Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica

IE-0117 Programación bajo plataformas abiertas

Profesor:
MSc. Andres Mora Zúñiga

Informe proyecto final-Videojuego

Participantes:

Alberto Mata, carné B74558
Mauricio Figueroa Jimenez, carné B72980
Jose Carlos González Cordero, carné B83403

I-2020

1. Introducción

1.1. Contexto y Justificación

Para la realización de este proyecto se tomaran los conocimientos de programación que se poseen y aplicarlos a la creación de un videojuego. Para estos se intentara crear un videojuego con ciertos niveles, interfaz gráfica y con la facilidad que sea controlado por las teclas de la computadora. Así mismo se la da importancia y enfoque ya que se incursionara en un aspecto nuevo del lenguaje de programación en Python para la creación del juego con la ayuda de la herramienta Pygame. Que en conjunto y mientras se desarrolla tiene como meta desarrollar las habilidades de programación y también para el entretenimiento como meta final ya que esa es la finalidad de un juego. Donde mas adelante se explicara que consiste en varios niveles de una nave espacial enfrentándose a diferentes oponentes como se vera mas a fondo después. Además de que a diferencia de otros juegos de este tipo con cierta monotonía que solo es mover y disparar a objetos que se mantienen estáticos o con movimientos predecibles. Se trata de agregar un cierto grado dificultad extra al tratar de cambiar esto y a los enemigos darles una cierta cantidad de movimiento autónomo o que se muevan libremente aun en fila pero de diferente manera a la que se esta acostumbrado. Haciendo así de un juego mucho mas atractivo y entretenido.

Objetivo General

- Realizar un juego de naves espaciales en el lenguaje de programación Python e implementando la biblioteca pygame. El juego consiste en utilizar una nave espacial,la cual se pueda mover de izquierda a derecha hasta cierto limite de la pantalla y que pueda disparar rayos. Los marcianos son los enemigos a los cuales les hay que disparar. Estos dependiendo del nivel van aumentando su velocidad y aumentando el numero de marcianos que matar. Además dependiendo del nivel, podrán disparar rayos como la nave espacial. Todos los marcianos se mueven juntos y al mismo lado, cuando llegan al limite de algún lado(antes de pegar contra pared izquierda o derecha) avanzan una fila para delante todos los marcianos y se mueven de manera simultanea. Cada nivel se pasa el usuario mediante la nave espacial logra matar todos los marcianos. Se pierde un nivel cuando los marcianos logran matar a al usuario mediante disparos o alcanzando su posición y tocarlo. El usuario cuenta con 3 vidas y una vez que se les acaben pierde. Si pierde y desea volver a jugar, el usuario tendrá que volver a empezar desde el nivel 1.

Objetivos Específicos

- Investigar a fondo la librería Pygame de Python. Explorando así todo lo que se puede hacer mediante esta herramienta
    - Meta: Utilizar el recurso de internet mediante exploradores y videos tutoriales acerca de todas las posibles acciones que la librería Pygame ofrece.
    - Entregable 1:Realizar un resumen que muestre las principales herramientas que ofrece la libreria pygame. Como todos los comandos, además como realizar estructuras básicas fundamentales para realizar un juego.
- Implementar una interfaz gráfica, la cual es la base del juego. En dicha interfaz se contara con los marcianos y la nave espacial

- Meta: Crear la interfaz gráfica que sirva de base para los niveles. Donde se observen las dimensiones que va a ocupar el juego. Además que en esta interfaz se observen los protagonistas(marcianos y nave espacial.

- Entregable 1: Mostrar mediante screenshots y el código la interfaz gráfica , explicando paso a paso lo que se hizo.

- Diseñar en la interfaz gráfica los movimientos básicos de la nave y de los marcianos.La nave debe poder mover moverse de derecha a izquierda y contar con un limite, al igual que los marcianos. La diferencia es que los marcianos los controla la computadora y cuando llegan a algún limite (derecho o izquierdo) proceden a moverse hacia al frente.

  - Meta: Mostrar el primer prototipo de interfaz gráfica que muestra los movimientos básicos que van a poseer los marcianos y los que el usuario va a poder controlar.

  - Entregable 1: Mostrar mediante un video la interfaz gráfica con los movimientos que realizan los marcianos, además de la nave espacial controlada por el usuario. Además de brindar el código explicando paso a paso lo que se hizo.

- Programar que la nave espacial lance rayos y que estos logren matar a los marcianos. Esta herramienta es fundamental ya que de esta forma(lanzando rayos) se logra avanzar hacia el siguiente nivel.Cuando un marciano se muere este debe desaparecer.

  - Meta: Crear la herramienta principal que brinde al usuario la posibilidad de matar a los marcianos mediante rayos.

  - Entregable 1: Realizar un vídeo el cual muestre la acción de la nave tirando rayos directamente matando a un marciano y que este luego de impactar con el rayo desaparezca. Además de brindar el código respectivo explicando paso a paso lo que se hizo.

- Programar que los marcianos lance rayos y que estos logren matar a la nave espacial. Luego de recibir 3 disparos el juego se termina y el usuario pierde.Además programar de igual manera que el usuario pierda una vida cuando el marciano alcanza su posición y lo toca.

  - Meta: Brindar las maneras de hacer que al usuario lo maten y pierda.

  - Entregable 1: Realizar un vídeo el cual muestre la acción de un marciano tirando rayos directamente y matando a la nave y que este luego de impactar con el rayo pierda una vida, luego de 3 impactos que finalice el juego. También mostrar que pasa cuando el marciano toca la nave espacial, que perdería una vida(si ya no le quedaran se terminaría el programa) y se reiniciaría el nivel. Además de brindar el código respectivo explicando paso a paso lo que se hizo

- Realizar el primer nivel y segundo nivel del juego, implementando de esta forma la transición entre un nivel y otro.

  - Meta: Poder jugar el primer nivel y que se logre la transición al segundo nivel.

  - Entregable 1: Realizar un vídeo que muestre la transición del primer nivel al segundo nivel. Además de brindar el código respectivo y su explicación.

- Realizar un total de 10 niveles, cada nivel con un nivel de dificultad superior al otro. Con forme se aumenta de nivel se aumenta en algunos casos el numero de marcianos o la velocidad o se aumentan las habilidades de los marcianos(disparar).
    - Meta: Poder jugar el total de niveles.
    - Entregable 1: Grabar un video que muestre al usuario jugando en los diversos niveles. Además de incluir el código comentado.
- Diseñar e implementar un menú principal que pregunte al usuario presionar una tecla para comenzar a jugar.
    - Meta: Iniciar el programa con un menú principal en vez de ir al primer nivel directamente.
    - Entregable 1: Realizar una captura de pantalla del menú , además de entregar el código comentado.
- Desplegar un menú que al perder le aparezca la opción al usuario de si quiere volver a jugar.
    - Meta: Incorporar que al finalizar el juego (ya sea por ganar o por perder) que aparezca la opción de reiniciar el juego empezando desde el nivel 1 o salir del programa.
    - Entregable 1: Enseñar de manera visual(con video) la opción de reincio del juego cuando el usuario perdió las 3 vidas o ganó el juego luego de superar los 10 niveles.

1.2. Alcances y limitaciones

Alcances

Como alcances se toma primeramente la posibilidad de lograr crear un juego eficiente y funcional. Además de que tambien se quiere agregar la capcidad de que sea controlado por las teclas del teclado y no tenan ningun inconveninte con esto, Ademas de una interfaz grafica que se pueda controlar para que el usuario del juego quiera hacer o modificar mientras hace uso de el. Finalmente lo que se se quiere lograr es un juego para computadora de facil uso y conciso.

Limitaciones

Se presentaron varias limitaciones durante el desarrollo del proyecto, las principales fueron, la falta de tiempo de desarrollo del programa, pues a pesar del planeamiento del cronograma, en el curso se redujo el tiempo por la realización de las otras entregas (Laboratorios, examen, etc), además, al virtualizar el semestre los demás cursos también aumentaron la demanda de tiempo impidiendo el desarrollo del programa con el tiempo que requería. Otra limitación fue la perdida de un integrante del grupo por deserción en el curso, lo cuál aumentó la carga de trabajo del resto de participantes en el proyecto.

1.3. Metodologia

La creación del juego primeramente se basara en los conocimientos de Python que ya se poseen, además de la utilización del modulo Pygame para la creación del juego. El cual hará la creación de ciertas cosas de una manera mas eficiente. Se comenzara con la investigación de Python y en especial el uso de Pygame para la implementación de esta misma dentro del código. Seguidamente luego de la parte de investigación se

comienza creando la base del juego mediante lo ya aprendido mediante Pygame, seguidamente comenzar a darles los aspectos característicos al juego. Tales como lo son la implementación de los personajes, darle movimientos a estos mismos. Seguidamente podemos ir implementando estos personajes en la interfaz gráfica para ver como se comportan y si el funcionamiento es correcto. Se seguiran una serie de pasos después de esto para crear una serie de niveles y de dificultad para el usuario mientras lo controla desde una interfaz gráfica. Hasta llegar a un punto donde se tiene creado el juego con todos sus niveles y dificultades.

2. Marco Teórico

Debido a que el proyecto se va a centrar en la librería de pygame. En esta sección se brinda un resumen básico acerca pygame y las distintas funciones que ofrece y brinda..Fundamental para el proyecto ya que se basa principalmente en el desarrollo del videojuego implementando pygame. El resumen se basa en una serie de 26 vídeos de youtube de un curso de pygame.[1]

2.1. ¿Qué es pygame?

Pygame corresponde a un módulo del lenguaje de programación Python. Este módulo permite de manera sencilla la creación de juegos en dos dimensiones.Esta herramienta permite utilizar objetos, cargar y mostrar imágenes en diferentes formatos, así como sonidos y muchas más opciones. Además que al ser especializado para la programación de videojuegos permite utilizar de manera sencilla el teclado o incluso utilizar un joystick. A pesar de que sea especializado para videojuegos, este no es su único uso ya que se pueden realizar diversos tipos de contenido multimedia como por ejemplo presentaciones,videos y animaciones.Posee la cualidad de ser una multiplataforma, lo que significa que se puede usar en distintos sistemas operativos como Linux, Apple o Windows y no debe de presentarse ningún problema.

2.2. Primer Ventana en pygame

Lo primero que se debe de realizar para crear una pantalla es importar pygame y sys por medio de **import pygame,sys**. Luego se proceden a importar todos los modulos mediante **from pygame.locals import** *. Antes de colocar cualquier modulo en pygame hay que poner **pugame.init()**. Para crear una ventana primeramente se apoya de un objeto de la manera **ventana = pygame.display.set_mode((600,400))** donde los números indican los valores del largo y ancho de la ventana.Note que ventana ahora es un objeto de tipo surface. Para colocarle un nombre a la ventana se realiza el comando **pygame.display.set_caption(**" **Hola Mundo** ")**.** Ahora para mostrar la ventana se corresponderá a crear un loop infinito con el while. En este loop la ventana se actualiza constantemente mediante **pygame.display.update()**. Posteriormente se crea un for dentro del while para recorrer una la lista de eventos predeterminados propios de pygame y así ir comparando que evento a sucedido. Para ello se utiliza la lista de eventos de pygame "*pygame.event.get*()" en el for de manera **for evento in pygame.event.get()**. Adentro del for se crea un if que pregunte si el evento fue de tipo quit(la equis sobre la ventana) de la manera **if evento.type == QUIT:**. Este if se hace verdadero cuando el usuario presiona la equis sobre la ventana, por ende dentro del if se detienen todos los modulos de pygame de la manera **pygame.quit()** y se cierra la ventana de la manera **sys.exit()**. Por ende el código seria:

```
1 import pygame, sys
2 from pygame.locals import*
3
4 pygame.init()
```

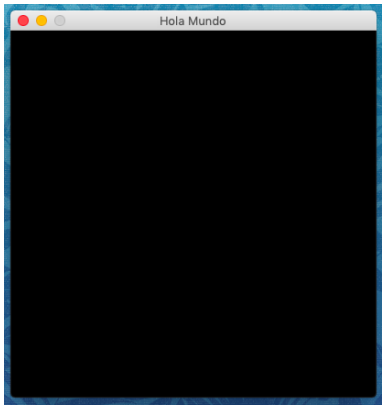

Figura 1: Ventana "Hola Mundo"

```
ventana = pygame.display.set_mode((400,400))
pygame.display.set_caption("Hola Mundo")

while True:
    for evento in pygame.event.get():
        if evento.type == QUIT:
            pygame.quit()
            sys.exit()
    pygame.display.update()
```

El resultado de la ventana se ve en la figura 2.2

2.3. Uso y creación de colores

Para hacer referencia a colores en pygame es necesario partir de la base de los 3 colores primarios los cuales corresponden al rojo, verde y azul. Este sistema es conocido como Sistema RGB por las siglas de los colores en inglés. Los rangos de color van de 0 a 255, por ende 0 es nulo y 255 es el color en su máximo esplendor. Por ende al aplicar algún color se pone de la manera **RGB(0,120,50)**, en este caso no se colocaría nada de rojo, se colocaría 120 de verde y 50 de azul. Mediante estas combinaciones de estos 3 colores es que se forman los diversos colores conocidos. Para crear un color se posee un objeto y se realiza la dupla.Por ende la primera forma de crear un color en pygame es de la manera **Color=(0,120,50)**. La otra manera de crear colores equivale a **Color2=pygame.Color(0,120,50)** que funciona de manera similar al método anterior. Para saber la tonalidad del color lo que se puede hacer es rellenar la pantalla, esto mediante el código **ventana.fill(Color2)** y así ir probando los colores. El código podría verse tomando como referencia el código de la subsección anterior como:

```
import pygame, sys
from pygame.locals import*

pygame.init()
Color1=[0,0,0]
Color2 = pygame.Color(120,20,20)
ventana = pygame.display.set_mode((400,400))
pygame.display.set_caption("Hola Mundo")
```

```
9
10  while True:
11      ventana.fill(Color2)
12      for evento in pygame.event.get():
13          if evento.type == QUIT:
14              pygame.quit()
15              sys.exit()
16      pygame.display.update()
```

El resultado del cambio de color de ventana se ve en la figura 2.3

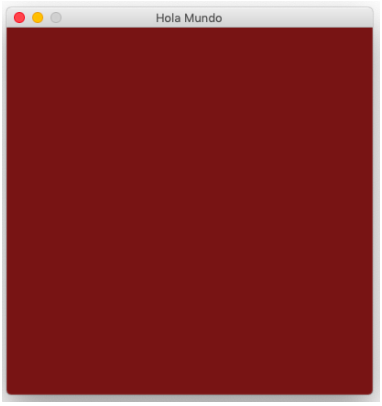

Figura 2: Cambio de color de Ventana "Hola Mundo"

2.4. Dibujo de Líneas

Para la creación de lines es fundamental ver la interfaz gráfica como una matriz donde cada coordenada esta especificada en X y Y y cada punto se llama pixel.Las coordenadas equis permiten desplazarse de izquierda a drecha y viceversa y las de Y de abajo hacia arriba y viceversa. Para poder dibujar cualquier figura geométrica en pygame se debe de utilizar pygame.draw como en el caso de dibujar una linea por ejemplo, esto se hace mediante **pygame.draw.line(ventana,Color2,[60,80],[160,100],8)**, donde lo primero que va dentro del paréntesis es el parámetro que indica donde se va a dibujar la linea, posteriormente posee la indicación del color de la linea,el tercer es muy importante ya que es la ubicación del punto inicial de la linea, por lo que el cuarto parámetro equivale al punto final de la linea. El quinto parámetro no es necesario incluirlo sin embargo cabe mencionar que es el ancho de la linea(un entero con la cantidad de píxeles que se indique).Basándose en las secciones anteriores, el código sería:

```
1  import pygame, sys
2  from pygame.locals import *
3
4  pygame.init()
5  Color1=[0,220,220]
6  Color2 = pygame.Color(120,0,0)
7  ventana = pygame.display.set_mode((400,400))
8  pygame.display.set_caption("Hola Mundo")
9  pygame.draw.line (ventana,Color1,(0,0),(400,400),10)
10
```

```
11
12 while True:
13     for evento in pygame.event.get():
14         if evento.type == QUIT:
15             pygame.quit()
16             sys.exit()
17     pygame.display.update()
```

El resultado de la linea se ve en la figura 2.4

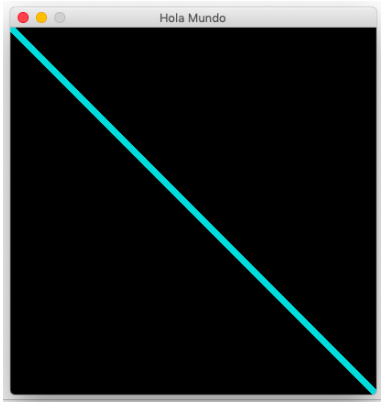

Figura 3: Dibujo de linea en Ventana "Hola Mundo"

2.5. Dibujo de figuras geométricas

Crear figuras geométricas es muy parecido a lo de la subsección anterior que consistía en dibujar lineas.En general para dibujar figuras geométricas los 2 primeros parámetros siempre consisten a donde se va a dibujar y el color. Para dibujar un cicule se hace por medio de **pygame.draw.circle(ventana, Color2, [60,80], [160,100],8)**. El tercer parámetro consistirá en la posición en la que se quiere ubicar el punto central del circulo, el cuarto parámetro es el tamaño del radio del circulo y quinto parámetro como en la sección pasada corresponde al grosor. Para un rectangulo se tiene que **pygame.draw.rect(ventana, Color2, [60,80,160,100],8)**. Donde el tercer parámetro posee 4 valores, donde los primeros 2 hacen referencia a la esquina izquierda superior del rectangulo, el tercer valor equivale al ancho del rectangulo y el cuarto valor corresponde a lo alto. Ahora para dibujar un poligono mediante **pygame.draw.polygon(ventana, Color1, ((0,0),(100,100),(20,90)**, el tercer parámetro consiste poner los puntos y el programa los va a unir, se pueden poner la cantidad de puntos que se deseen. Por ende el código es:

```
1  import pygame, sys
2  from pygame.locals import *
3
4  pygame.init()
5  Color1=[0,220,220]
6  Color2 = pygame.Color(120,0,0)
7  ventana = pygame.display.set_mode((400,400))
8  pygame.display.set_caption("Hola Mundo")
9  pygame.draw.line (ventana,Color1,(0,0),(400,400),10)
10
```

```
pygame.draw.circle(ventana,Color2,(200,200),100,10)
pygame.draw.rect(ventana,Color1,(240,150,100,40))
pygame.draw.polygon(ventana,Color2,((200,200),(250,250),(10,150)))

while True:
    for evento in pygame.event.get():
        if evento.type == QUIT:
            pygame.quit()
            sys.exit()
    pygame.display.update()
```

El resultado de la linea se ve en la figura 2.5

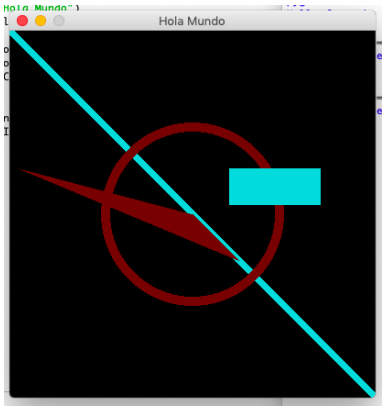

Figura 4: Dibujo de figuras en Ventana "Hola Mundo"

2.6. Métodos de load y blit

Primeramente para cargar una imagen se crea un objeto de manera **Miimagen = pygame.image.load("Miimagen.png")** donde la imagen que se quiere importar deberá ubicarse con el nombre exacto. En caso de que esté dentro de otra carpeta se pone como por ejemplo Misimagenes/Miimagen.png. Posteriormente se va a ver la posición en la que se va a trabajar mediante **posX,posY=130,70** o de manera equivalente poner posX=130 y posY=70. Cabe mencionar que ya que posee forma rectangular, el punto que se coloca es el superior izquierdo.Ahora se utiliza **venatana.blit(Miimagen,(posX,posY)**. El código se muestra en:

```
import pygame,sys
from pygame.locals import *

pygame.init()
Color1=[0,220,220]
Color2 = pygame.Color(120,0,0)
ventana = pygame.display.set_mode((400,400))
pygame.display.set_caption("Hola Mundo")
pygame.draw.line (ventana,Color1,(0,0),(400,400),10)

#pygame.draw.circle(ventana,Color2,(200,200),100,10)
#pygame.draw.rect(ventana,Color1,(240,150,100,40))
#pygame.draw.polygon(ventana,Color2,((200,200),(250,250),(10,150)))
```

```
14
15 Imagen = pygame.image.load("imagen1.png")
16 posX,posY= 130,70
17
18 ventana.blit(Imagen,(posX,posY))
19
20 while True:
21     for evento in pygame.event.get():
22         if evento.type == QUIT:
23             pygame.quit()
24             sys.exit()
25     pygame.display.update()
```

El resultado de la linea se ve en la figura 2.6

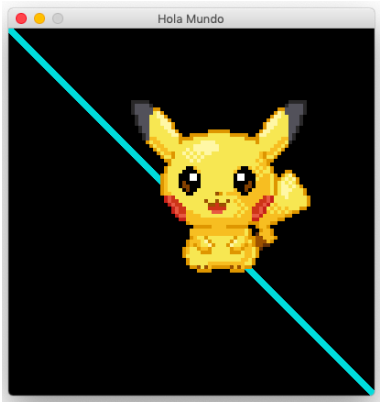

Figura 5: Imagen en Ventana "Hola Mundo"

2.7. Método Random

Mediante este código se logra dibujar la imagen insertada en la sección anterior en otras coordenadas y esto es posible gracias a Random. Primeramente se importa randint mediante **import random import randint** para crear coordenadas aleatorias. Para utilizar randint hay que ingresar 2 números, el primero es de que numero se va a partir para tener coordenadas aleatorias y el segundo parámetro es el máximo de manera **posX=randint(10,400**. Por ende se tiene el código:

```
1  import pygame,sys
2  from pygame.locals import *
3  from random import randint
4
5  pygame.init()
6  Color1=[0,220,220]
7  Color2 = pygame.Color(120,0,0)
8  ventana = pygame.display.set_mode((400,400))
9  pygame.display.set_caption("Hola Mundo")
10 pygame.draw.line (ventana,Color1,(0,0),(400,400),10)
11
12 #pygame.draw.circle(ventana,Color2,(200,200),100,10)
13 #pygame.draw.rect(ventana,Color1,(240,150,100,40))
```

```
#pygame.draw.polygon(ventana,Color2,((200,200),(250,250),(10,150)))

Imagen = pygame.image.load("imagen1.png")
posX=randint(0,300)
posY= randint(0,300)

ventana.blit(Imagen,(posX,posY))

while True:
    for evento in pygame.event.get():
        if evento.type == QUIT:
            pygame.quit()
            sys.exit()
    pygame.display.update()
```

El resultado de la linea se ve en la figura 2.7 y 2.7 donde cada vez que se compila la imagen de Pikachu aparece en un nuevo luegar.

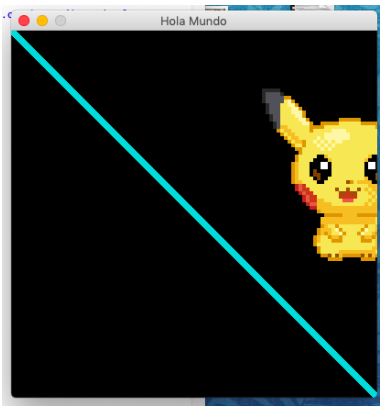

Figura 6: Imagen con coordenadas aleatorias Ventana "Hola Mundo"

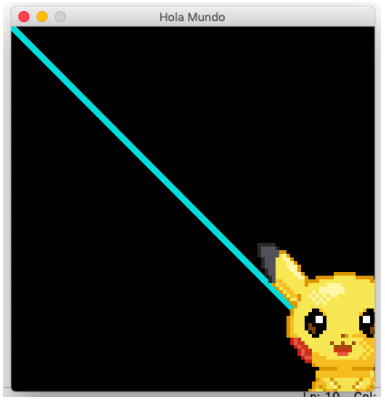

Figura 7: Imagen con coordenadas aleatorias Ventana "Hola Mundo"

2.8. Animaciones Básicas

Primeramente para crear una animacion de la imagen se mueve la sentencia **ventana.blit(Imagen,(posX, posY))** dentro del ciclo **while**, lo cual permitirá que la imagen se vaya actualizando dentro del while con el método **update**. Apoyándose en este método se agrega la velocidad a la que queremos que se mueva, tratando de que sea fluida se le da un valor bajo y estable. Para el movimiento de la imagen creamos un algoritmo con un **if** que mientras no se salga de los margenes que se siga moviendo y cuando se acerque este disminuya y no se salga de la ventana. O mediante la creación de una variable booleana podemos hacer que este se mueva de derecha a izquierda tal como se ve en el ejemplo de código siguiente. Que lo que hace es que va sumando el valor de la velocidad que le damos hasta que llega al limite que se le impone en el ciclo **if**.

```
import pygame, sys
from pygame.locals import*
from random import randint

pygame.init()
ventana = pygame.display.set_mode((###,###))
pygame.display.set_caption("Ejemplo de codigo")

Imagen = pygame.image.load("Imagen deseada")
posX= 200
posY= 100
#velocidad que se le quiere dar de movimiento por pixel
velocidad=###
#Fondo de la ventana en la que se corre la animacion y su color
color=(###,###,###)

derecha=True;

while True:
    ventana.fill(color)
    ventana.blit(Imagen,(posX,posY))
    for event in pygame.event.get():
        if evento.type == QUIT:
            pygame.quit()
            sys.exit()
    #los valores se le dan hasta donde quiere que llegue la img
    if derecha=True:
        if posX<400:
            posX+=velocidad
        else:
        derecha=false

    else:
        if posX>1:
            posx+=velocidad
        else:
            derecha=True


    pygame.display.update()
```

2.9. Movimientos por medio del teclado

Se procede así de nueva manera dentro del ciclo **for**, si no fue un evento de tipo salida o **QUIT**, la cual nos indica la X en nuestra ventana, que nos pregunte que tipo fue el que ocurrió. Así como si alguna tecla fue tocada y si así es cual de estas fue, y si fue alguna de estas que se mueva en la dirección que se toca en el teclado. Se puede lograr controlar estos movimientos mediante el código siguiente. Diciendo así que en caso de presionar una de las teclas ya sea derecha o izquierda se mueva a la velocidad de píxeles asignados las veces que presionemos las teclas.

```
import pygame, sys
from pygame.locals import*
from random import randint

pygame.init()
ventana = pygame.display.set_mode((###,###))
pygame.display.set_caption("Ejemplo de codigo")

Imagen = pygame.image.load("Imagen deseada")
posX= 200
posY= 100
#velocidad que se le quiere dar de movimiento por pixel
velocidad=###
#Fondo de la ventana en la que se corre la animacion y su color
color=(###,###,###)

derecha=True;

while True:
    ventana.fill(color)
    ventana.blit(Imagen,(posX,posY))
    for event in pygame.event.get():
        if evento.type == QUIT:
            pygame.quit()
            sys.exit()

       elifevent.type==pygame.KEYDOWN
            if event.key==K_LEFT:
                posX-=velocidad
            elif event.key==K_RIGHT:
                posX+=velocidad


    pygame.display.update()
```

2.10. Uso del cursor

Lo que se desea de esta manera es que mediante el cursor se pueda mover la imagen o en el caso del juego algún personaje a nuestra conveniencia. Para esto se usa el comando **pygame.mouse.get_pos()** el cual indica la posición del cursor dentro de la ventana creada para el juego mediante una tupla. Debidamente como ya tenemos los objetos creados para la posición en X y en Y dentro de la ventana solo se igualan a este método para poder controlarlo con el cursor ya que estos forman una tupla que son lo que es lo que recibe como entrada el método. Así como en el siguiente codigo.

```
import pygame, sys
```

```
from pygame.locals import*
from random import randint

pygame.init()
ventana = pygame.display.set_mode((###,###))
pygame.display.set_caption("Ejemplo de codigo")

Imagen = pygame.image.load("Imagen deseada")
posX= 200
posY= 100
#velocidad que se le quiere dar de movimiento por pixel
velocidad=###
#Fondo de la ventana en la que se corre la animacion y su color
color=(###,###,###)

derecha=True;

while True:
    ventana.fill(color)
    ventana.blit(Imagen,(posX,posY))
    for event in pygame.event.get():
        if evento.type == QUIT:
            pygame.quit()
            sys.exit()

       elifevent.type==pygame.KEYDOWN
            if event.key==K_LEFT:
                posX-=velocidad
            elif event.key==K_RIGHT:
                posX+=velocidad

        #metodo que nos entrega lamosicion asignado al control de la posicion en X y en Y
            de la imagen dada
    posX,posY= pygame.mouse.get_pos()

    pygame.display.update()
```

Al igual que visto por movimientos por teclados el movimiento por cursor funciona también para posicionar la imagen en este caso de donde lo queremos. Tal como se ve en las figuras 2.10 y 2.10 los dos métodos se pueden usar como controladores.

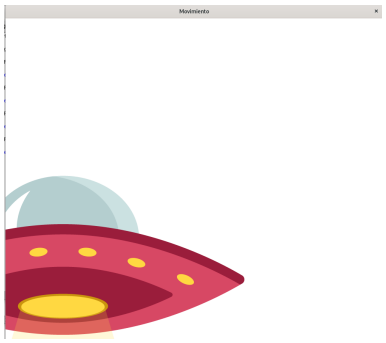

Figura 8: Figura de un lado controlado por el cursor

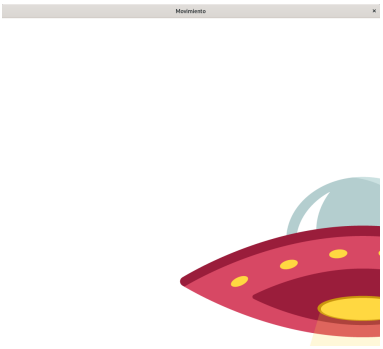

Figura 9: Figura del lado opuesto

2.11. Colisiones

Esta parte es muy útil ya que puede ser de mucha utilidad al momento de crear juegos de tirador o si es necesario que se toquen los personajes en el juego se le puede dar una condición mas adelante, de que este pueda morir, cause daño, se acelere o aspectos así dependiendo lo que queremos hacer con esto. Para esta parte el comando **objeto.colliderect(objeto2)** dentro de un ciclo **if**. Ya que si dos objetos llegaran a colisionar se le pueda decir que haga después de esto. Así como se puede ver en el ejemplo siguiente, creando dos objetos que al momento de colisionar uno de ellos se detenga. En este caso lo que se produce es que los dos objetos mientras colisionan hace que se detengan.

```
1  import pygame,sys
2  from pygame.locals import*
3  from random import randint
4
5  pygame.init()
6  ventana = pygame.display.set_mode((###,###))
7  pygame.display.set_caption("Ejemplo de codigo")
8
9  Imagen = pygame.image.load("Imagen deseada")
10 posX= 200
11 posY= 100
12 #velocidad que se le quiere dar de movimiento por pixel
13 velocidad=###
14 #Fondo de la ventana en la que se corre la animacion y su color
15 color=(###,###,###)
16
17 derecha=True;
18 objeto=pygame.Rect(0,0,100,50)
19 objeto2=pygame.Rect(0,0,100,50)
20
21 while True:
22     ventana.fill(color)
23     pygame.draw.rect(ventana,(100,70,70), objeto)
24
25     pygame.draw.rect(ventana,(100,70,70), objeto2)
26
27     objeto.left,objeto.top=pygame.mouse.get_pos()
28     if objeto.colliderect(objeto2):
```

```
        velocidad=0

    for event in pygame.event.get():
        if evento.type == QUIT:
            pygame.quit()
            sys.exit()

    if derecha=True:
        if posX<400:
            posX+=velocidad
            objeto2.left=posX
        else:
        derecha=false

    else:
        if posX>1:
            posx+=velocidad
            objeto2.left=posX
        else:
            derecha=True


    pygame.display.update()
```

2.12. Texto

Esta herramienta es muy útil también, ya que nos permite visualizar cualquier tipo de texto en la ventana que estamos creando para el juego. Así que mediante el podemos crear instrucciones durante el juego, dar indicaciones, si se esta jugando agregar animaciones de texto o en caso de que sucediera algo dentro del juego que este tire un texto diciendo lo que sucede. Se puede hacer creando objetos para tipo fuente mediante el método **pygame.font.Font()**. y el metodo **render** el cual creara el texto para utilizar en la ventana del juego. Además de que e le pueden dar parámetros para hacer que el texto se vea claro. Además de que se pueden extraer fuentes del sistema con el comando **pygame.font.SysFont("La fuente que se desea",)**. Todo esto como ejemplo en el siguiente código.

```
import pygame,sys
from pygame.locals import*

pygame.init()
ventana = pygame.display.set_mode((400,400))
pygame.display.set_caption("Prueba")


Fuente1=pygame.font.Font(None,30)
texto1=Fuente1.render("Prueba",0,(200,65,85))

Fuente2=pygame.font.SysFont("Arial",30)
texto2=Fuente2.render("Prueba2",0,(200,65,85))


while True:
    for evento in pygame.event.get():
        if evento.type == QUIT:
            pygame.quit()
            sys.exit()
```

```
21
22      ventana.blit(texto1,(0,0))
23      ventana.blit(texto2,(50,50))
24      pygame.display.update()
```

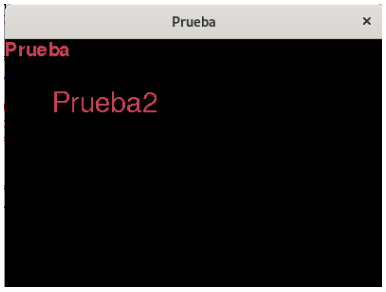

Figura 10: Ventana creada mostrando el texto dado

2.13. Tiempo o temporizador

Herramientas como esta son útiles en caso de que si se esta creando un nivel y se toma por ejemplo los juegos de Mario Bros, que por niveles tienen tiempo predeterminando para realizarlos y si no se pierde. Esta herramienta puede servir para esto ya que podría darse limites de tiempo, también contar el tiempo transcurrido de juego o pruebas con tiempo dentro del mismo juego. Para esto se usa **pygame.time.get_ticks()** el cual cuenta el tiempo transcurrido en milisegundos desde que ocurre la ejecución. Se puede usar así para contar o tomar el tiempo de las cosas previamente mencionadas. En el siguiente ejemplo se toma el tiempo y se muestra en pantalla utilizando las herramientas de texto previamente mencionadas.

```
1   import pygame,sys
2   from pygame.locals import*
3
4   pygame.init()
5   ventana = pygame.display.set_mode((400,400))
6   pygame.display.set_caption("Prueba␣TIEMPO")
7
8
9   Fuente2=pygame.font.SysFont("Arial",30)
10  aux=1
11
12  while True:
13      ventana.fill((255,255,255))
14      Tiempo=pygame.time.get_ticks()/1000
15
16      if aux==Tiempo:
17          aux+=1
18          print (Tiempo)
19      for evento in pygame.event.get():
20          if evento.type == QUIT:
21              pygame.quit()
22              sys.exit()
23
24      contador=Fuente2.render("Tiempo:" + str(Tiempo),0,(120,50,10))
```

```
25      ventana.blit(contador,(50,50))
26      pygame.display.update()
```

Figura 11: Ventana creada que muestra un temporizador

2.14. Sprites

Los sprites son imágenes bidimensionales que forman parte del juego. Son por lo general objetos que llegan a interactuar entre ellos para hacer que el juego funciones. La biblioteca Pygame soporta el uso de estos mismos por lo que se puede usar de una manera muy fácil. Dándoles instrucciones de movimiento y demás comportamiento dentro del juego. Las figuras 2.14 y 2.14 muestran algunos de estos usados anteriormente para los juegos.

Figura 12: Ejemplo de sprites usados por consolas pasadas

2.15. Movimiento de enemigos

Para el juego los enemigos deben tener una un movimiento remoto donde se mueven hacia los lados y conforme completan el movimiento bajan acercándose al jugador. El movimiento de la imagen se basa en tres atributos; el primero para permitir el movimiento (de derecha a izquierda en este caso), el segundo será un contador para limitar el movimiento y que que no sea infinito hacia un lado y permita el descenso y el tercero para limitar un descenso máximo. el código de movimientos se muestra a continuación:

```
1   def _movimientos(self):
2       if self.contador <3:
3           self._movimientoLateral()
4       else:
```

Figura 13: Ejemplo de sprites usados por consolas pasadas

```
        self.rect.top +=1

def _desenso(self):
    if self.Maxdesenso == self.rect.top
        selfcontador = 0
        self.Maxdesenso == self.rec.top + 40
    else:
        self.rect.top +=1

def _movimientoLateral(self):
    if self.derecha == True:
        self.rect.left =     self.rect.left + self.velocidad
        if self.rect.left > 500:
            self.derecha = False

            self.contador +=1
    else:
        self.rect.left. = self.rect.left - self.velocidad
        if self.rect.left < 0:
            self.derecha = True
```

2.16. Música y sonidos

Toda la música ysonido para implementar en pygame se encuentra en la extensión textbpygame.mixer. Para cargar los sonidos y reproducirlos se utiliza textbfpygame.mixer.music.load('ruta') y textbfpygame.mixer.music.play())(en este caso el indica la cantidad de veces) respectivamente como están mencionadas.

Para agregar sonidos específicos se crea un atributo especifico para la acción que requiere el sonido y se le asigna el sonido como se mencionó anteriormente, y para reproducirlo se asigna el sonido en la función del a acción, si tomamos como ejemplo disparar el código será:

```

```

```
pygame.mixer.music.load('Sonidos/Intro.mp3')
pygame.mixer.music.play(3)

self.sonidoDisparo = pygame.mixer.Sound("Sonido/laserSpace.wav")

def disparar(self,x,y):
    miProyectil = Proyectil (x,y,"Imagenes/disparoa.jpg", True)
    self.listaDisparo.append(miProyectil)
    self.sonidoDisparo.play()
```

2.17. Lista de enemigos

Para incluir a todos los enemigos se deben hacer modificaciones en las funciones anteriores, primero se modifica el rango de movimiento del enemigo ya insertado, posteriormente se le deben agregar dos nuevos parámetros a la función invasor para que aplique para todos los enemigos, los parámetros de imagen y uno de distancia. Además, agregar dos atributos para limitar el movimiento de izquierda y movimiento de derecha. Ya modificado creamos una función para cargarlos y definimos una lista. Por último se brinda un condicional para saber si aun hay enemigos y si el usuario ganó.

```
def cargarEnemigos():
    posx=100
    for x in range(1,5):
        enemigo = Invasor(100,100,40, 'Imagenes/MarcianoA.jpg', ImagenesMarcianoB.jpg)
        listaEnemigo.append(enemigo)
        posx = posx + 200
    posx=100
    for x in range(1,5):
        enemigo = Invasor(100,100,40, 'Imagenes/Marciano2A.jpg', ImagenesMarciano2B.jpg)
        listaEnemigo.append(enemigo)
        posx = posx + 200
    posx=100
    for x in range(1,5):
        enemigo = Invasor(100,100,40, 'Imagenes/Marciano3A.jpg', ImagenesMarciano3B.jpg)
        listaEnemigo.append(enemigo)
        posx = posx + 200
```

2.18. Coliciones contra enemigos

En este apartado modificamos la función spaceInvader en la parte donde se recorre la lista de enemigos y se compara mediante un for la posición del proyectil lanzado con la posición de todos los enemigos de la lista, mediante un condicional si alguna posición coincide se elimina el enemigo

```
for enemigo in listaEnemigo:
    if x.rect.colliderect(enemigo.rect):
        listaEnemigo.remove(enemigo)
        jugador.listaDisparo.remove(x)
```

2.19. Empaquetamiento

Se debe crear un paquete para añadir las diferentes clases que se implementaron durante el juego, las cuales son tres invasor, nave y proyectil. Para esto se crean crear tres archivos distintos, uno para cada

clase y copiar el código de cada clase en su archivo correspondiente. Posterior a esto se crea un archivo llamado *__init__.py* e importar las clases.

```
from Invasor import Invasor
from Proyectil import naveEspacial
```

Por último, en el archivo principal se debe implementar el paquete que está en *__init__.py*

```
from Clases import Nave
from Clases import invasor as Enemigo
```

2.20. Fin del juego

Para finalizar el juego se deben realizar varios detalles como lo es la implementación de una función para detener los procesos llamada detenerTodo() y creando el implemento llamado conquista. relacionando con un condicional para saber cuando se haya ganado. Se detiene la música y hace que se despliegue fin de juego en texto.

```
def detenerTodo()
    for enemigo in listaEnemigo:
        for disparo in enemigo.listaDisparo:
            enemigo.listaDisparo.remove(disparo)
        enemigo.conquista = True

#al final del c digo de  la funci n space invaders antes de actualizar el juego:
Texto = miFuenteSistema.render("Fin del juego",0,(120,100,40))
if enJuego= False:
    pygame.mixer.music.fadeout(3000)
    venta.blit(Texto,(300,300))
```

Por [ultimo mediante la función destrucción se le agrega la imagen de explosión y se reproduce una música de perdida.

3. Estado del arte

EL vídeo juego a realizar en el proyecto está basado en el famoso juego "space invasors"(Invasores del espacio), adaptado al entorno e interfaz de python a través de pygame que es un modulo que forma parte de python que permite y facilita la creación de juegos de dos dimensiones .

Juegos como Space Invaders instauro el concepto de Hi-Score (Puntaje mas alto) y el de jugar con una cantidad limitada de vidas , permitiéndole al los jugadores jugar un mayor tiempo , este juego es pionero de los juegos de disparo adicional a este la influencia del cine es los años de su creación películas como Start Wards potencializaron la venta del mismo

Space Invaders es un videojuego de disparos, japonés lanzado en 1978 por la compañía Taito . Fue creado por Tomohiro Nishikado , un japones creador de videojuegos. Este juego se considera precursor de los videojuegos modernos y fue esencial para el crecimiento de la industria de los videojuegos. Primero se lanzó como un juego de arcade y luego se rehizo en diferentes plataformas; relanzamientos incluyen portada y versiones actualizadas. Las versiones portadas generalmente presentan diferentes gráficos y opciones de juego adicionales, que incluyen búnkers de defensa en movimiento, disparos en zigzag, alienígenas invisibles y modos de dos jugadores . Space Invaders es una de las franquicias de videojuegos más taquilleras

de todos los tiempos y como se menciono anteriormente es un de lo juegos bases para el desarrollo de distintas modificaciones del mismo.

En los primeros años posteriores a la salida del juego, Taito lanzó diferentes versiones del juego añadiendo algunas funcionalidades, colores y gráficos al juego en las versiones.

1979: Space Invaders ™ Deluxe se convierte en el primer título de Space Invaders ™ que permite a los jugadores ingresar su nombre. El juego también introdujo el OVNI de 500 puntos, y el error en el que los jugadores recibieron una bonificación después de golpear una combinación de arco iris se reconoció como una característica oficial. (Space invaders, 2020) [2].

Posterior a esto se fueron sacando versiones del juego para diferente consolas, entre las primeras plataformas fue para NES (consola antigua de Nintendo) en 1985, hasta incluso llegar a estar para Play Station (1997) y Gameboy(2002). En la actualidad se han creado versiones para las nuevas consolas basadas en la idea de juego, Por ejemplo el juego 8 bit invaders que a pesar de ser un juego creado para la consola play station 4 la cual cuenta con recursos muy superiores en comparación a los que se tenían hace algunas décadas la funcionabilidad se base en el space invaders.

Space Invaders y varias de sus secuelas de arcade a menudo se incluyen en las compilaciones de videojuegos lanzadas por Taito. Este videojuego fue tomado como referencia para muchos otros juegos, muchas compañías crearon copias del juego, como Super Invader y TI Invaders, mientras que otros se basaron en el juego original, como Galaxian y Galaga.

A continuación se presenta un código de programación del juego space invaders. [3]

```
#-----!!!!SPACE INVADERS!!!!-----
import pygame, sys
from pygame.locals import *
#-----MAIN FUNCTIONS-----
def movement(move_x):
    if event.type == KEYDOWN:
        if event.key == K_LEFT:
            move_x = -5
        if event.key == K_RIGHT:
            move_x = 5
    if event.type == KEYUP:
        if event.key == K_LEFT:
            move_x = 0
        if event.key == K_RIGHT:
            move_x = 0
    return move_x


#-----FFRAME RAEE / SCREEN SIZE-----
clock = pygame.time.Clock()
w,h = 800,800
screen = pygame.display.set_mode((w,h))

#-----SETTING IMAGES-----
pygame.mouse.set_visible(0)

ship = pygame.image.load("spaceship.png")
ship = pygame.transform.scale(ship,(100,50))
ship_top = screen.get_height() - ship.get_height()
ship_left = screen.get_width()/2 - ship.get_width()/2

screen.blit(ship, (ship_left,ship_top))

shot1 = pygame.image.load("SingleBullet.png")
```

```
shot1 = pygame.transform.scale(shot1,(25,25))
shot2 = shot1
shot_count = 0
shot_y = 0
shot_y_2 = 0

#-----GLOBAL VARIABLES-----
x = 0
resetShot = 0
move_x = 0
#-----MAIN GAME LOOP-----
while True:
    clock.tick(60)
    screen.fill((0,0,0))
    #x,y = pygame.mouse.get_pos()
    screen.blit(ship, (x-ship.get_width()/2,ship_top))

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()

        move_x = movement(move_x)

        if event.type == KEYDOWN:
            if event.key == K_SPACE and shot_count == 0:
                shot_y = h-50
                shot_x = x
            elif event.type == K_SPACE and shot_count == 1:
                shot_y_2 = h-50
                shot_x_2 = x
            print(h, ' ', shot_y, shot_count)
        if event.type == KEYUP:
            if event.key == K_SPACE and shot_count == 0:
                resetShot = 0
            elif event.type == K_SPACE and shot_count == 1:
                resetShot = 0


    if shot_y > 0:
        screen.blit(shot1, (shot_x-shot1.get_width()/2,shot_y))
        shot_y -= 15
    if shot_y_2 > 0:
        screen.blit(shot2, (shot_x_2-shot1.get_width()/2,shot_y_2))
        shot_y_2 -= 15

    x+=move_x
    pygame.display.update()
```

4. Cronograma

Semana	Objetivo Específico	Entregable
1-2	1	Realizar un resumen que muestre las principales herramientas que ofrece la libreria pygame. Como todos los comandos, además como realizar estructuras básicas fundamentales para realizar un juego.
2	2	Mostrar mediante screenshots y el código la interfaz gráfica , explicando paso a paso lo que se hizo.
2-3	3	Mostrar mediante un video la interfaz gráfica con los movimientos que realizan los marcianos, además de la nave espacial controlada por el usuario. Además de brindar el código explicando paso a paso lo que se hizo.
2-3	4	Realizar un vídeo el cual muestre la acción de la nave tirando rayos directamente matando a un marciano y que este luego de impactar con el rayo desaparezca. Además de brindar el código respectivo explicando paso a paso lo que se hizo.
3	5	Realizar un vídeo el cual muestre la acción de un marciano tirando rayos directamente y matando a la nave y que este luego de impactar con el rayo pierda una vida, luego de 3 impactos que finalice el juego. También mostrar que pasa cuando el marciano toca la nave espacial, que perdería una vida(si ya no le quedaran se terminaría el programa) y se reiniciaría el nivel. Además de brindar el código respectivo explicando paso a paso lo que se hizo
3-4	6	Realizar un vídeo que muestre la transición del primer nivel al segundo nivel. Además de brindar el código respectivo y su explicación.
3-5	7	Grabar un video que muestre al usuario jugando en los diversos niveles. Además de incluir el código comentado.
6	8	Realizar una captura de pantalla del menú , además de entregar el código comentado.
6	9	Enseñar de manera visual(con video) la opción de reincio del juego cuando el usuario perdió las 3 vidas o ganó el juego luego de superar los 10 niveles.

Cuadro 1: Tabla de entregables

5. Experimento y Análisis de resultados

En esta sección se dará una explicación sobre el funcionamiento del código según el orden de prioridad y uso que se les da, se explicaran en su mayoría dividiendo por clases y por último explicando el funcionamiento del juego como función principal, programa está constituido por varias clases, como lo son la *"naveEspacial"*(jugador), *"Alienverdef"*, *"Alien*2*"*, *"proyectil"* y tres funciones, una principal y las otras dos secundarias, las cuales son *"Juego"* y *"cargarEnemigos"*, respectivamente como fueron mencionadas . A continuación se explicará detalladamente cada clase y función, demás de las funciones que sirven dentro de las clases:

5.1. Clase naveEspacial

Esta es una de las clases principales ya que crea al jugador con todas las caracteristicas que se quieren, además de que se la dan ciertos atributos, para asi poder usar **Sprites** en la misma clase, que se dibuje en la ventana, ademas de que recibe las posiciones donde se quiere dibujar, la velocidad de su movimiento y se obtiene el rectangulo de la imagen para poder crear colisiones con los aliens. Se define también una lista para almecenar los disparo del mismo y rango para definir estos disparos. Tiene definidos diferentes

tipos de metodos, asi como el metodo **disparar**, el cual usando las coordenadas de la nave se las atribuye al proyectil a crear y se agrega a la lista de disparos de la misma clase para así poder disparar cuantas veces se quiera. Finalmente teniendo una función **dibujar** la cual usa la ventana como parametro para dibujarse en ella. Por último se define un metodo llamado **Movimiento** para restringir el movimiento de la nave dentro de la ventana. Acontinuación se muestra el código de la clase.

```
class naveEspacial(pygame.sprite.Sprite):

    def __init__(self):
        #Metodo para poder usar los sprites
        pygame.sprite.Sprite.__init__(self)
        #Creamos la imagen del sprite
        self.ImageNave =pygame.transform.scale(pygame.image.load("spaceship.png"),(70,70)
            )
        #Obtenemos el rectangulo proveniente de la img
        self.rect = self.ImageNave.get_rect()
        #Corrdenadas X y Y de cada
        self.rect.centerx = ancho-400

        self.rect.centery = alto-100


        #Lista para almacenar disparos
        self.listaDisparo = []
        #Vida inicial que la iniciamos siempre en True
        self.vida = True
        #Velocidad para moverse en pantalla
        self.velocidad=100
        #self.sonidoDisparo= pygame.mixer.Sound("laser2.wav")

    def movimiento(self):
        #Se restringe el movimiento dentro de la ventana
        if self.vida==True:

            if self.rect.left<=0:
                self.rect.left=0
            elif self.rect.right>=800:
                self.rect.right=800

    def disparar(self,x,y):
        #Creacion del objeto proyectil con las coordenadas x y y
        miProyectil=proyectil(x, y, True,"disparoa.jpg")
        #self.sonidoDisparo.play()
        self.listaDisparo.append(miProyectil)

    #Se crea la img en la ventana
    def dibujar(self, superficie):
        #Pone la imagen de la nave en la ventana
        superficie.blit(self.ImageNave, self.rect)
```

5.2. Clase proyectil

Esta clase al igual que la de naveEspacial, se define primeramente poder usar los sprites. Del mismo se le asignan valores de posicion y velocidad ademas de obtener el rectangulo de la imagen para poder colisionarlos después. Se define un estado de personaje(alien o nave Espacial) el cual nos dice que si es **True**

o **False**, se le asigna un movimiento ya sea para arriba o para abajo. Esto mediante el metodo **trayectoria**. Incluyendo tambien el metodo **dibujar** para poder dibujar la imagen sobre la ventana cada vez que el programa actualice. A continuación una demostración del código.

```
class proyectil(pygame.sprite.Sprite):

    def __init__(self, posx, posy, personaje,ruta):

        pygame.sprite.Sprite.__init__(self)
        #cargamos la img de disparo
        self.disparo =pygame.transform.scale(pygame.image.load(ruta),(20,20))
        #Se obtiene el rectangulo
        self.rect = self.disparo.get_rect()

        #Atributos de colocacion y velocidad
        self.velocidadDisparo=10
        self.rect.top=posy
        self.rect.left=posx-10

        self.disparoPersonaje = personaje

    def trayectoria(self):
        if self.disparoPersonaje:
            self.rect.top=self.rect.top-self.velocidadDisparo
        else:
            self.rect.top=self.rect.top+self.velocidadDisparo

    def dibujar(self, superficie):
        superficie.blit(self.disparo, self.rect)
```

5.3. Clase Alienverdedef

En esta clase se llama a un enemigo para que aparezca en pantalla. Se definen en el so de los sprites dentro del mismo. Se le dan los atributos de velocidad, posicion y se obtiene el rectangulo de la figura dado por el parametro ruta. Asi al llamarlo en la funion principal se le da como uno de los argumentos la img a cargar. Se define un rango para asi poder crear un metodo **__ataque**, el cual al llamarlo crea numeros aleatorios y si estos estan dentro del rango se llama la funcion **__disparo** para así poder disparar aleatoriamente con el marciano, además, se agregan los disparos a la lista creada llamada *listaDisparo* para que dispare ilimitadamente de forma aleatoria, es decir, llamando a la función **Randint** de la biblioteca ramdom y cuando se encuentre en el rango (variable definida) dispare. A continuación una muestra del código

```
class Alienverdedef(pygame.sprite.Sprite):

    def __init__(self, posx, posy,velocidad,ruta):

        pygame.sprite.Sprite.__init__(self)

        self.alienverde =pygame.transform.scale(pygame.image.load(ruta),(50,50))
        self.rect = self.alienverde.get_rect()
        self.velocidadAlienverde=velocidad
        self.rect.y=posy#posicion del alien en y
        self.rect.x=posx#posicion del alien en x
        self.listaDisparos=[]
```

```
13          self.disparo=True
14
15          self.rango=10
16
17          self.__ataque()
18
19      def __disparo(self):
20          x,y=self.rect.center
21          miProyectil=proyectil(x, y, False, "disparo.jpg")
22          self.listaDisparos.append(miProyectil )
23
24      def __ataque(self):
25          if(randint(0, 1500)<self.rango):
26              self.__disparo()
27
28      def dibujar(self, superficie):
29          self.__ataque()
30          superficie.blit(self.alienverde, self.rect)
```

5.4. Clase Alien2

En esta clase se crea un alien con las funciones mas basicas, no dispara y avanza a una velocidad estandar. Lo primero que se realiza es el llamado de los sprites al self y con esto se incializan las demás variables de la clase como posición, velocidad y obtención del rectangulo de la figura para las colisiones, además de cargar la imagen del alien. en esta clase se encuentra la función **dibujar** que consisteen introducir en la ventana la imagen en la ventana cada vez que se actualiza.

```
1   class Alien2(pygame.sprite.Sprite):
2
3       def __init__(self, posx, posy,velocidad,ruta):
4
5           pygame.sprite.Sprite.__init__(self)
6
7           self.alienverde =pygame.transform.scale(pygame.image.load(ruta),(50,50))
8           self.rect = self.alienverde.get_rect()
9           self.velocidadAlienverde=velocidad
10          self.rect.y=posy#posicion del alien en y
11          self.rect.x=posx#posicion del alien en x
12          self.disparo=False
13
14
15      def dibujar(self, superficie):
16          superficie.blit(self.alienverde, self.rect)
```

5.5. Función def cargarEnemigos

Esta función tiene como objetivo cargar los enemigos creados a una lista de enemigos,llamada *listaEnemigos*[], es con el fin de saber cuantos enemigos qeudan en la lista y saber cuando el jugador a eliminado a todos los enemigos y puede pasar al siguiente nivel. A continuación una desmostación des código:

```
1   def cargarEnemigos(enemigo):
2       listaEnemigos.append(enemigo)#Carga enemigos en la lista
```

5.6. Función principal: def Juego

En esta función se desarrolla todo lo entorno al juego y se implementan todas las clases y funciones anteriormente mencionadas. La función funciona como programa principal e inicializa todas las variables del juego, excepto las tres variables globales, las cuales son *alto*, *ancho* y *listaEnemigos*[]. La función primero inicializa pygame, carga los sonidos del juego y las imágenes para general la interfaz gráfica de fondo, también declara algunas variables como *jugador, disparo* para asignarles las clases **naveEspacial** y **proyectil** respectivamente, además de dos variables booleanas llamadas enJuego y mouse para poder ver cuando se puede jugar y poder mover el mouse al principio del juego. A continuación se adjunta el código de lo explicado.

```
def Juego():


    pygame.init()
    #Sonido de fondo
    pygame.mixer.init()
    pygame.mixer.music.load('song.mp3')
    pygame.mixer.music.play(-1)

    #Creacion de la centana
    ventana = pygame.display.set_mode((ancho,alto))
    #Nombre de la ventana
    pygame.display.set_caption("Cow Protectors")

    #Imagen de fondo
    Fondo = pygame.image.load("fondo2.png")# cargar imagen de fondo
    Fondo= pygame.transform.scale(Fondo, (1200,600))


    #Imagen de la vaca
    vaca= pygame.image.load("vaca.png")#Importar nave
    vaca= pygame.transform.scale(vaca, (70,40))#modificar tamano nave
    posXvaca,posYvaca=400,452
    #Activacion del mouse
    mouse=True
    #Para bloquear el juego cuando salen fondos
    enJuego= False
    #creamos el objeto de la nave
    jugador = naveEspacial()
    disparo= proyectil(ancho,alto, True)
```

Posterior a esto se inicializan todas la variables para los aliens del juego, se cargar las imagenes de todos los aliens según el tipo que sea se le asigna una clase, se incializan posición velocidad, derecha(permite el movimiento de los aliens), además de las variables para la exploción y muerte de los aliens. Acontinuación la desmotración del código.

```
****Dentro de funci n Juego *****

 #Datos Aliens:
    posXverde,posYverde=0,0
    posXverde2,posYverde2=posXverde+100,posYverde
    posXverde3,posYverde3=posXverde+200,posYverde
    posXverde4,posYverde4=posXverde+300,posYverde
    posXverde5,posYverde5=posXverde+400,posYverde
    posXverde6,posYverde6=0,posYverde-100
```

```
10      posXverde7 , posYverde7=posXverde+100 , posYverde-100
11      posXverde8 , posYverde8=posXverde+200 , posYverde-100
12      posXverde9 , posYverde9=posXverde+300 , posYverde-100
13      posXverde10 , posYverde10=posXverde+400 , posYverde-100
14
15      derecha , derecha2 , derecha3 , derecha4 , derecha5=True , True , True , True , True
16      derecha6 , derecha7 , derecha8 , derecha9 , derecha10=True , True , True , True , True
17
18      velocidad , velocidad2 , velocidad3 , velocidad4 , velocidad5 =1 ,1 ,1 ,1 ,1
19      velocidad6 , velocidad7 , velocidad8 , velocidad9 , velocidad10 =1 ,1 ,1 ,1 ,1
20
21
22      #Cargar aliens que no disparan:
23      alienv=Alien2(posXverde , posYverde , velocidad , "alienverde.png")
24      alienv2=Alien2(posXverde2 , posYverde2 , velocidad2 , "alienrojo.png")
25      alienv3=Alien2(posXverde3 , posYverde3 , velocidad3 , "alienamarillo.png")
26      alienv4=Alien2(posXverde4 , posYverde4 , velocidad4 , "alienrosado.png")
27      alienv5=Alien2(posXverde5 , posYverde5 , velocidad5 , "alienverde.png")
28
29      #Cargar aliens que disparan
30      alienv6=Alienverdedef(posXverde6 , posYverde6 , velocidad6 , "alienrosado.png")
31      alienv7=Alienverdedef(posXverde7 , posYverde7 , velocidad7 , "alienamarillo.png")
32      alienv8=Alienverdedef(posXverde8 , posYverde8 , velocidad8 , "alienverde.png")
33      alienv9=Alienverdedef(posXverde9 , posYverde9 , velocidad9 , "alienrojo.png")
34      alienv10=Alienverdedef(posXverde10 , posYverde10 , velocidad10 , "alienrosado.png")
35
36      #Ver si aliens estan vivos o muertos
37      muertealienv , muertealienv2 , muertealienv3 , muertealienv4 , muertealienv5=False , False ,
            False , False , False
38      muertealienv6 , muertealienv7 , muertealienv8 , muertealienv9 , muertealienv10=False , False ,
            False , False , False
39
40      #Imagen de explosion Alien
41      explosionAlien= pygame.image.load("explosionalien.png")#Importar explosion alien1
42      explosionAlien= pygame.transform.scale(explosionAlien , (300,300))
43      explosion= pygame.image.load("explosion.png")#Importar explosion
44
45      #Tiempo para que se vea la explosion
46      tiempoexplosionalienv , tiempoexplosionalienv2 , tiempoexplosionalienv3 ,
            tiempoexplosionalienv4 , tiempoexplosionalienv5 =0 ,0 ,0 ,0 ,0
47      tiempoexplosionalienv6 , tiempoexplosionalienv7 , tiempoexplosionalienv8 ,
            tiempoexplosionalienv9 , tiempoexplosionalienv10 =0 ,0 ,0 ,0 ,0
48
49      #Copia de coordenada aliens:
50      posXverdecopia , posYverdecopia=-1000,-1000
51      posXverdecopia2 , posYverdecopia2=-1000,-1000
52      posXverdecopia3 , posYverdecopia3=-1000,-1000
53      posXverdecopia4 , posYverdecopia4=-1000,-1000
54      posXverdecopia5 , posYverdecopia5=-1000,-1000
55      posXverdecopia6 , posYverdecopia6=-1000,-1000
56      posXverdecopia7 , posYverdecopia7=-1000,-1000
57      posXverdecopia8 , posYverdecopia8=-1000,-1000
58      posXverdecopia9 , posYverdecopia9=-1000,-1000
59      posXverdecopia10 , posYverdecopia10=-1000,-1000
```

Por último en la declaración se inicializan algunas variables, como las imágenes de portada y subir de nivel, además, variables como *inicio, ganar, perder, subirnivel, contadorNivel, enJuego*, las cuales son variables básicas para el funcionamiento del programa principal. Estas variables booleanas son de gran impor-

tancia debido a que dependiendo de su valor se define la transición entre pantallas, además que se verifica el cambio de nivel, si se esta en la portada u en las otras ventanas.

```
  ****En funcion def Juego*****

#Texto que indica que perdio
    textoFinal=pygame.font.SysFont(None, 50)
    textoFinal=textoFinal.render("Perdiste",0,(200,10,40))

    #Asignacion de vidas
    vidas=3

    #Texto que indica que indica las vidas
    textovida=pygame.font.SysFont(None, (50))
    textovidas=textovida.render("Vidas:"+str(vidas),0,(255,255,255))

    #Indica que esta en portada
    inicio=True

    #cargar portada
    portada= pygame.image.load("portada.png")# cargar imagen de fondo
    portada= pygame.transform.scale(portada, (ancho,alto))

    #cargar imagen de subir nivel
    levelup= pygame.image.load("levelup.png")#
    levelup= pygame.transform.scale(levelup, (ancho,alto))

    #variable booleana para saber si se cambio de nivel
    subirnivel=False

    #contadorNivel:
    contadorNivel=1
    listaEnemigos=[1]
    #Texto que indica el nivel
    textolevel=pygame.font.SysFont(None, (50))
    textonivel=textolevel.render("Nivel:"+str(contadorNivel),0,(255,255,255))

    ganar=False
    #cargar imagen para ganar
    gana= pygame.image.load("ganar.png")#
    ganarimagen= pygame.transform.scale(gana, (ancho,alto))
```

Posterior a la declaración de variables la función juego llama a un while igual a true para hacer un ciclo infinito y se actualice la ventana que muestra el juego.Luego a esto llama a diferentes condicionales para realizar diferentes acciones según se cumplan o no esto, donde en estos condicionales realiza procesos como cargar la imagen de inicio en caso que la variable booleana inicio este activada, caso similar para cuando se presenta para cuando el jugador pierde,gana o sube de nivel. En caso de que ninguna de estas variables booleana este activada, solo enJuego, se procede a la pantalla donde se pueda jugar.

```
****Dentro de funci n Juego *****

while True:


        if inicio==True:
            #Aparece portada
            ventana.blit(portada,(0,0))
        if subirnivel==True:
```

```
10                  ventana.blit(levelup,(0,0))
11                  sonidoLevelup=pygame.mixer.Sound("transicion.wav")
12                  i=1
13                  while i==1:
14                      sonidoLevelup.play()
15                      i+=1
16              if ganar==True:
17
18                  ventana.blit(ganarimagen,(0,0))
19                  sonidoLevelup=pygame.mixer.Sound("victory.wav")
20                  i=1
21                  while i==1:
22                      sonidoLevelup.play()
23                      i+=1
24              if perder==True:
25
26                  ventana.blit(perderimagen,(0,0))
27                  sonidoLevelup=pygame.mixer.Sound("gameover.wav")
28                  i=1
29                  while i==1:
30                      sonidoLevelup.play()
31                      i+=1
32
33              if inicio==False and subirnivel==False and ganar==False and perder==False:
34                  ##Lenamos el fondo
35                  ventana.blit(Fondo,(0,0))
36                  #Colocamos la vaca
37                  ventana.blit(vaca,(posXvaca,posYvaca))
38                  #Colocamos el pasto
39                  pygame.draw.line (ventana,(0,155,0),(0,alto),(ancho,alto),20)
40                  #Se llama al jugador
41                  disparo.trayectoria()
42                  jugador.movimiento()
43                  #Se colocan las posiciones y velocidades de los aliens:
44                  alienv.rect.y,alienv.rect.x,alienv.velocidadAlienverde=posYverde,posXverde,
                        velocidad
45                  alienv2.rect.y,alienv2.rect.x,alienv2.velocidadAlienverde=posYverde2,
                        posXverde2,velocidad2
46                  alienv3.rect.y,alienv3.rect.x,alienv3.velocidadAlienverde=posYverde3,
                        posXverde3,velocidad3
47                  alienv4.rect.y,alienv4.rect.x,alienv4.velocidadAlienverde=posYverde4,
                        posXverde4,velocidad4
48                  alienv5.rect.y,alienv5.rect.x,alienv5.velocidadAlienverde=posYverde5,
                        posXverde5,velocidad5
49                  alienv6.rect.y,alienv6.rect.x,alienv6.velocidadAlienverde=posYverde6,
                        posXverde6,velocidad6
50                  alienv7.rect.y,alienv7.rect.x,alienv7.velocidadAlienverde=posYverde7,
                        posXverde7,velocidad7
51                  alienv8.rect.y,alienv8.rect.x,alienv8.velocidadAlienverde=posYverde8,
                        posXverde8,velocidad8
52                  alienv9.rect.y,alienv9.rect.x,alienv9.velocidadAlienverde=posYverde9,
                        posXverde9,velocidad9
53                  alienv10.rect.y,alienv10.rect.x,alienv10.velocidadAlienverde=posYverde10,
                        posXverde10,velocidad5
54                  #Texto de vidas disponibles
55                  textovidas=textovida.render("Vidas:"+str(vidas),0,(255,255,255))
56                  textonivel=textolevel.render("Nivel:"+str(contadorNivel),0,(255,255,255))
```

Continua seguido por un bucle que recorre todos los eventos para realizar. Se recorre cada evento según las teclas presionadas, como inciar a jugar a jugar con espacio, moverse con el mouse o las teclas, disparar, salirse(con la tecla N), entre otras, además de reiniciar todas la variables según el nivel por el que se vaya, pues algunas variables cambian con el nivel, como la velocidad de los aliens.Esto se logra activando y desactivando las variables booleanas.A continuación el código mencionado.

```
****Dentro de funcion Juego *****
        for evento in pygame.event.get():

            if evento.type == QUIT:

                pygame.quit()

                sys.exit()
            if enJuego==False:
                if inicio==True:
                    if evento.type ==pygame.KEYDOWN:
                        if evento.key==K_SPACE:
                            inicio=False
                            enJuego=True
                if inicio==False and subirnivel==True and ganar==False and perder==False:
                    if evento.type ==pygame.KEYDOWN:
                        if evento.key==K_s:
                            #Pasar ya a pantalla de juego
                            subirnivel=False
                            #Cambiar de nivel
                            contadorNivel+=1
                            #************************Reinicio***************
                            #Volver a poner jugar con el mouse
                            mouse=True
                            #Activar el modo juego
                            enJuego= True
                            #Revivir a los aliens
                            muertealienv,muertealienv2,muertealienv3,muertealienv4,
                                muertealienv5=False,False,False,False,False
                            muertealienv6,muertealienv7,muertealienv8,muertealienv9,
                                muertealienv10=False,False,False,False,False
                            #Resetear las coordenadas de los aliens:
                            posXverde,posYverde=0,0
                            posXverde2,posYverde2=posXverde+100,posYverde
                            posXverde3,posYverde3=posXverde+200,posYverde
                            posXverde4,posYverde4=posXverde+300,posYverde
                            posXverde5,posYverde5=posXverde+400,posYverde
                            posXverde6,posYverde6=0,posYverde-100
                            posXverde7,posYverde7=posXverde+100,posYverde-100
                            posXverde8,posYverde8=posXverde+200,posYverde-100
                            posXverde9,posYverde9=posXverde+300,posYverde-100
                            posXverde10,posYverde10=posXverde+400,posYverde-100
                            #Reiniciar movimiento a la derecha:
                            derecha,derecha2,derecha3,derecha4,derecha5=True,True,True,
                                True,True
                            derecha6,derecha7,derecha8,derecha9,derecha10=True,True,True,
                                True,True
                            #Velocidad de acuerdo al nivel, y lista de enemigos
                            if contadorNivel==1:
                                velocidad,velocidad2,velocidad3,velocidad4,velocidad5
                                    =1,1,1,1,1# velocidad enemigos en nivel1
                                velocidad6,velocidad7,velocidad8,velocidad9,velocidad10
```

```
                =1,1,1,1,1
48              listaEnemigos=[1]
49
50          elif contadorNivel==2:
51              velocidad,velocidad2,velocidad3,velocidad4,velocidad5
                    =2,2,2,2,2# velocidad enemigos en nivel2
52              velocidad6,velocidad7,velocidad8,velocidad9,velocidad10
                    =2,2,2,2,2
53              listaEnemigos=[1,2]
54
55          elif contadorNivel==3:
56              velocidad,velocidad2,velocidad3,velocidad4,velocidad5
                    =3,3,3,3,3# velocidad enemigos en nivel3
57              velocidad6,velocidad7,velocidad8,velocidad9,velocidad10
                    =3,3,3,3,3
58              listaEnemigos=[1,2,3]
59          elif contadorNivel==4:
60              velocidad,velocidad2,velocidad3,velocidad4,velocidad5
                    =4,4,4,4,4# velocidad enemigos en nivel4
61              velocidad6,velocidad7,velocidad8,velocidad9,velocidad10
                    =4,4,4,4,4
62              listaEnemigos=[1,2,3,4]
63          if contadorNivel==5:
64              velocidad,velocidad2,velocidad3,velocidad4,velocidad5
                    =5,5,5,5,5# velocidad enemigos en nivel5
65              velocidad6,velocidad7,velocidad8,velocidad9,velocidad10
                    =5,5,5,5,5
66              listaEnemigos=[1,2,3,4,5]
67          if contadorNivel==6:
68              velocidad,velocidad2,velocidad3,velocidad4,velocidad5
                    =5,5,5,5,5# velocidad enemigos en nivel6
69              velocidad6,velocidad7,velocidad8,velocidad9,velocidad10
                    =5,5,5,5,5
70              listaEnemigos=[1,2,3,4,5,6]
71          if contadorNivel==7:
72              velocidad,velocidad2,velocidad3,velocidad4,velocidad5
                    =5,5,5,5,5# velocidad enemigos en nivel7
73              velocidad6,velocidad7,velocidad8,velocidad9,velocidad10
                    =5,5,5,5,5
74              listaEnemigos=[1,2,3,4,5,6,7]
75          if contadorNivel==8:
76              velocidad,velocidad2,velocidad3,velocidad4,velocidad5
                    =5,5,5,5,5# velocidad enemigos en nivel8
77              velocidad6,velocidad7,velocidad8,velocidad9,velocidad10
                    =5,5,5,5,5
78              listaEnemigos=[1,2,3,4,5,6,7,8]
79          if contadorNivel==9:
80              velocidad,velocidad2,velocidad3,velocidad4,velocidad5
                    =5,5,5,5,5# velocidad enemigos en nivel9
81              velocidad6,velocidad7,velocidad8,velocidad9,velocidad10
                    =5,5,5,5,5
82              listaEnemigos=[1,2,3,4,5,6,7,8,9]
83          if contadorNivel==10:
84              velocidad,velocidad2,velocidad3,velocidad4,velocidad5
                    =5,5,5,5,5# velocidad enemigos en nivel9
85              velocidad6,velocidad7,velocidad8,velocidad9,velocidad10
                    =5,5,5,5,5
86              listaEnemigos=[1,2,3,4,5,6,7,8,9,10]
```

```
87
88                          #Resetear el tiempo de explosion del alien:
89                          tiempoexplosionalienv,tiempoexplosionalienv2,
                                tiempoexplosionalienv3,tiempoexplosionalienv4,
                                tiempoexplosionalienv5=0,0,0,0,0
90                          tiempoexplosionalienv6,tiempoexplosionalienv7,
                                tiempoexplosionalienv8,tiempoexplosionalienv9,
                                tiempoexplosionalienv10=0,0,0,0,0
91
92                  if inicio==False and subirnivel==False and ganar==True and perder==False:
93                      if evento.type==pygame.KEYDOWN:
94                          if evento.key==K_s:
95                              #Pasar ya a pantalla de juego
96                              ganar=False
97                              #Reiniciar desde el nivel 1
98                              contadorNivel=1
99                              #************************Reinicio***************
100                             #Volver a poner jugar con el mouse
101                             mouse=True
102                             #Activar el modo juego
103                             enJuego= True
104                             #Revivir a los aliens
105                             muertealienv,muertealienv2,muertealienv3,muertealienv4,
                                    muertealienv5=False,False,False,False,False
106                             muertealienv6,muertealienv7,muertealienv8,muertealienv9,
                                    muertealienv10=False,False,False,False,False
107                             #Resetear las coordenadas de los aliens:
108                             posXverde,posYverde=0,0
109                             posXverde2,posYverde2=posXverde+100,posYverde
110                             posXverde3,posYverde3=posXverde+200,posYverde
111                             posXverde4,posYverde4=posXverde+300,posYverde
112                             posXverde5,posYverde5=posXverde+400,posYverde
113                             posXverde6,posYverde6=0,posYverde-100
114                             posXverde7,posYverde7=posXverde+100,posYverde-100
115                             posXverde8,posYverde8=posXverde+200,posYverde-100
116                             posXverde9,posYverde9=posXverde+300,posYverde-100
117                             posXverde10,posYverde10=posXverde+400,posYverde-100
118                             #Reiniciar movimiento a la derecha:
119                             derecha,derecha2,derecha3,derecha4,derecha5=True,True,True,
                                    True,True
120                             derecha6,derecha7,derecha8,derecha9,derecha10=True,True,True,
                                    True,True
121                             #Velocidad de acuerdo al nivel, y lista de enemigos
122                             if contadorNivel==1:
123                                 velocidad,velocidad2,velocidad3,velocidad4,velocidad5
                                        =1,1,1,1,1# velocidad enemigos en nivel1
124                                 velocidad6,velocidad7,velocidad8,velocidad9,velocidad10
                                        =1,1,1,1,1
125                                 listaEnemigos=[1]
126                             #Resetear el tiempo de explosion del alien:
127                             tiempoexplosionalienv,tiempoexplosionalienv2,
                                    tiempoexplosionalienv3,tiempoexplosionalienv4,
                                    tiempoexplosionalienv5=0,0,0,0,0
128                             tiempoexplosionalienv6,tiempoexplosionalienv7,
                                    tiempoexplosionalienv8,tiempoexplosionalienv9,
                                    tiempoexplosionalienv10=0,0,0,0,0
129                             vidas=3
130
```

```
131                 if evento.key==K_n:
132                     pygame.quit()
133 
134                     sys.exit()
135 
136 
137             if inicio==False and subirnivel==False and ganar==False and perder==True:
138                 if evento.type ==pygame.KEYDOWN:
139                     if evento.key==K_s:
140 
141                         #Pasar ya a pantalla de juego
142                         perder=False
143                         #Reiniciar desde el nivel 1
144                         contadorNivel=1
145                         #************************Reinicio***************
146                         #Volver a poner jugar con el mouse
147                         mouse=True
148                         #Activar el modo juego
149                         enJuego= True
150                         #Revivir a los aliens
151                         muertealienv,muertealienv2,muertealienv3,muertealienv4,
                                muertealienv5=False,False,False,False,False
152                         muertealienv6,muertealienv7,muertealienv8,muertealienv9,
                                muertealienv10=False,False,False,False,False
153                         #Resetear las coordenadas de los aliens:
154                         posXverde,posYverde=0,0
155                         posXverde2,posYverde2=posXverde+100,posYverde
156                         posXverde3,posYverde3=posXverde+200,posYverde
157                         posXverde4,posYverde4=posXverde+300,posYverde
158                         posXverde5,posYverde5=posXverde+400,posYverde
159                         posXverde6,posYverde6=0,posYverde-100
160                         posXverde7,posYverde7=posXverde+100,posYverde-100
161                         posXverde8,posYverde8=posXverde+200,posYverde-100
162                         posXverde9,posYverde9=posXverde+300,posYverde-100
163                         posXverde10,posYverde10=posXverde+400,posYverde-100
164                         #Reiniciar movimiento a la derecha:
165                         derecha,derecha2,derecha3,derecha4,derecha5=True,True,True,
                                True,True
166                         derecha6,derecha7,derecha8,derecha9,derecha10=True,True,True,
                                True,True
167                         #Velocidad de acuerdo al nivel, y lista de enemigos
168                         if contadorNivel==1:
169                             velocidad,velocidad2,velocidad3,velocidad4,velocidad5
                                    =1,1,1,1,1# velocidad enemigos en nivel1
170                             velocidad6,velocidad7,velocidad8,velocidad9,velocidad10
                                    =1,1,1,1,1
171                             listaEnemigos=[1]
172                         #Resetear el tiempo de explosion del alien:
173                         tiempoexplosionalienv,tiempoexplosionalienv2,
                                tiempoexplosionalienv3,tiempoexplosionalienv4,
                                tiempoexplosionalienv5=0,0,0,0,0
174                         tiempoexplosionalienv6,tiempoexplosionalienv7,
                                tiempoexplosionalienv8,tiempoexplosionalienv9,
                                tiempoexplosionalienv10=0,0,0,0,0
175                         #Restaurar vidas
176                         vidas=3
177 
178                 if evento.key==K_n:
```

```
                    pygame.quit()

                    sys.exit()


            elif enJuego==True:
                if evento.type ==pygame.KEYDOWN:#mover con las teclas la nave
                    if evento.key==K_LEFT:
                        jugador.rect.left-=jugador.velocidad #moverlo si se presiona tecla izquierda
                        mouse=False#si se usa una tecla izquierda ya se desactiva el mouse
                    elif evento.key==K_RIGHT:
                        jugador.rect.right+=jugador.velocidad #mover si se presiona tecla derecha
                        mouse=False#si se usa una tecla derecha ya se desactiva el mouse
                    #    Tecla que define el disparo
                    elif evento.key==K_SPACE :
                        sonidoDisparo=pygame.mixer.Sound("laser2.wav")
                        sonidoDisparo.play()
                        x,y=jugador.rect.center
                        jugador.disparar(x, y)
```

Posterior al for para recorrer los eventos, realizan condicionales para el movimiento de los aliens, empezando con que si aplica en el nivel que aparece se mueva. se realiza para cada alien según el nivel por el que vaya el jugador, pues conforme avanza en los niveles aprecen más, posterior a esto se realiza el llamado al introducir la nave espacial del jugador. A continuación se muestra los condiconales de movimiento para cada alien.

```
****Dentro de funcion Juego *****
        if inicio==False and subirnivel==False and ganar==False:

                if derecha==True:#para que la imagen se vaya moviendo a la derecha
                    if posXverde<730:
                        posXverde+=alienv.velocidadAlienverde
                    else:
                        derecha=False#para que se mueva a la izquierda
                        posYverde+=100#para que en cada cambio se mueva para el frente
                if derecha==False:
                    if posXverde>1:
                        posXverde-=alienv.velocidadAlienverde
                    else:
                        derecha=True
                        posYverde+=100


                if contadorNivel>=2:
                    if derecha2==True:#para que la imagen se vaya moviendo a la derecha
                        if posXverde2<730:
                            posXverde2+=alienv2.velocidadAlienverde
                        else:
                            derecha2=False#para que se mueva a la izquierda
                            posYverde2+=100#para que en cada cambio se mueva para el frente
                    if derecha2==False:
                        if posXverde2>1:
                            posXverde2-=alienv2.velocidadAlienverde
```

```
28                          else:
29                              derecha2=True
30                              posYverde2+=100
31
32                  if contadorNivel>=3:
33                      if derecha3==True:#para que la imagen se vaya moviendo a la derecha
34                          if posXverde3<730:
35                              posXverde3+=alienv3.velocidadAlienverde
36                          else:
37                              derecha3=False#para que se mueva a la izquierda
38                              posYverde3+=100#para que en cada cambio se mueva para el
                                    frente
39                      if derecha3==False:
40                          if posXverde3>1:
41                              posXverde3-=alienv3.velocidadAlienverde
42                          else:
43                              derecha3=True
44                              posYverde3+=100
45
46
47                  if contadorNivel>=4:
48                      if derecha4==True:#para que la imagen se vaya moviendo a la derecha
49                          if posXverde4<730:
50                              posXverde4+=alienv4.velocidadAlienverde
51                          else:
52                              derecha4=False#para que se mueva a la izquierda
53                              posYverde4+=100#para que en cada cambio se mueva para el
                                    frente
54                      if derecha4==False:
55                          if posXverde4>1:
56                              posXverde4-=alienv4.velocidadAlienverde
57                          else:
58                              derecha4=True
59                              posYverde4+=100
60                  if contadorNivel>=5:
61                      if derecha5==True:#para que la imagen se vaya moviendo a la derecha
62                          if posXverde5<730:
63                              posXverde5+=alienv5.velocidadAlienverde
64                          else:
65                              derecha5=False#para que se mueva a la izquierda
66                              posYverde5+=100#para que en cada cambio se mueva para el
                                    frente
67                      if derecha5==False:
68                          if posXverde5>1:
69                              posXverde5-=alienv5.velocidadAlienverde
70                          else:
71                              derecha5=True
72                              posYverde5+=100
73                  if contadorNivel>=6:
74                      if derecha6==True:#para que la imagen se vaya moviendo a la derecha
75                          if posXverde6<730:
76                              posXverde6+=alienv6.velocidadAlienverde
77                          else:
78                              derecha6=False#para que se mueva a la izquierda
79                              posYverde6+=100#para que en cada cambio se mueva para el
                                    frente
80                      if derecha6==False:
81                          if posXverde6>1:
```

```
82                          posXverde6-=alienv6.velocidadAlienverde
83                      else:
84                          derecha6=True
85                          posYverde6+=100
86              if contadorNivel>=7:
87                  if derecha7==True:#para que la imagen se vaya moviendo a la derecha
88                      if posXverde7<730:
89                          posXverde7+=alienv7.velocidadAlienverde
90                      else:
91                          derecha7=False#para que se mueva a la izquierda
92                          posYverde7+=100#para que en cada cambio se mueva para el
                                frente
93                  if derecha7==False:
94                      if posXverde7>1:
95                          posXverde7-=alienv7.velocidadAlienverde
96                      else:
97                          derecha7=True
98                          posYverde7+=100
99
100             if contadorNivel>=8:
101                 if derecha8==True:#para que la imagen se vaya moviendo a la derecha
102                     if posXverde8<730:
103                         posXverde8+=alienv8.velocidadAlienverde
104                     else:
105                         derecha8=False#para que se mueva a la izquierda
106                         posYverde8+=100#para que en cada cambio se mueva para el
                                frente
107                 if derecha8==False:
108                     if posXverde8>1:
109                         posXverde8-=alienv8.velocidadAlienverde
110                     else:
111                         derecha8=True
112                         posYverde8+=100
113
114             if contadorNivel>=9:
115                 if derecha9==True:#para que la imagen se vaya moviendo a la derecha
116                     if posXverde9<730:
117                         posXverde9+=alienv9.velocidadAlienverde
118                     else:
119                         derecha9=False#para que se mueva a la izquierda
120                         posYverde9+=100#para que en cada cambio se mueva para el
                                frente
121                 if derecha9==False:
122                     if posXverde9>1:
123                         posXverde9-=alienv9.velocidadAlienverde
124                     else:
125                         derecha9=True
126                         posYverde9+=100
127
128             if contadorNivel>=10:
129                 if derecha10==True:#para que la imagen se vaya moviendo a la derecha
130                     if posXverde10<730:
131                         posXverde10+=alienv10.velocidadAlienverde
132                     else:
133                         derecha10=False#para que se mueva a la izquierda
134                         posYverde10+=100#para que en cada cambio se mueva para el
                                frente
135                 if derecha10==False:
```

```
                    if posXverde10>1:
                        posXverde10-=alienv10.velocidadAlienverde
                    else:
                        derecha10=True
                        posYverde10+=100

                #Se dibuja a la nave
                jugador.dibujar(ventana)
```

Como continuación se muestra un condicional para los niveles mayores o iguales a 6 en los cuales los aliens disparan reiniciando en caso de que el jugador pierda una vida, reinicia todas las variables según para el nivel por donde vaya el nivel y cargando los disparos de los aliens, este proceso se realiza para el contador nivel mayor a 6 en adelante, el proceso se realiza para cada número mayor hasta 10. A continuación se muestra el código comple del if cuando es mayor a 6 y la declaración de los demás los cuales tendrían el código equivalente.

```
****Dentro de funcion Juego *****
                #AQUI VAn solo los aliens que disparan
                if contadorNivel>=6:
                    if len(alienv6.listaDisparos)>0:
                        for x in alienv6.listaDisparos:

                            x.dibujar(ventana)
                            x.trayectoria()
                            if x.rect.colliderect(jugador.rect):
                                if vidas<=1:
                                    alienv6.listaDisparos.remove(x)
                                    mouse=False
                                    jugador.velocidad=0
                                    velocidad,velocidad2,velocidad3,velocidad4,velocidad5
                                        =0,0,0,0,0# velocidad enemigos en nivel1
                                    velocidad6,velocidad7,velocidad8,velocidad9,
                                        velocidad10=0,0,0,0,0
                                    vidas-=1
                                elif vidas>1:
                                    alienv6.listaDisparos.remove(x)
                                    #************************Reinicio***************
                                    #Volver a poner jugar con el mouse
                                    mouse=True
                                    #Activar el modo juego
                                    enJuego= True
                                    #Revivir a los aliens
                                    muertealienv,muertealienv2,muertealienv3,
                                        muertealienv4,muertealienv5=False,False,False,
                                        False,False
                                    muertealienv6,muertealienv7,muertealienv8,
                                        muertealienv9,muertealienv10=False,False,False,
                                        False,False
                                    #Resetear las coordenadas de los aliens:
                                    posXverde,posYverde=0,0
                                    posXverde2,posYverde2=posXverde+100,posYverde
                                    posXverde3,posYverde3=posXverde+200,posYverde
                                    posXverde4,posYverde4=posXverde+300,posYverde
                                    posXverde5,posYverde5=posXverde+400,posYverde
                                    posXverde6,posYverde6=0,posYverde-100
                                    posXverde7,posYverde7=posXverde+100,posYverde-100
                                    posXverde8,posYverde8=posXverde+200,posYverde-100
                                    posXverde9,posYverde9=posXverde+300,posYverde-100
```

```
37                  posXverde10, posYverde10=posXverde+400, posYverde-100
38                  #Reiniciar movimiento a la derecha:
39                  derecha, derecha2, derecha3, derecha4, derecha5=True, True
                        , True, True, True
40                  derecha6, derecha7, derecha8, derecha9, derecha10=True,
                        True, True, True, True
41                  #Velocidad de acuerdo al nivel, y lista de enemigos
42                  if contadorNivel==1:
43                      velocidad, velocidad2, velocidad3, velocidad4,
                            velocidad5=1,1,1,1,1# velocidad enemigos en
                            nivel1
44                      velocidad6, velocidad7, velocidad8, velocidad9,
                            velocidad10=1,1,1,1,1
45                      listaEnemigos=[1]
46
47                  elif contadorNivel==2:
48                      velocidad, velocidad2, velocidad3, velocidad4,
                            velocidad5=2,2,2,2,2# velocidad enemigos en
                            nivel2
49                      velocidad6, velocidad7, velocidad8, velocidad9,
                            velocidad10=2,2,2,2,2
50                      listaEnemigos=[1,2]
51
52                  elif contadorNivel==3:
53                      velocidad, velocidad2, velocidad3, velocidad4,
                            velocidad5=3,3,3,3,3# velocidad enemigos en
                            nivel3
54                      velocidad6, velocidad7, velocidad8, velocidad9,
                            velocidad10=3,3,3,3,3
55                      listaEnemigos=[1,2,3]
56                  elif contadorNivel==4:
57                      velocidad, velocidad2, velocidad3, velocidad4,
                            velocidad5=4,4,4,4,4# velocidad enemigos en
                            nivel4
58                      velocidad6, velocidad7, velocidad8, velocidad9,
                            velocidad10=4,4,4,4,4
59                      listaEnemigos=[1,2,3,4]
60                  if contadorNivel==5:
61                      velocidad, velocidad2, velocidad3, velocidad4,
                            velocidad5=5,5,5,5,5# velocidad enemigos en
                            nivel5
62                      velocidad6, velocidad7, velocidad8, velocidad9,
                            velocidad10=5,5,5,5,5
63                      listaEnemigos=[1,2,3,4,5]
64                  if contadorNivel==6:
65                      velocidad, velocidad2, velocidad3, velocidad4,
                            velocidad5=5,5,5,5,5# velocidad enemigos en
                            nivel6
66                      velocidad6, velocidad7, velocidad8, velocidad9,
                            velocidad10=5,5,5,5,5
67                      listaEnemigos=[1,2,3,4,5,6]
68                  if contadorNivel==7:
69                      velocidad, velocidad2, velocidad3, velocidad4,
                            velocidad5=5,5,5,5,5# velocidad enemigos en
                            nivel7
70                      velocidad6, velocidad7, velocidad8, velocidad9,
                            velocidad10=5,5,5,5,5
71                      listaEnemigos=[1,2,3,4,5,6,7]
```

```
72                                  if contadorNivel==8:
73                                      velocidad,velocidad2,velocidad3,velocidad4,
                                            velocidad5=5,5,5,5,5# velocidad enemigos en
                                            nivel8
74                                      velocidad6,velocidad7,velocidad8,velocidad9,
                                            velocidad10=5,5,5,5,5
75                                      listaEnemigos=[1,2,3,4,5,6,7,8]
76                                  if contadorNivel==9:
77                                      velocidad,velocidad2,velocidad3,velocidad4,
                                            velocidad5=5,5,5,5,5# velocidad enemigos en
                                            nivel9
78                                      velocidad6,velocidad7,velocidad8,velocidad9,
                                            velocidad10=5,5,5,5,5
79                                      listaEnemigos=[1,2,3,4,5,6,7,8,9]
80                                  if contadorNivel==10:
81                                      velocidad,velocidad2,velocidad3,velocidad4,
                                            velocidad5=5,5,5,5,5# velocidad enemigos en
                                            nivel9
82                                      velocidad6,velocidad7,velocidad8,velocidad9,
                                            velocidad10=5,5,5,5,5
83                                      listaEnemigos=[1,2,3,4,5,6,7,8,9,10]
84
85                                  #Resetear el tiempo de explosion del alien:
86                                  tiempoexplosionalienv,tiempoexplosionalienv2,
                                        tiempoexplosionalienv3,tiempoexplosionalienv4,
                                        tiempoexplosionalienv5=0,0,0,0,0
87                                  tiempoexplosionalienv6,tiempoexplosionalienv7,
                                        tiempoexplosionalienv8,tiempoexplosionalienv9,
                                        tiempoexplosionalienv10=0,0,0,0,0
88                                  #Se resta una vida
89                                  vidas-=1
90
91
92                              if x.rect.top>505:
93                                  alienv6.listaDisparos.remove(x)
94
95                  if contadorNivel>=7:
96                      ***********
97                      #Respectivo codigo
98                      ***********
99                  if contadorNivel>=8:
100                     ***********
101                     #Respectivo codigo
102                     *******
103                 if contadorNivel>=9:
104                     ***********
105                     #Respectivo codigo
106                     ***********
107                 if contadorNivel>=10:
108                     ***********
109                     #Respectivo codigo
110                     ***********
```

Posteriormente se sigue con condicionales para cada alien según sea el nivel, si el alien no fue eliminado se dibuje, si alien fue eliminado reproduzca la explosión, además, de esto se llama a un else del if principal anterior para reiniciar el nivel y todas las variables según el nivel que sea. Acontinuación se presenta la escrtitura o muerte de los primeros 3 alien y la inicialización de los demás y el else, pues es basicamente el

mismo proceso.

```
****Dentro de funcion Juego *****
                if muertealienv==False:
                    alienv.dibujar(ventana)
                if muertealienv==True and tiempoexplosionalienv<10:
                    if posXverde!=-1000 and posYverde!=-1000:
                        posXverdecopia,posYverdecopia=posXverde,posYverde
                    ventana.blit(explosionAlien,(posXverdecopia-100,posYverdecopia-100))
                    tiempoexplosionalienv+=1
                    posXverde,posYverde=-1000,-1000

                if contadorNivel>=2:
                    if muertealienv2==False:
                        alienv2.dibujar(ventana)
                    if muertealienv2==True and tiempoexplosionalienv2<10:
                        if posXverde2!=-1000 and posYverde2!=-1000:
                            posXverdecopia2,posYverdecopia2=posXverde2,posYverde2
                        ventana.blit(explosionAlien,(posXverdecopia2-100,posYverdecopia2
                            -100))
                        tiempoexplosionalienv2+=1
                        posXverde2,posYverde2=-1000,-1000

                if contadorNivel>=3:
                    if muertealienv3==False:
                        alienv3.dibujar(ventana)
                    if muertealienv3==True and tiempoexplosionalienv3<10:
                        if posXverde3!=-1000 and posYverde3!=-1000:
                            posXverdecopia3,posYverdecopia3=posXverde3,posYverde3
                        ventana.blit(explosionAlien,(posXverdecopia3-100,posYverdecopia3
                            -100))
                        tiempoexplosionalienv3+=1
                        posXverde3,posYverde3=-1000,-1000

                ***********Sigue respectivamente para cada n mero de 4 a 10**********
                if contadorNivel>=4:
                **********
                Codigo respectivo
                ***********
                if contadorNivel>=5:
                **********
                Codigo respectivo
                ***********
                if contadorNivel>=6:
                **********
                Codigo respectivo
                ***********
                if contadorNivel>=7:
                **********
                Codigo respectivo
                ***********
                if contadorNivel>=8:
                **********
                Codigo respectivo
                ***********
                if contadorNivel>=9:
                **********
                Codigo respectivo
                ***********
```

```
                if contadorNivel>=10:
                **********
                Codigo respectivo
                ***********
                else:
                    ***********
                    respectivo c digo de reinicio de variables en el else
                    ********
```

Posterior a esto se revisa por medio de condicionales si el jugador chocó con algún alien, si se cumple y tiene una vida, que se le reste y haga que los aliens se detengan y el jugador pierda, si tiene más de una vida, que se reinicie el nivel por el cual va, realizando una reinicio de todas las variables de los aliens, según el nivel por cual se vaya, se realiza para cada nivel sucesivamente la comparación si el cada alien chocó con la nave. A continuación se muestra el código realizado para el nivel 1 y se mencionan la incialiación para los demás niveles

```
****Dentro de funcion Juego *****
                if jugador.rect.colliderect(alienv.rect):
                    if vidas<=1:
                        mouse=False
                        jugador.velocidad=0
                        velocidad,velocidad2,velocidad3,velocidad4,velocidad5=0,0,0,0,0#
                            velocidad enemigos en nivel1
                        velocidad6,velocidad7,velocidad8,velocidad9,velocidad10=0,0,0,0,0
                        vidas-=1

                    else:
                        #************************Reinicio***************
                        #Volver a poner jugar con el mouse
                        mouse=True
                        #Activar el modo juego
                        enJuego= True
                        #Revivir a los aliens
                        muertealienv,muertealienv2,muertealienv3,muertealienv4,
                            muertealienv5=False,False,False,False,False
                        muertealienv6,muertealienv7,muertealienv8,muertealienv9,
                            muertealienv10=False,False,False,False,False
                        #Resetear las coordenadas de los aliens:
                        posXverde,posYverde=0,0
                        posXverde2,posYverde2=posXverde+100,posYverde
                        posXverde3,posYverde3=posXverde+200,posYverde
                        posXverde4,posYverde4=posXverde+300,posYverde
                        posXverde5,posYverde5=posXverde+400,posYverde
                        posXverde6,posYverde6=0,posYverde+100
                        posXverde7,posYverde7=posXverde+100,posYverde+100
                        posXverde8,posYverde8=posXverde+200,posYverde+100
                        posXverde9,posYverde9=posXverde+300,posYverde+100
                        posXverde10,posYverde10=posXverde+400,posYverde+100
                        #Reiniciar movimiento a la derecha:
                        derecha,derecha2,derecha3,derecha4,derecha5=True,True,True,True,
                            True
                        derecha6,derecha7,derecha8,derecha9,derecha10=True,True,True,True
                            ,True
                        #Velocidad de acuerdo al nivel, y lista de enemigos
                        if contadorNivel==1:
                                velocidad,velocidad2,velocidad3,velocidad4,velocidad5
                                    =1,1,1,1,1# velocidad enemigos en nivel1
```

```
36                  velocidad6 , velocidad7 , velocidad8 , velocidad9 , velocidad10
                        =1,1,1,1,1
37                  listaEnemigos=[1]
38
39              elif contadorNivel==2:
40                  velocidad , velocidad2 , velocidad3 , velocidad4 , velocidad5
                        =2,2,2,2,2# velocidad enemigos en nivel2
41                  velocidad6 , velocidad7 , velocidad8 , velocidad9 , velocidad10
                        =2,2,2,2,2
42                  listaEnemigos=[1,2]
43
44              elif contadorNivel==3:
45                  velocidad , velocidad2 , velocidad3 , velocidad4 , velocidad5
                        =3,3,3,3,3# velocidad enemigos en nivel3
46                  velocidad6 , velocidad7 , velocidad8 , velocidad9 , velocidad10
                        =3,3,3,3,3
47                  listaEnemigos=[1,2,3]
48              elif contadorNivel==4:
49                  velocidad , velocidad2 , velocidad3 , velocidad4 , velocidad5
                        =4,4,4,4,4# velocidad enemigos en nivel4
50                  velocidad6 , velocidad7 , velocidad8 , velocidad9 , velocidad10
                        =4,4,4,4,4
51                  listaEnemigos=[1,2,3,4]
52              if contadorNivel==5:
53                  velocidad , velocidad2 , velocidad3 , velocidad4 , velocidad5
                        =5,5,5,5,5# velocidad enemigos en nivel5
54                  velocidad6 , velocidad7 , velocidad8 , velocidad9 , velocidad10
                        =5,5,5,5,5
55                  listaEnemigos=[1,2,3,4,5]
56              if contadorNivel==6:
57                  velocidad , velocidad2 , velocidad3 , velocidad4 , velocidad5
                        =6,6,6,6,6# velocidad enemigos en nivel6
58                  velocidad6 , velocidad7 , velocidad8 , velocidad9 , velocidad10
                        =6,6,6,6,6
59                  listaEnemigos=[1,2,3,4,5,6]
60              if contadorNivel==7:
61                  velocidad , velocidad2 , velocidad3 , velocidad4 , velocidad5
                        =7,7,7,7,7# velocidad enemigos en nivel7
62                  velocidad6 , velocidad7 , velocidad8 , velocidad9 , velocidad10
                        =7,7,7,7,7
63                  listaEnemigos=[1,2,3,4,5,6,7]
64              if contadorNivel==8:
65                  velocidad , velocidad2 , velocidad3 , velocidad4 , velocidad5
                        =8,8,8,8,8# velocidad enemigos en nivel8
66                  velocidad6 , velocidad7 , velocidad8 , velocidad9 , velocidad10
                        =8,8,8,8,8
67                  listaEnemigos=[1,2,3,4,5,6,7,8]
68              if contadorNivel==9:
69                  velocidad , velocidad2 , velocidad3 , velocidad4 , velocidad5
                        =9,9,9,9,9# velocidad enemigos en nivel9
70                  velocidad6 , velocidad7 , velocidad8 , velocidad9 , velocidad10
                        =9,9,9,9,9
71                  listaEnemigos=[1,2,3,4,5,6,7,8,9]
72              if contadorNivel==10:
73                  velocidad , velocidad2 , velocidad3 , velocidad4 , velocidad5
                        =10,10,10,10,10# velocidad enemigos en nivel9
74                  velocidad6 , velocidad7 , velocidad8 , velocidad9 , velocidad10
                        =10,10,10,10,10
```

```
                    listaEnemigos=[1,2,3,4,5,6,7,8,9,10]

                #Resetear el tiempo de explosion del alien:
                tiempoexplosionalienv,tiempoexplosionalienv2,
                    tiempoexplosionalienv3,tiempoexplosionalienv4,
                    tiempoexplosionalienv5=0,0,0,0,0
                tiempoexplosionalienv6,tiempoexplosionalienv7,
                    tiempoexplosionalienv8,tiempoexplosionalienv9,
                    tiempoexplosionalienv10=0,0,0,0,0
                #Se resta una vida
                vidas-=1

            ******** Se repite un equivalente para los siguientes de 2 hasta
                10**********

        if contadorNivel>=2:
        **********
        codigo respectivo
        **********
        if contadorNivel>=3:
        **********
        codigo respectivo
        **********
        if contadorNivel>=4:
        **********
        codigo respectivo
        **********
        if contadorNivel>=5:
        **********
        codigo respectivo
        **********
        if contadorNivel>=6:
        **********
        codigo respectivo
        if contadorNivel>=7:
        **********
        codigo respectivo
        **********
        if contadorNivel>=8:
        **********
        codigo respectivo
        **********
        if contadorNivel>=9:
        **********
        codigo respectivo
        **********
        if contadorNivel>=10:
        **********
        codigo respectivo
        **********
```

Por último para terminar el código de la función principal Juego, se compara mediante un condicional la lista de disparos del jugador y se dibujan los disparos, además se verifica si los disparos coinciden con alguno los aliens, y si lo hace, marcar la variable muerte de los aliens como True. para poder eliminarlos y hacerlos explotar. Posterior a esta comparación se realizan tres últimos condicionales, uno para saber si ganó, otro para saber si perdió y otro para saber si pasó el nivel; si perdió se detenie el juego y se marca como $perder = True$, si gana se señala como $ganar = True$ y si avanza de nivel $subirNivel = True$. A

continuación una demostación del código explicado

```
                if len(jugador.listaDisparo)>0:
                    for x in jugador.listaDisparo:

                        x.dibujar(ventana)
                        x.trayectoria()

                        if x.rect.top<-1000:

                            jugador.listaDisparo.remove(x)
                        if alienv.rect.colliderect(x.rect):
                                jugador.listaDisparo.remove(x)
                                muertealienv=True
                                if len(listaEnemigos)>0:
                                    listaEnemigos.pop()

                        if contadorNivel>=2:
                            if alienv2.rect.colliderect(x.rect):
                                    jugador.listaDisparo.remove(x)
                                    muertealienv2=True
                                    if len(listaEnemigos)>0:
                                        listaEnemigos.pop()

                        if contadorNivel>=3:
                            if alienv3.rect.colliderect(x.rect):
                                    jugador.listaDisparo.remove(x)
                                    muertealienv3=True
                                    if len(listaEnemigos)>0:
                                        listaEnemigos.pop()
                        if contadorNivel>=4:
                            if alienv4.rect.colliderect(x.rect):
                                    jugador.listaDisparo.remove(x)
                                    muertealienv4=True
                                    if len(listaEnemigos)>0:
                                        listaEnemigos.pop()

                        if contadorNivel>=5:
                            if alienv5.rect.colliderect(x.rect):
                                    jugador.listaDisparo.remove(x)
                                    muertealienv5=True
                                    if len(listaEnemigos)>0:
                                        listaEnemigos.pop()

                        if contadorNivel>=6:
                            if alienv6.rect.colliderect(x.rect):
                                    jugador.listaDisparo.remove(x)
                                    muertealienv6=True
                                    if len(listaEnemigos)>0:
                                        listaEnemigos.pop()

                        if contadorNivel>=7:
                            if alienv7.rect.colliderect(x.rect):
                                    jugador.listaDisparo.remove(x)
                                    muertealienv7=True
                                    if len(listaEnemigos)>0:
                                        listaEnemigos.pop()

                        if contadorNivel>=8:
```

```
                            if alienv8.rect.colliderect(x.rect):
                                    jugador.listaDisparo.remove(x)
                                    muertealienv8=True
                                    if len(listaEnemigos)>0:
                                        listaEnemigos.pop()

                        if contadorNivel>=9:
                            if alienv9.rect.colliderect(x.rect):
                                    jugador.listaDisparo.remove(x)
                                    muertealienv9=True
                                    if len(listaEnemigos)>0:
                                        listaEnemigos.pop()

                        if contadorNivel>=10:
                            if alienv10.rect.colliderect(x.rect):
                                    jugador.listaDisparo.remove(x)
                                    muertealienv10=True
                                    if len(listaEnemigos)>0:
                                        listaEnemigos.pop()



                #Muestra nivel
                ventana.blit(textonivel,(200,alto-50))
                if vidas>=1:
                    #Muestra vidas
                    ventana.blit(textovidas,(50,alto-50))
            if vidas<1:
                enJuego=False
                subirnivel=False
                inicio=False
                perder=True

            if len(listaEnemigos)<=0:
                subirnivel=True
                inicio=False
                enJuego=False

            if contadorNivel>=11:
                enJuego=False
                subirnivel=False
                inicio=False
                ganar=True
```

A continuacion se muestran unas imagenes de la funcionalidad del codigo, tales como cuando se pierde, se gana, cambio de nivel y al momento de jugar.

Figura 14: Pantalla principal al iniciar el juego

Figura 15: Nivel 5 del juego

Figura 16: Nivel 8 del juego

Figura 17: Pantalla que aparece al cambiar de nivel

Figura 18: Pantalla que aparece al quedarse sin vidas

Figura 19: Pantalla que aparece si se pasan los 10 niveles

6. Conclusiones

- Se aprendió durante la investigación del proyecto sobre la librería Pygame de python para la implementación de esta en el desarrollo de todo el programa.

- Se logró mediante la librería Pygame de python realizar una interfaz gráfica, diseñando los aliens y cargando imagénes de fondo, además cargar la nave espacial para el jugador.

- Se obtuvo el movimiento de los aliens mediante la implementación en la función**def Juego**. Además en esta misma función se implementó el movimiento de la nave espacial por teclado y mouse.
  ‘

- Se agregó a la clase **naveEspacial** la posibilidad de realizar disparos apretando la tecla *espacio*. Para con esto se pudieran eliminar aliens y avanzar de nivel.

- Se incluyó a la clase **alien2** la posibilidad de disparar, mediante una función aleatoria, cuando se mantuviera en el rango asignado(Variable rango) disparara y si los rayos disparados coincidieran con la nave, esta perdiera una vida y si el jugador se queda sin vidas pierda.

- Se realizó la transición de niveles, con pantallas de por medio, permitiendo que el usuario al presionar la tecla *espacio* empezara el siguiente nivel.

- Se aumentó la dificultad de los niveles conforme el jugador avanza en eljuego aumentando la velocidad,la cantidad y la habilidad (disparar) de los aliens, llegando a un tope de 10 niveles.

- Se realizó una portada de juego, permitiendo un menú que le permitiera al jugador al presionar la tecla *espacio* comenzar el juego.

- Se habilitó la opción para el usuario que al perder pudiera volver al menú para empezar a jugar de nuevo.

‘

6.1. Trabajo futuro y Recomendaciones

Durante el desarrollo del proyecto se aprendieron muchos conocimientos en la librería Pygame y en general en el idioma de programación python, esto nos permite para trabajos futuros el desarrollo de múltiples videojuegos de arcade basados en la misma lógica de creación, además de utilizar estos conocimientos como herramientas a la hora de desarrollar programas con interfaz gráfica. Con recomendaciones se pueden sugerir las siguientes:

- Investigar e informarse acerca de todas las herramientas que pueden servir para la realización del planeamiento para el proyecto.Donde se tome en cuenta con los recursos con los que se cuentan para el desarrollo correcto del proyecto, esto como las distintas bibliotecas para trabajar con interfaz gráfica como lo es pygame, además de investigar como realizar su uso adecuado implementando clases y funciones para su óptimo funcionamiento.

- Establecer una buena comunicación en el grupo de trabajo, pues al trabajar con un programas de código debe haber una buena división de tareas para que los integrantes no realicen las mismas tareas, además que todos participantes logren entender el código de algún integrante.
- Establecer metas realistas, tomando en cuenta el tiempo con el que se cuenta, la diversidad de herramientas y la cantidad de información con la que se dispone.
- Realizar pequeñas reuniones de trabajo para discutir resultados del proyecto y que se apoyen entre los diferentes miembros del grupo con sus tareas y realizar soluciones más rápidas y eficientes.
- Realizar un planeamiento crítico para de esta manera separar las tareas indispensables para el funcionamiento del programa de las tareas secundarias para mejorar este.Esto para empezar por las indispensables y dejar al final las secundarias, así en caso de falta de tiempo no es tan grave la pérdida de una tarea secundaria.

Referencias

[1] Codigofacilito. Configuración de python y pygame. Online, 22 de setiembre del 2014. Canal de youtube en: https://www.youtube.com/watch?v=PGRhWuYjPdwlist=PLpOqH6AE0tNherBf6bzGiDM1uIy$_E$0WJH.

[2] Square Enix Games. Space invadors / legacy. url https://spaceinvaders.square-enix-games.com/legacy, 2020. Accedido 20-05-2020.

[3] B Lewandoski. Sapce invader project. Online, 13 de noviembre. Obtenido de: https://stackoverflow.com/questions/19966094/space-invaders-project.