people to reminisce their childhood game and make it even more interesting with the graphic to draw younger generations’ interest with this game.

The main goal to implement the system is we need to think about how the winning strategies are set once the goal is reached. For example, once the player has shot the aliens in the game, they will get 1 point. Getting a point and winning will give a satisfying feeling to the player as a reward. This feeling will encourage the user to constantly play it as a hobby.

Besides that, Mohan (2015) mentioned that the algorithm approach could eventually be applied to any complex and multidimensional task requiring a series of decisions. So, for a player to control the direction of the spaceship using the keystrokes as the instruction to the agent (spaceship), many errors and trials need to be done to improve the system and the winning strategies. Therefore, Games require more exploration, planning, and complex route-finding.

Almost digital games had been created just focusing on the male player only and some of them not suitable for certain ages. But, this game can be play both gender and also all ages. According to William et al. (2009), their claim that digital games not only being played by males but it also being played by female.

# 2.0 Motivation

This project is meant to bring back the entertainment of the Atari Game. Nowadays most games are made with a good graphic with the help of technologies. Since the graphic of Space Invader back then was not as good as today's game, we want it to make more interesting by using a 2D graphic. By this, it can give the chance to the younger generation to play this game which is by implementing back the game with a better graphic. Space Invaders can bring a feeling of nostalgia to our parents by reminiscing the fun of playing this game.

Space Invaders also being recognized by Artificial Intelligent where it let younger programmer find what method is used to create these games and find out the mechanism. It let the young programmer create this model and upgrade it into something news. According to Morrison (2003), he claims that 30 years ago, the digital game never is concern by people at all but it grows out important for every sector.

Most import thing about this game, it let younger programmer create their world using all the skills they had and also imagination. According to Morrison (2003), he claims that video game lets users participant on the story setting while enjoying the developer ‘dream world’. According to Cheok et al. (2018), their claims this game may able to improve the user gaming skills and their health through the bonus round and also as wellbeing. Sherman et al. (2004), claim that playing a video game can improving eye-hand coordination, reaction time, and also spatial sense.

# 3.0 Related work

There is a technique called Reinforcement Learning in the knowledge of Artificial Intelligence which allows machine learning to learn itself. Based on Gupta (2019), when we apply this to Space Invaders, the agent extracts observations from the environment which will help it decide the right action to take. The only way the machine learning can ‘observe’ the environment is the use of a Neural Network.

In the beginning, the creator of Space Invader (Tomohiro Nishikado, Japanese video game developer) using around 512×256 pixels to create all images for this game. According to Clarke and Mitchell (2007), this system almost similar to arcade computer games. “Over the years, I've seen how Space Invaders helped grow the video-game industry and inspire younger designers,” said Nishikado during his interview with The New Yorker magazine.

Besides, Nishikado also creates it using Intel 8080 central processors (second 8-bit microprocessor) to display the images he wanted to created. According to a magazine known as Gamer Informer (n.d.), their claims that microcomputers not that strong to handle a heavy and complex task in designing and programming where it let Nishikado-san using his custom hardware and tools to create this game.

This system involves Pygame where all the modules to make the game interesting are available in Pygame. So, to store the graphic and sound effect on the system, Pygame is needed. The final line is optional and puts a subset of the most frequently used Pygame functions and constants into the global namespace (Rutledge, 2004). To attempt the initialization of the game, Pygame has the term pygame.init() that can be applied to the system. All the Pygame modules do not need to be initialized, but this command automatically initializes the ones that do.

All the images, sound effects, and background need to be made with functions in preparation to load. Functions also contribute to the game loop by giving instructions to the agent to react with the environment. For example, Spaceship as an agent while the environment as the enemy. This is to teach how machine learning works with the decision by instructions.

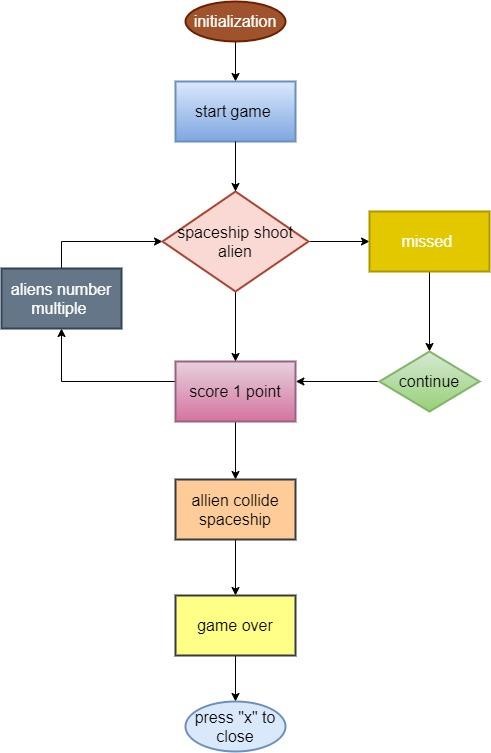
According to Neumeyer (2015), his stated that James Newman (English singersongwriter) give an opinion on why video games had been ignored as serious research of study. One of the reasons is video games had been considering as children medium and also quite childish and cartoonish. After the creation of Space Invader, the musical style in video games had been inquiry some research.

For example, Neumeyer (2015) claim that the tune been used for “marching alien feet” consider as nondiegetic sound. He explained that this tune important because it let the player imagine that they in the game setting story directly or know as diegetic sound. They're also a few pieces being used in this game where it can explain the situation properly such as

“Romance d’Amour” (Spanish guitar piece) being used to give a signal for the battle begin, and also Beethoven’s *Fur Elise* if the player completing the final level of the game.

**4.0 Methodology**

Flowchart



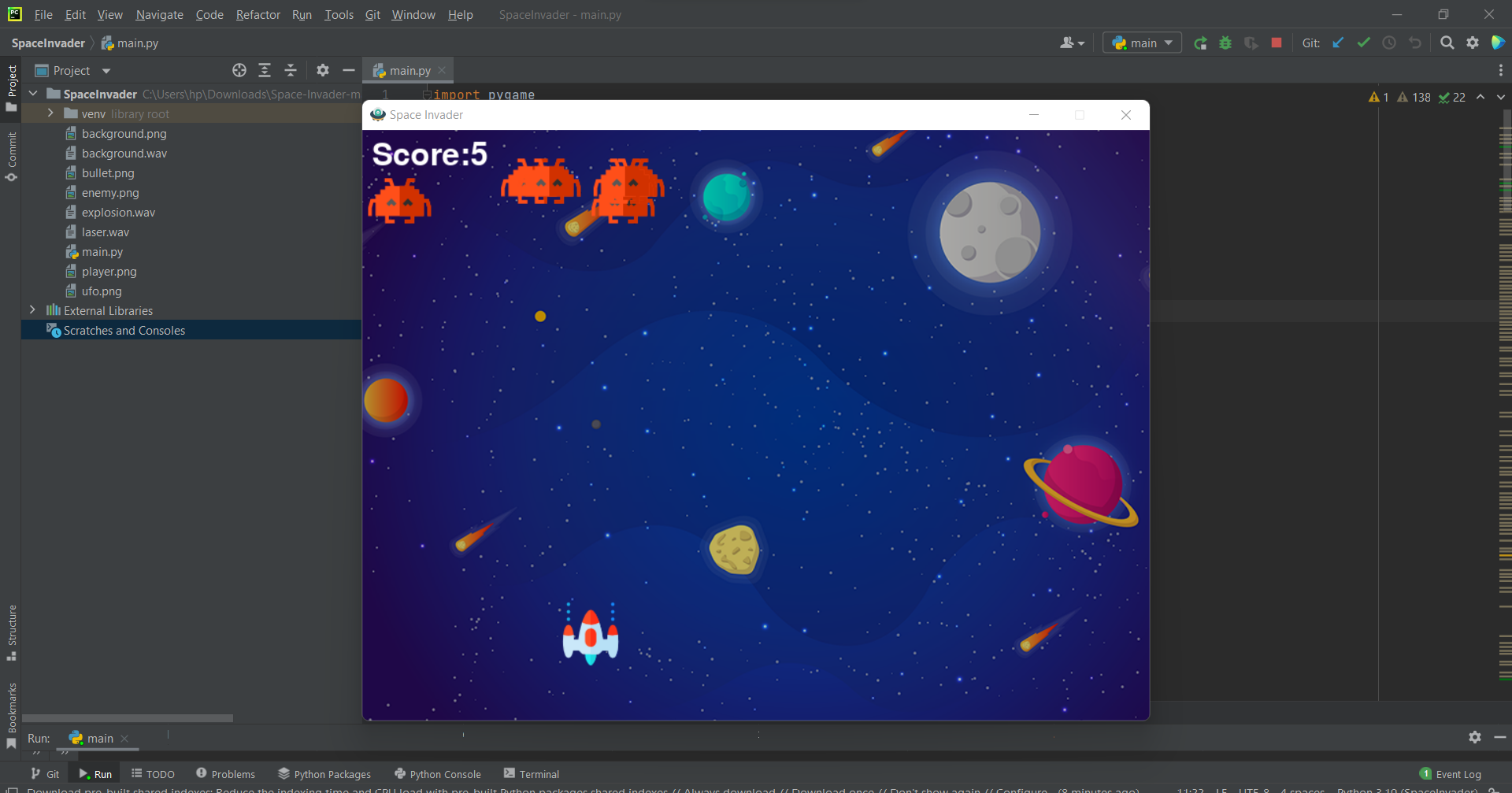
# 5.0 Plan of Action

1. Python and Pycharm installation
2. Creating the Game Window
3. Modify the Title, Logo and Background Color
4. Attach Images into Space Invader Game
5. Motion Mechanics in Game Development
6. Keyboard Insert Controls & Key Pressed Event
7. Attach Boundaries to Our Game
8. Design the Enemy
9. Motion Mechanics of the Enemy Space Invader
10. Attach a Background Image
11. Generate Bullets for Shooting
12. Shooting Multiple Bullets at Space Invaders
13. Collision Detection
14. Generating Multiple Enemies
15. Attach Text and Displaying Score
16. Attach Sounds and Background Music
17. Game Over

**6.0 Implementation and Code**

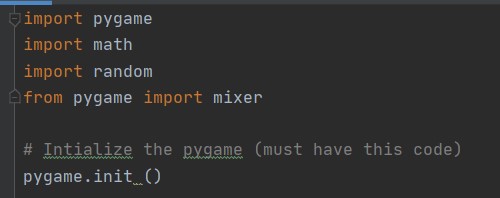
import pygame  
import random  
import math  
from pygame import mixer  
  
#Intialize the pygame  
pygame.init()  
dis\_width=800  
dis\_height=600  
#creating a screen  
screen = pygame.display.set\_mode((dis\_width,dis\_height))  
  
#Backgroud  
bg=pygame.image.load('background.png')  
  
#backgound sound  
mixer.music.load('background.wav')  
mixer.music.play(-1)  
  
#Title and Icone  
pygame.display.set\_caption('Space Invader')  
icone =pygame.image.load('ufo.png')  
pygame.display.set\_icon(icone)  
  
#Player  
playerimg =pygame.image.load('player.png')  
playerx=370  
playery=480  
playerx\_cng=0  
  
enemyimg =[]  
enemyX=[]  
enemyY=[]  
enemyX\_cng=[]  
enemyY\_cng=[]  
num\_of\_enemies = 6  
for i in range(num\_of\_enemies):  
 enemyimg.append(pygame.image.load('enemy.png'))  
 enemyX.append(random.randint(0,734))  
 enemyY.append(random.randint(50,150))  
 enemyX\_cng.append(4)  
 enemyY\_cng.append(20)  
  
#Bullte  
#you cant see the screen  
#fire - the bullte is moving  
bullimg =pygame.image.load('bullet.png')  
bullX=0  
bullY=480  
bullX\_cng=0  
bullY\_cng=10  
bull\_state="ready"  
#Score  
score=0  
font = pygame.font.Font('freesansbold.ttf',32)  
textX=10  
textY=10  
#Game over text  
game\_font=pygame.font.Font('freesansbold.ttf',364)  
def game\_over\_text():  
 game\_text = font.render("G A M E O V E R",True,(255,255,255))  
 screen.blit(game\_text, (200,250))  
def show\_score(x, y):  
 score\_t=font.render("Score:"+ str(score),True,(255,255,255))  
 screen.blit(score\_t, (x, y))  
def player(x,y):  
 screen.blit(playerimg,(x,y))  
  
def enemy(x,y,i):  
 screen.blit(enemyimg[i],(x,y))  
def fire\_bull(x,y):  
 global bull\_state  
 bull\_state="fire"  
 screen.blit(bullimg,(x+16,y+10)) #(x+16,y+10) is use to set the bullte in the middel of ship  
  
def isColl(enemyX,enemyY,bullX,bullY):  
 distance = math.sqrt(math.pow(enemyX - bullX, 2) + (math.pow(enemyY - bullY, 2)))  
 if distance<27:  
 return True  
 else:  
 return False  
  
  
#Game Loop  
running =True  
while running:  
  
 screen.fill((150, 0, 0))  
 #backgorudn img  
 screen.blit(bg,(0,0))  
  
 for event in pygame.event.get():  
 if event.type==pygame.QUIT:  
 running=False  
  
 #if key stroke is press check  
 if event.type==pygame.KEYDOWN:  
  
 if event.key==pygame.K\_LEFT:  
 playerx\_cng =-5  
 if event.key==pygame.K\_RIGHT:  
 playerx\_cng = 5  
 if event.key==pygame.K\_SPACE:  
 bull\_sound=mixer.Sound('laser.wav')  
 bull\_sound.play()  
 #if bull\_state is "ready":  
 bullX=playerx  
 fire\_bull(bullX,bullY)  
 if event.type== pygame.KEYUP:  
 if event.key==pygame.K\_LEFT or event.key==pygame.K\_RIGHT:  
 playerx\_cng = 0  
#checking for boundaries  
 playerx+=playerx\_cng  
 if playerx<=0:  
 playerx=0  
 elif playerx >=735:  
 playerx=735  
  
#Enemy Movement  
 for i in range(num\_of\_enemies):  
 #Game over  
 if enemyY[i]>200:  
 for j in i in range(num\_of\_enemies):  
 enemyY[j]=200  
 game\_over\_text()  
 break  
 enemyX[i]+= enemyX\_cng[i]  
 if enemyX[i]<= 0:  
 enemyX\_cng[i] =4  
 enemyY[i]=enemyY\_cng[i]  
 elif enemyX[i] >= 735:  
 enemyX\_cng[i] =-4  
 enemyY[i]+= enemyY\_cng[i]  
 # Collution  
 coll = isColl(enemyX[i], enemyY[i], bullX, bullY)  
 if coll:  
 exp\_sound= mixer.Sound('explosion.wav')  
 exp\_sound.play()  
 bullY = 480  
 bull\_state = "ready"  
 score += 1  
  
 enemyX[i] = random.randint(0, 800)  
 enemyY[i] = random.randint(50, 150)  
 enemy(enemyX[i],enemyY[i],i)  
  
 #Bullet movement  
 if bullY<=0:  
 bullY=480  
 bull\_state="ready"  
 if bull\_state is "fire":  
 fire\_bull(bullX,bullY)  
 bullY-=bullY\_cng  
  
 player(playerx,playery)  
 show\_score(textX,textY)  
 pygame.display.update()

## Figure 1: Output



**7.0 Findings**

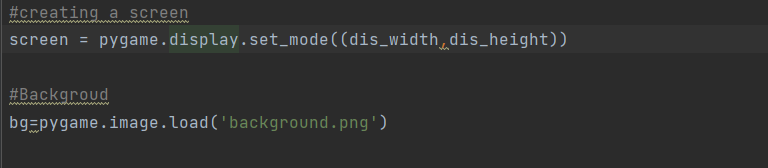
# i) Pygame



## Figure 2: Import package

Before started coding, the user who wants to create a game, they need to install a pygame package in PyCharm. To install it, users need to click File, go to Setting, find your project name, click project interpreter, go to install button which is +, and the type pygame and install that package. According to Tagliaferri (2017), she claims that this package is an opensource module for this Python language. This module can help programmer to make games and other multimedia applications.

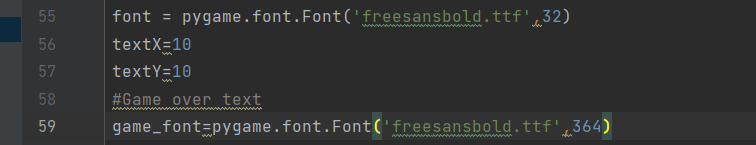
By using this module, the programmer can take over the logic and graphics without worrying about other stuff that may require a task that involves video and audio. Besides this package also can be run across any platform and operating system such as Windows, Mac, Linux, etc. To activate this package, the programmer needs to write pygame. init () because by this code it will initialize this package. This code will automatically start up all the modules in this package that require to initialized such as pygame.font.Font (), pygame.image.load (), pygame.display.set\_mode (), and others.



## Figure 3: Code for background

There are several ways to write a code for coding. For example, pygame.image.load

(‘file\_name.png’), or pygame.transform.scale(pygame.image.load (‘file\_name.png’), (width, height)). By using only pygame.image.load (‘file\_name.png’), the programmer needs to resize the images manually if the size to small than the size of the screen (width and height. So by using this code, the coding becomes more dynamic. According to pygame documentation (n.d.), it stated that this code's main function is to resize any type of images into something new in resolution aspects.



If the programmer using pygame.font.SysFont (), the programmer needs to write down pygame.font.init () to initialize the module and use it but if the programmer uses pygame.font.Font (), there not need to initialize it because it already activated from pygame. init (). According to pygame documentation (n.d.), it stated that this module allows rendering TrueType fonts into a new surface object while accepts any UCS-2 character (‘u0001’ to’Uffff’). By using pygame.font.Font (), the module itself will create Font object while the programmer can use fonts been provide in the system by using this code, pygame.font.SysFont ().

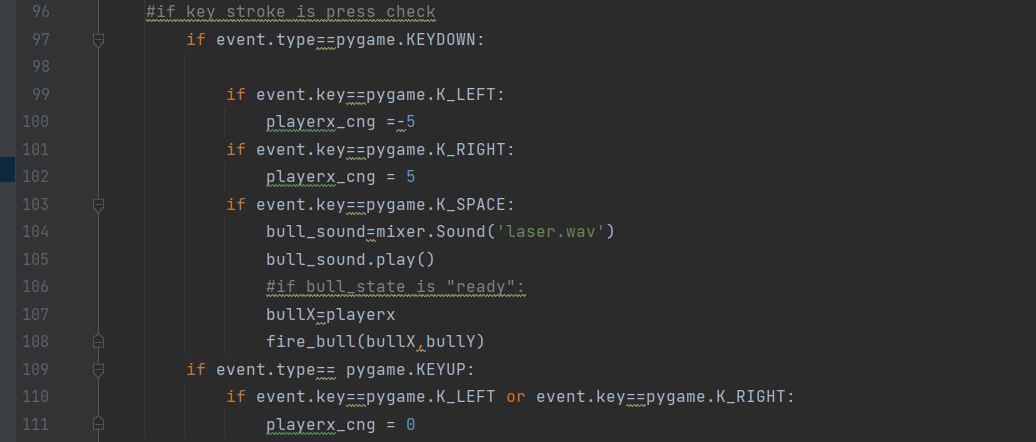
# ii) Multiple enemies

|  |  |
| --- | --- |
|  |  |
|  |  |

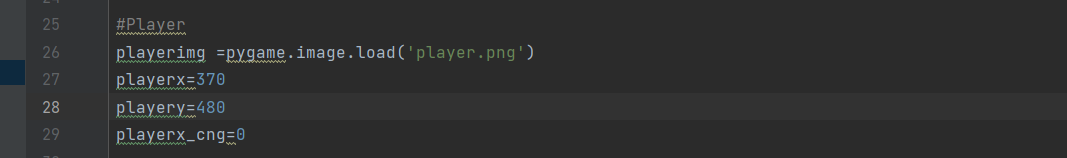
## Figure 6: Comparison

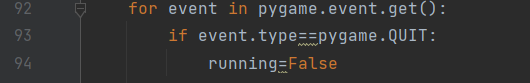
The programmer also able to manipulate the number of enemies in the game. The number of enemies becomes permanent because when players successful to shoot down one enemy, another new enemy will appear in a random place. By using the append () method, it able to adds a single item into the existing list. This method does not return a new list of items but will adjust the original list by adding the item into the list. The enemies appear randomly because users use import random. By using this library, it contains a variety of things to do with random generation.

# iii) The keystroke to control the spaceship



## Figure 7: Keystroke of the game





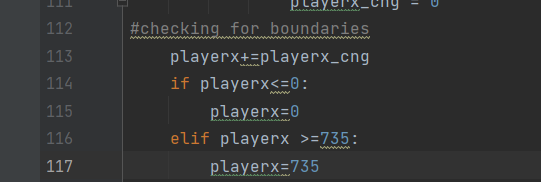
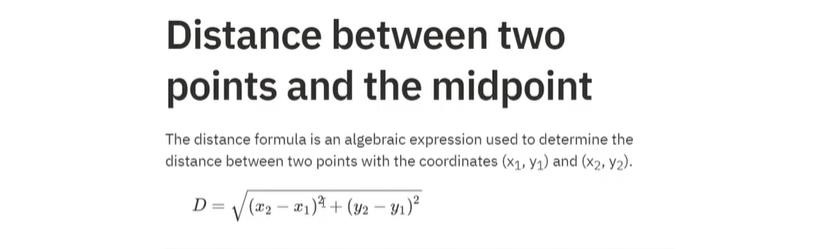


Figure 8: added new variable called Player\_cng = 0

Since the spaceship moves automatically, we need to control it using a keyboard to play. The command line for 92 in **figure 7** needed to be included so that it will loop through all the events that are being stored inside it. From there, it will check whether the keystroke has been pressed or not. Any keystroke that is pressed on the keyboard is an event where it is happening inside our game as input control. For the direction of the spaceship, it can be made by writing pygame.KEYDOWN: , pygame.K\_LEFT: , pygame.K\_RIGHT: , pygame.K\_SPACE: ‘ under the Pygame package. The K\_SPACE is for shooting the bullets. Line 99 and 101 are the speed of movement that we had to choose. From line 112 to 117 is for the spaceship’s boundaries so it won’t go outside the screen. But as we mentioned before, the spaceship will move automatically even though we had decided the key. Therefore, we need to make a new variable which is playerx\_cng = 0.

# iv) Collision concept



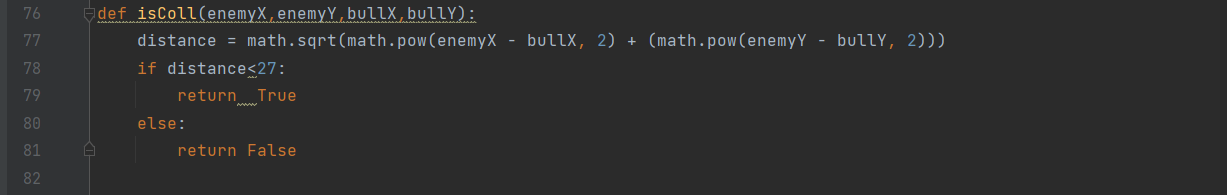
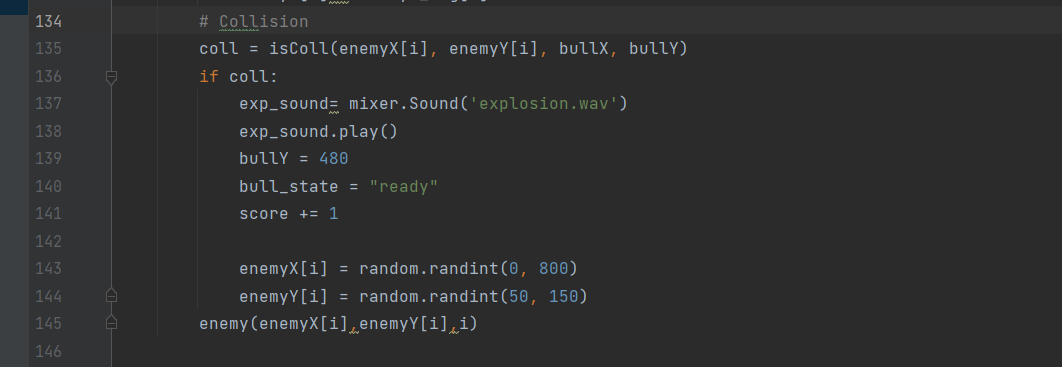


Figure 9: Formula for distance

## Figure 10: Function and distance formula applied Python



## Figure 11: Player who managed to shoot the bullet to the enemy gets 1 point

There are 6 functions are being used in this game. For function hit, it is for the collision. The hit function will help us to define whether the bullet for the enemy had occurred or not. So, to make the two objects collided (bullet and enemey) we need to know the distance formula. Based on picture 1, we had imported math in our coding. The formula that is used in Python is in line 77. By adding the bull\_state = “ready” and a variable of score += 1, whenever the bullet hit the enemey, the player will get 1 point. Line 143 and 144 is for the enemy to be in a default position whenever the bullet managed to shoot it.

### 8.0 Conclusion

During the try and error, our group found that we could not import the pygame package into the game. Therefore, after days of working hard on trying to figure out how to import the pygame we look into digital media where some developer shared their file into their account and it is public mode. From there, we learn to extract the pygame package into our coding. Besides that, one of our group member’s IDE was broken even though he had tried to redownloading it. In fact, it tooks few days to figure out what he should do. Without wasting any time, he decided to learn and use other IDE such as Pycharm to overcome her problems in making this project.