

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

CAMPUS POLITÉCNICO “ING. JOSÉ RUBÉN ORELLANA”

**PROGRAMACION AVANZADA**

**PROYECTO FINAL**

Juan Collaguazo

Omar Espin

Fernando San Martin

1. INTRODUCCION

En este documento lo que vamos a proceder a realizar es el proyecto de fin de semestre el cual es orientado a la aplicación de los conocimientos adquiridos en la materia de Programación Avanzada

1. PROPOSITO DE LA PRACTICA

Consiste en el desarrollo de juego tipo arcade en Python

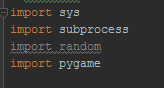
1. OBGETIVO GENERAL

Desarrollar un juego tipo arcade en Python utilizando todos los conocimientos y aplicaciones que hemos adquirido en la materia

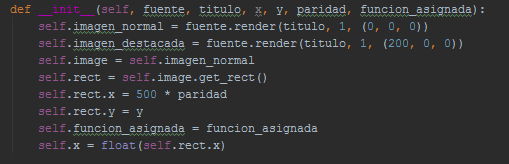
1. DESARROLLO DE ACONTENIDO

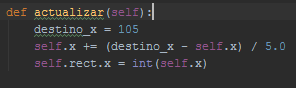
* Para el **desarrollo del juego** se crea un nuevo proyecto en Python el cual contendrá los archivos que sean necesarios para la creación del juego

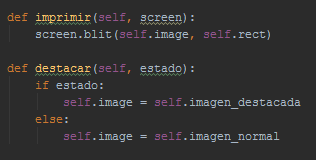
En la clase “menú\_merojado.py” utilizamos ciertas librerías para que nuestro código ejecute de manera correcta



En esta clase nosotros creamos el menú del juego, primero definimos funciones que va a permitir que funcione el juego





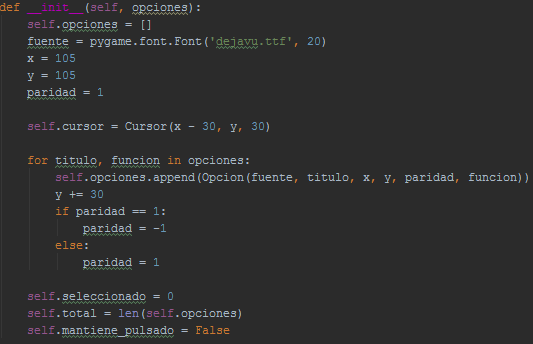




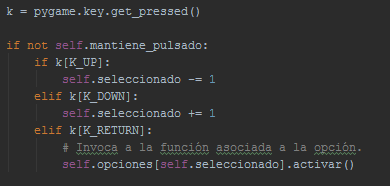
Sin estas funciones nosotros podríamos tener algún problema al momento de ejecutar el programa

Al igual definimos funciones para crear el cursor, para su actualización en el juego, al momento de seleccionar la opción que desea en el menú, y la otra función de imprimir es decir el que va a elegir

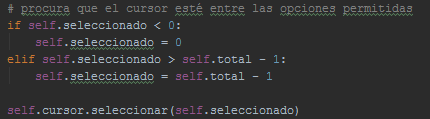
Creamos una clase para crear el Menú el cual va a representar las opciones para el juego



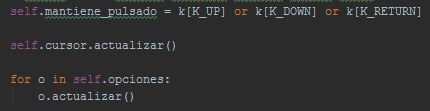
En la función de actualizar es el que va alterar el valor del “self.seleccionado” con las direccionales del cursor



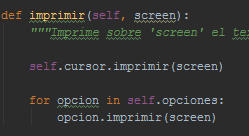
Una cosa importante en el menú de juego es realizar condiciones las cuales nos van a servir para el cursor el procura que este entre las opciones permitidas



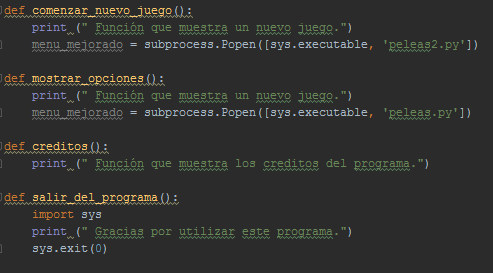
Otra condición que me va indicar si el usuario mantiene pulsada alguna tecla



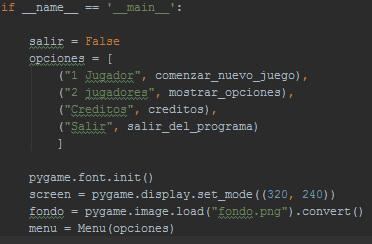
Para comprobar que se escoge cualquier opción del menú definimos una función con el nombre imprimir el cual trata de imprimir sobre el screen el texto de cada opción del menú



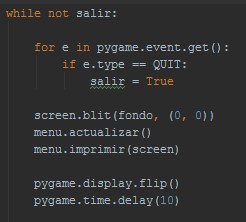
Estas funciones que creamos van a llamar a las otras clases que son las de peleas, una para jugar contra el usuario, y la segunda juego para dos, en créditos que nos muestre los créditos del programa y la otra función salir del programa



Con esta condición verificamos si ya se escogió alguna opción del menú, y un while para la salida del juego en caso que el usuario ya desee salir del juego



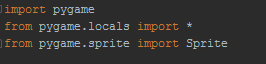
Con “while”



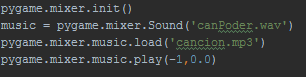
* Como **segundo punto** vamos a crear la clase de “peleas.py” el cual contiene el código de la ejecución para un solo jugador

Como podemos observar

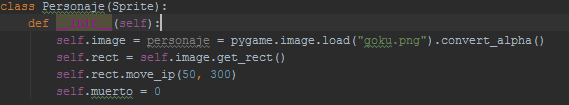
Utilizamos las librerías



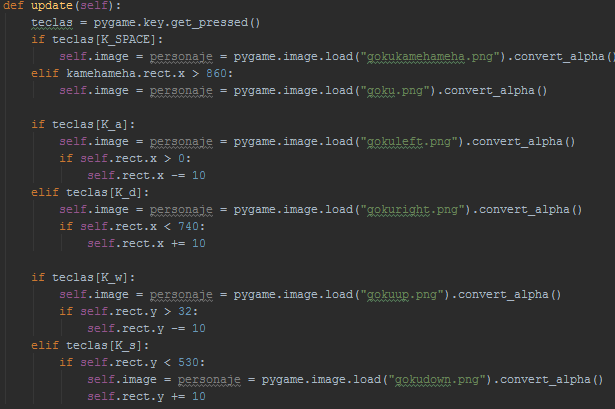
Añadimos música a nuestro juego con el siguiente código también verificamos que la música que se va emplear su tipo sea conveniente, aquí le añadimos dos canciones, para los poderes y la segunda para el fondo del juego



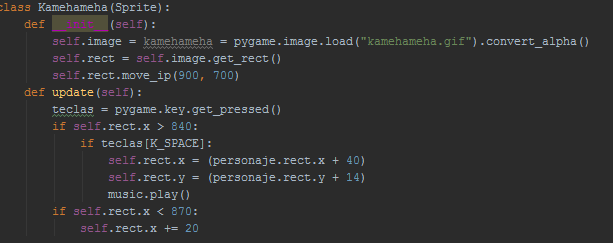
Dentro de la clase “peleas.py” vamos a crear los personajes, es decir los objetos del juego, primero creamos el personaje que es “goku”, y añadimos la imagen y podemos la dirección que sea adecuada



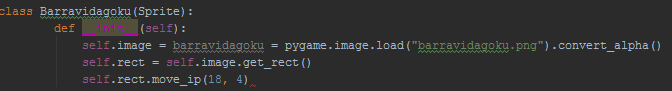
Dentro de la misma clase de personaje definimos otra función que sirve para el movimiento, de arriba, abajo, izquierda, derecha con sus respectivas imágenes



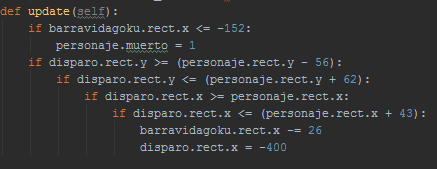
Creamos una clase para objeto de “kamehameha” le agregamos la imagen correspondiente y su posición, y dentro del movimiento del objeto le llamamos a la función de la música



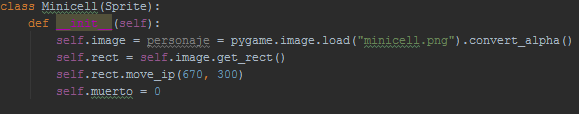
Se crea otra clase para la “barra de la vida de goku” definimos una función para la creación con su imagen respectiva y otra función para el cálculo del daño que le proporcione el adversario



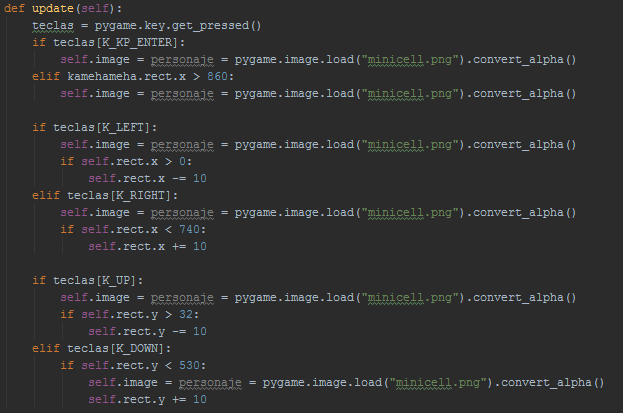
Calculo de vida



Creamos otra clase para el “minicell” que viene hacer el adversario, definimos una función para crear al objeto con la imagen respectiva y otra función el movimiento

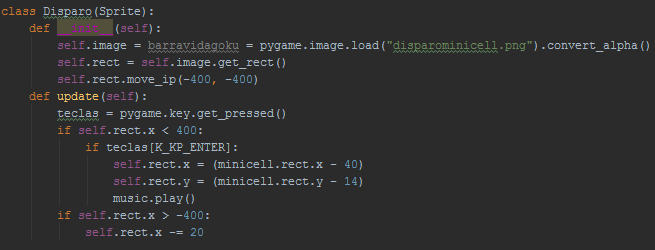


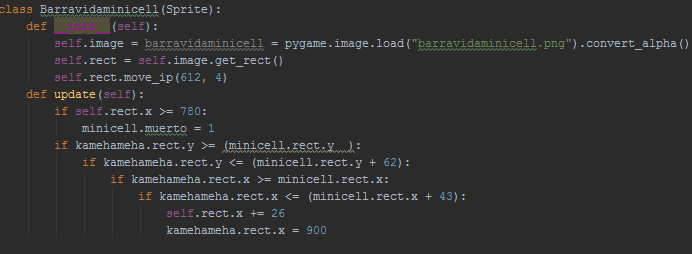
Movimiento



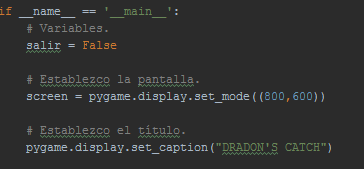
Igual que el objeto “goku” creamos las mismas funciones es decir de disparo, el cálculo de la vida de “minicell” pero con sus imágenes correspondientes

Como se puede observar

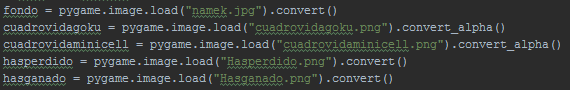




Creamos una condición para la ejecución de la clase “peleas.py” en donde vamos a establecer el tamaño de la pantalla, el título del juego

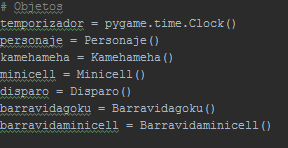


Creamos dos objetos Surface

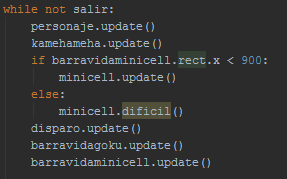


El convert() convierte la superficie a un formato de color que permite imprimirlas mucho más rápido

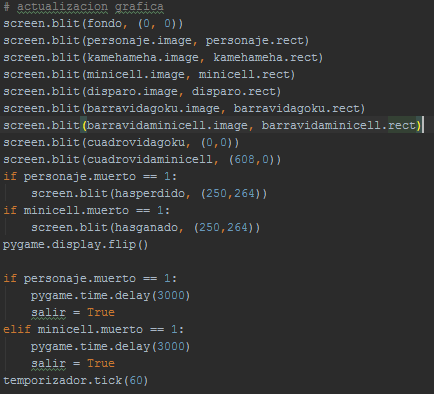
Llamamos a las funciones creadas y añadimos el temporizador



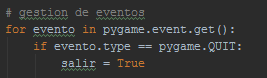
Con un while procedemos a realizar el movimiento del personaje, esto es muy importante para que pueda ejecutarse de manera correcta, y en donda llamamos a las funciones de movimiento de cada objeto



La actualización grafica es decir se actualizar cada vez que realicemos alguna acción y dentro del if para que al momento de cualquier personaje se quede sin vida el time quede ahí congelado y nos muestre quien salió de triunfador



Gestor de eventos

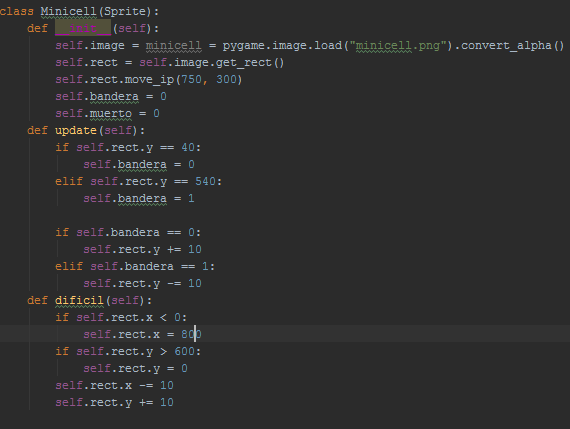


* Como **tercer punto** vamos a crear la clase de “peleas2.py” el cual contiene el código de la ejecución para jugar dos jugadores

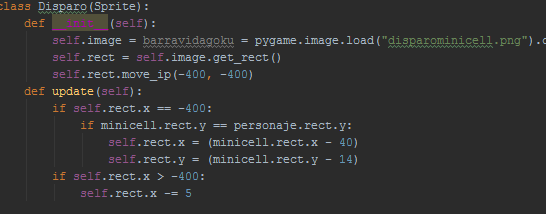
Para esta clase para hacer las mismas clases y funciones de la clase ”peleas.py” lo único que cambia que vamos a realizar una clase para que el segundo jugador pueda jugar también

Para esto se borra la clase del “minicell” de la anterior que era del usuario y creamos una nueva donde vamos a dar parámetros

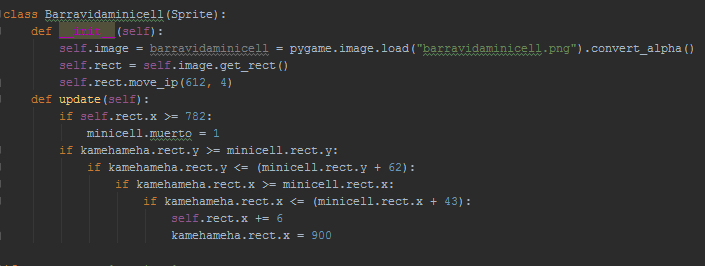
La clase “minicell” creamos al personaje con la imagen correspondiente y el movimiento para que el jugador pueda jugar también en el teclado



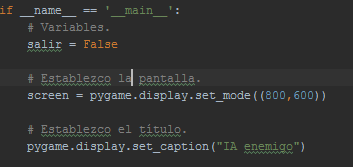
Clase disparo



Clase para la barra de vida de mini Cell

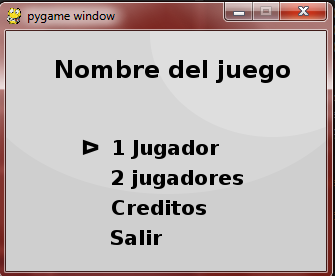


lo único que en esta clase en la función para la ejecución establecemos otro título y establezco el fondo de pantalla como se puede observar a continuación



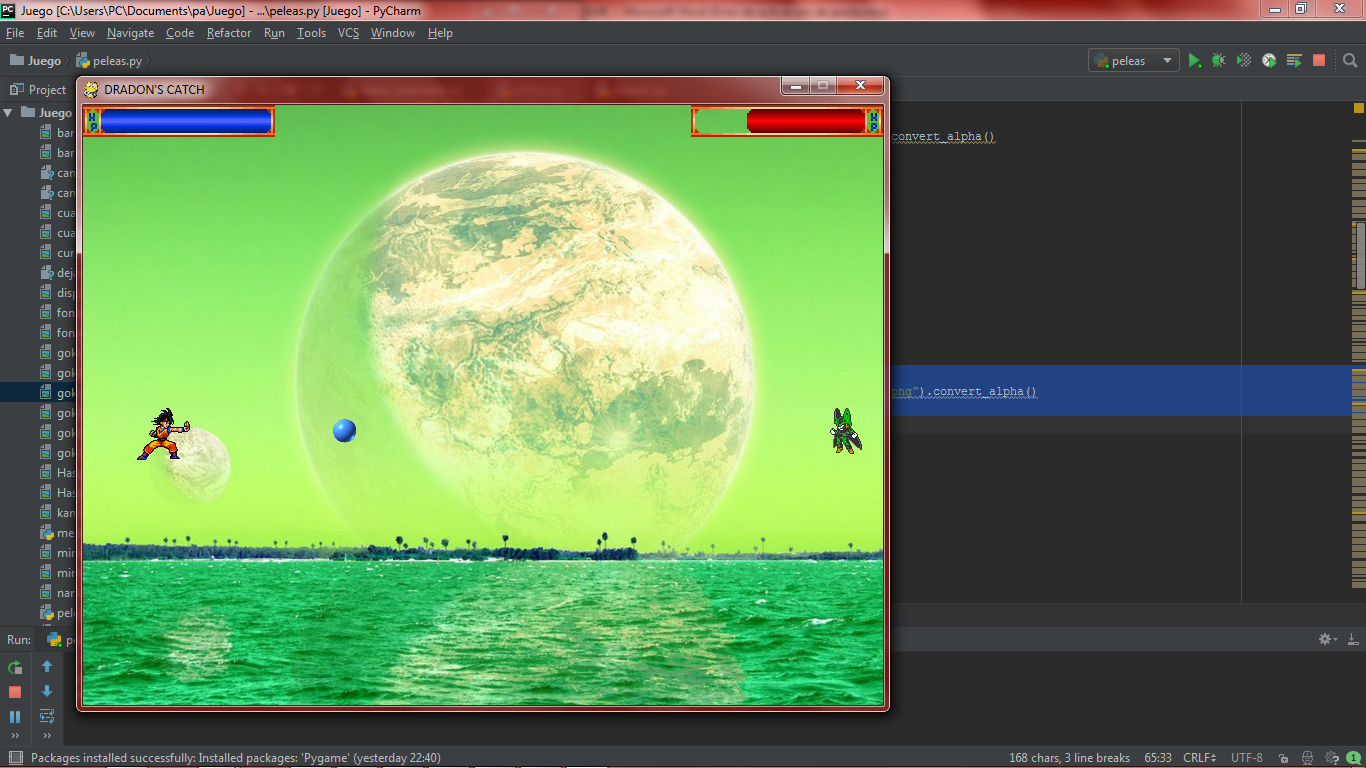
1. RESULTADOS

**MENU**



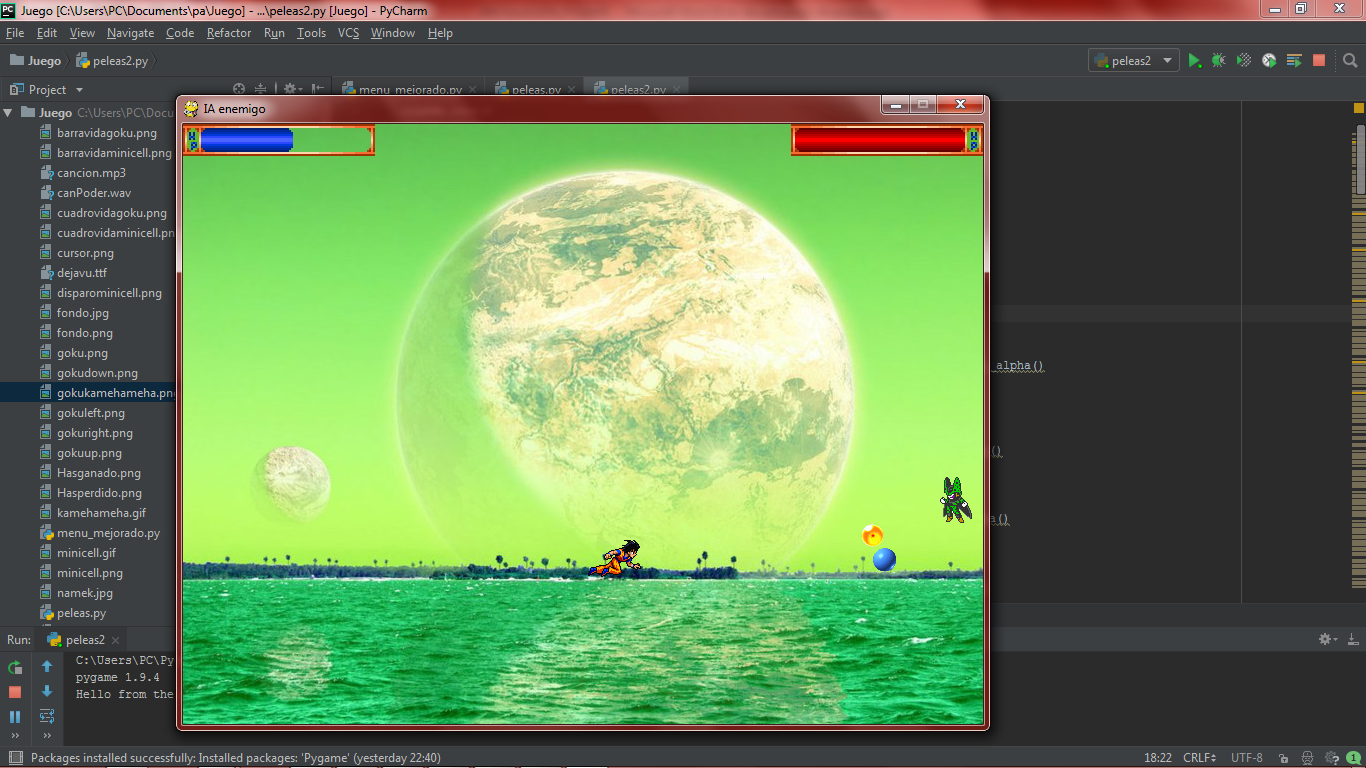
**OPCION 1**

PARA UN JUGADOR

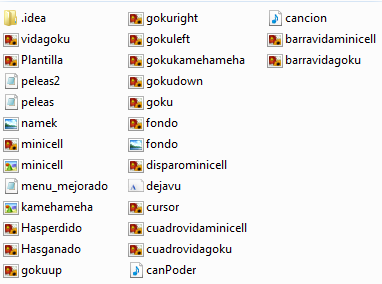


**OPCION 2**

PARA DOS JUGADORES



Archivos que se emplearon en el juego



1. CONCLUSION

Al acabar de realizar el juego en Python nos a conocer muchas cosas tales como crear un juego dependiendo del usuario que desea o que quiere implementar, también por otro punto es muy interesante, te atrae te hace hacer que te guste programar ya que da una satisfacción y que por emplear la aplicación de los conocimientos adquiridos en la materia de Programación Avanzada puedes realizar cualquier problema de la vida de real en un computador

1. RECOMENDACIÓN

Es recomendable utilizar imágenes de tipo que sean compatibles con el programa de Python al igual que la música y otras cosas implementadas en el juego, otra cosa es la función del main no puede faltar porque con eso hace ejecutar al juego y de manera correcta.

1. REFERENCIAS

* 2018. [Online]. Available: https://www.lfd.uci.edu/~gohlke/pythonlibs/#pygame