ARCADE

Universidad De Oriente Campus Poza Rica, Marzo 20, 2024, México

Grover Fernando Campos Medrano, Jorge Alberto González Ruiz, Yeshua Eduardo Mahe Romero

**Abstract-El proyecto “Arcade” es un conjunto de juegos los cuales podremos tener acceso mediante un menú, contando con más de un juego y tener un buen método de entretenimiento. En este documento veremos el desarrollo del mismo con el lenguaje de Python.**

INTRODUCCION

Todos hemos alguna vez probado algún videojuego en nuestra vida, ya sea por interés propio o por la recomendación de un amigo, los videojuegos son un buen medio de entretenimiento a la hora de querer distraernos o pasarla bien, una de las mejores maneras de que los videojuegos se expandieran ante el publico y estos mismo tuvieran la oportunidad de probar uno fue mediante la llegada de los arcade entre los 70 y 80, por eso mismo nosotros mismos fan de los videojuegos quisimos recrear un conjunto de estas experiencias de 8 bits, contando con el mítico ping pong e incluso un juego original de aventura con nuestra mascota universitaria.

1. LIBRERIAS

A la hora del desarrollo de nuestro código utilizamos una variedad de librerías para poder tener funciones especificas para el funcionamiento del código.

**Pygame:** Facilita el desarrollo de videojuegos y aplicaciones, dándonos herramientas como el manejo y uso de sprites, gráficos y animaciones, ventanas y pantallas, sonido, música y las colisiones.

**Tkinter:**  Nos proporciona lo que son las interfaces graficas de usuario o GUI, para poder desarrollar aplicaciones de escritorio, es fácil de aprender, tiene widgets personalizables, como puede ser el tamaño, color de ventanas, botones, etc.

**Turtle:** Este es un módulo de Python que proporciona una forma simple de crear gráficos, es una buena herramienta a la hora de la programación y la geometría del juego. Como puede ser configurar el grosor de una línea, velocidad de las líneas, etc.

**Math:** Es una biblioteca estándar de Python que proporciona funciones y constantes matemáticas para realizar operaciones matemáticas más avanzadas, es flexible, contiene muchas funciones matemáticas.

**Sys:** Es parte de la biblioteca estándar de Python, proporciona acceso a unas variables específicas como un control de salida, variables y funciones del sistema, rutas de importación y contiene una lista de argumentos.

**Time:** Este es un módulo que Python proporciona, contiene diversas funciones relacionadas con la medición del tiempo, tanto como formato de fecha y hora, esperas y pausas, obtención de la hora actual y la medición de las operaciones.

**Subprocess:** Modulo que proporciona Python,

Nota: **Tkinter** como **Pygame** son librerías que tienes que instalar desde la CMD, con los siguientes comandos y ejecutándolas desde la terminal podrás instalarlas.

**pip install pygame**

**pip install tkinter**

1. HERRAMIENTAS

**Visual Basic Studio:** Utilizamos este compilador de código ya que se nos hace más cómodo a la hora de trabajar proyectos, con un fácil uso de su interfaz y las extensiones y demás herramientas que ofrece.

**Python:** Python es uno de nuestros lenguajes de programación favorito, es fácil de utilizar, adaptable y las funciones que contiene fueron clave para realizar este proyecto.

1. DESARROLLO DEL CODIGO

MENU

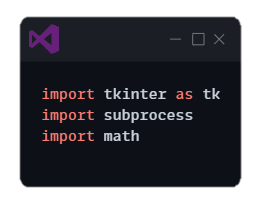
El primer paso es importar las librerías que vamos a utilizar, para el menú se utilizaron tres, las cuales son **Tkinter**, **Math** y **Subprocess**

Ilustración , Librerías

Para el siguiente paso se definen los botones del menú los cuales darán acceso o un aviso si no se encuentra el videojuego, con un total de 6 funciones o 6 botones.

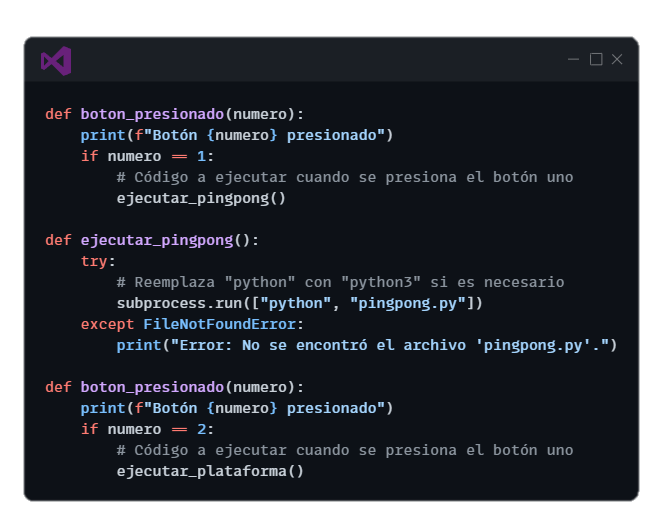
****

Ilustración , Botones

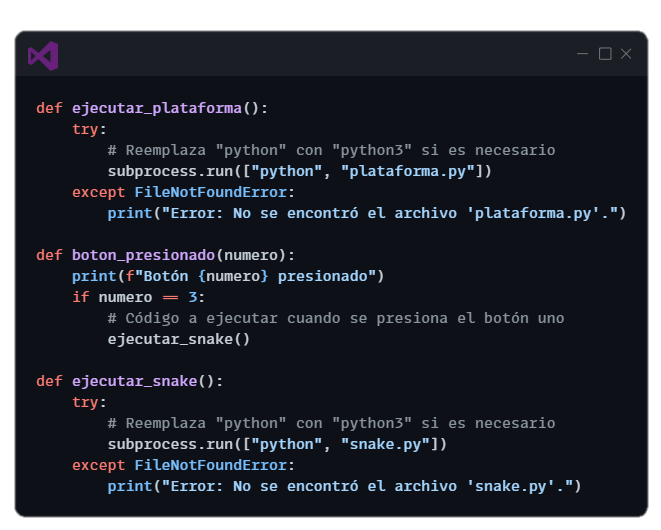


Ilustración .1, Botones

Cada función tiene un propósito los cuales son ejecutar distintos juegos a la hora de seleccionar uno de los tantos botones.

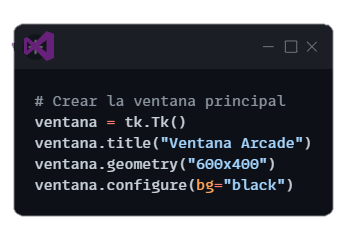


Ilustración , ventana del menú

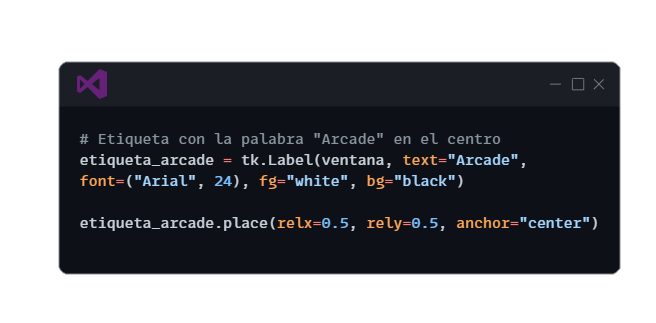
En esta parte del código ese ejecuta la ventana principal, donde se va a visualizar el menú.

Ilustración , librerías ping pong

Ilustración , Etiqueta

Este apartado sirve para que la ventana del menú contenga la etiqueta “menú” en el centro de la ventana.

Ilustración , crear gráficamente botones y loop

Esta sección de código creamos los 6 botones de manera grafica dándole los valores del tipo de forma, color y tamaño de los mismos.

PING PONG

PING PONG

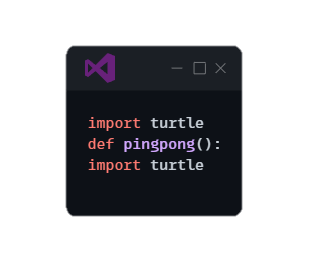
Importamos las librerías**, Turtle**, y definimos pingpong. En este caso solo sería una librería la cual importar.



Ilustración , ventana grafica

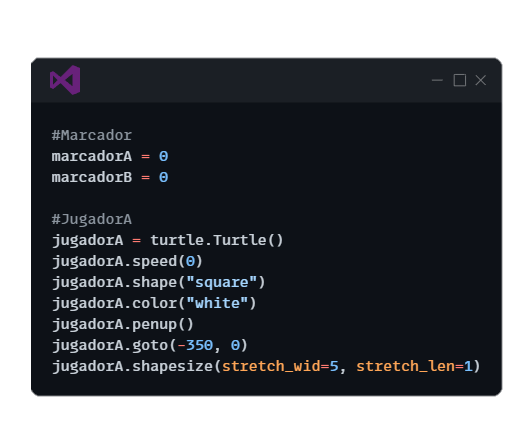
Aquí se ejecuta la ventana del juego, colocamos los valores, el nombre de la ventana, el color, el tamaño y la colisión.

Ilustración , marcador y jugadores

Se crea el marcador el cual dará los puntajes de cada jugador al anotar, así mismo se crean los jugadores, en este caso serán las barras que se mueven de manera vertical, dándole valores como la longitud, el color y que tipo de forma tendrán. Aplicando la misma línea del código para el jugador B.

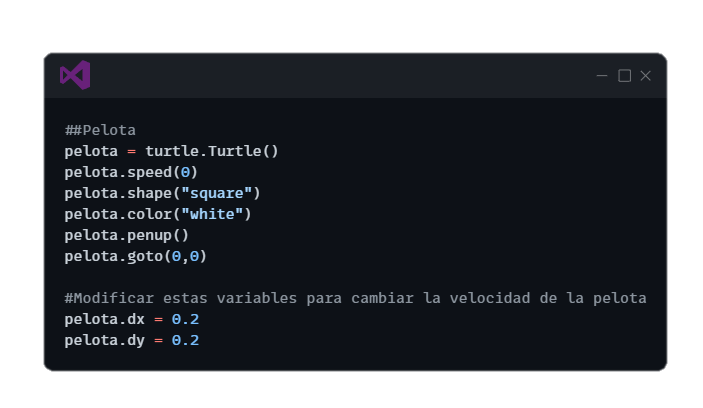
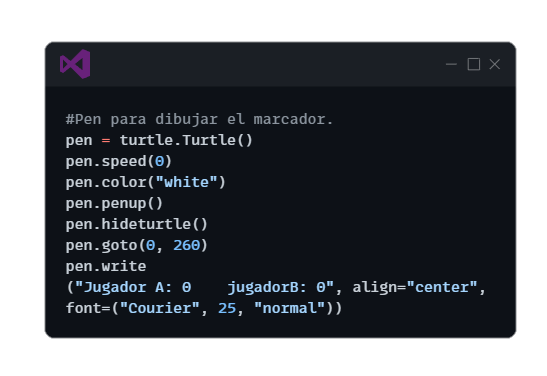


Ilustración , pelota y su velocidad

Se realiza la pelota y la velocidad con la cual esta va a moverse, dándole un aspecto cuadrado, que sea de color blanco y dándole una velocidad de movimiento de 0.2.



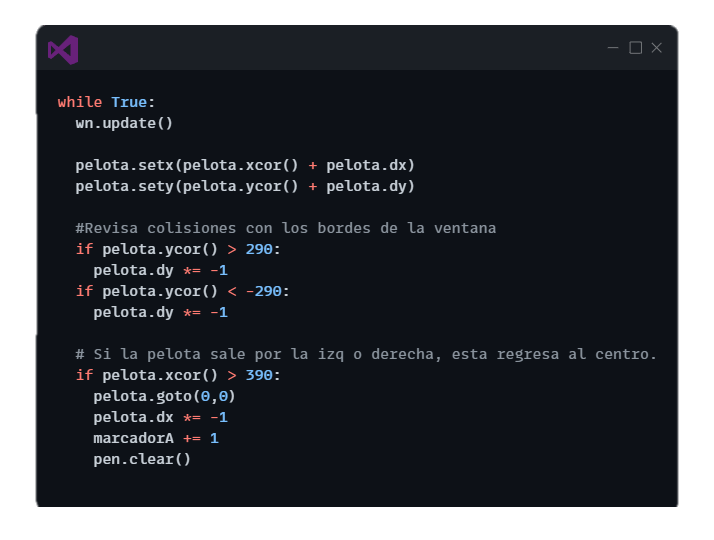


Ilustración , rebote y actualizacion pelota

Ilustración , Marcado grafico

Es aquí donde vamos a crear de manera grafica nuestro marcador, anteriormente solo creamos donde se iba a guardar la puntuación y aquí como se va a mostrar durante el juego.

Asignamos el movimiento de nuestros jugadores, los cuales podrán moverse de arriba hacia abajo, también las teclas las cuales va a reconocer el juego para que este se mueva. Aplica el mismo código para el jugador B.

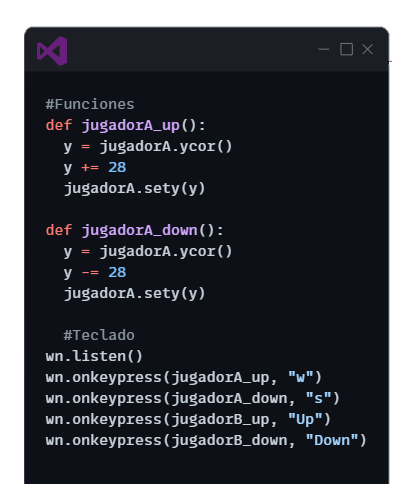


Ilustración , movimiento y teclas

Este bloque de código se encarga de actualizar la posición de la pelota, se hace una limpieza en la pantalla mientras esta se mueve de posición, ya se de manera vertical, horizontal, también detecta si estas alcanzaron los bordes de la ventana, haciendo que están rebote.

Aquí se actualiza el marcador, cada que se anote se borra y actualiza con el puntaje anotado.

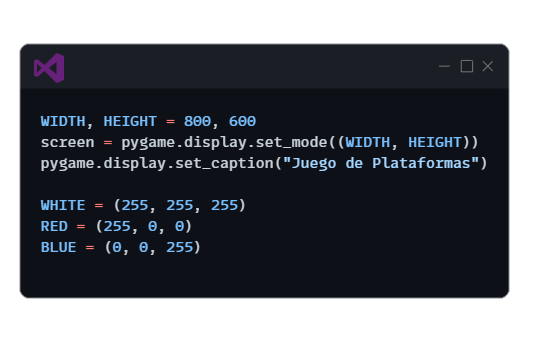


Ilustración , actualizar marcador

Ilustración , ventana

Aquí manejamos las colisiones de los jugadores con la pelota, lo que hace que este rebote en los bloques que serían los jugadores.

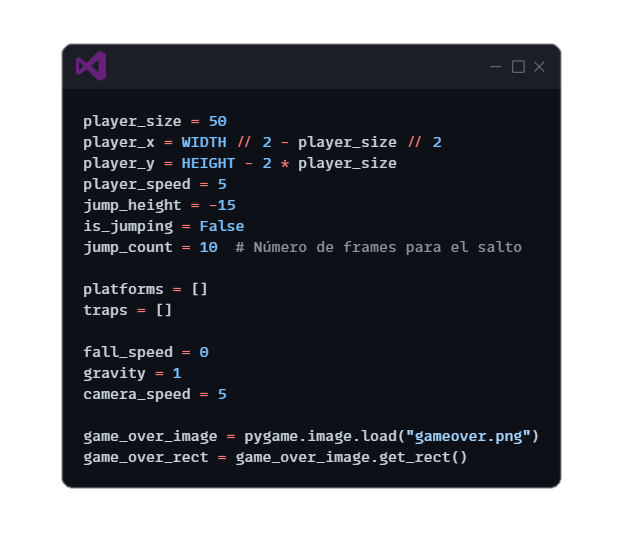


Ilustración , colisiones

PLATAFORMERO

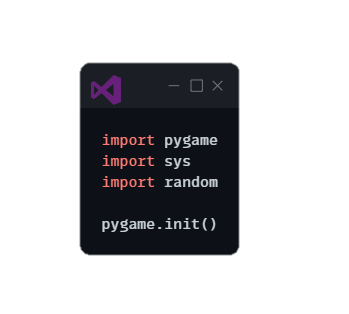
Se importan las librerías necesarias para poder ejecutar el código, estas son **pygame**, **sys** y **random.** Se inicia pygame para poder usar las funciones de la librería y estas funcionen.

Ilustración , personaje, plataforma, game over

Ilustración , librerías

Se crea la ventana y personaliza, el tamaño, ancho y alto, el color, el nombre, el color que va a contener dando valores más blancos.

Creamos al personaje, dándole ciertos valores como la velocidad, la altura que tendrá, como el salto, también creamos las plataformas, la velocidad de caída y la gravedad, colocamos un mensaje o imagen de game over si este llega a perder en el juego.

Se van a crear las plataformas, así como las trampas de manera aleatoria y se va asignar un reloj para el bucle del juego.

Ilustración , reloj de bucle del juego, generar plataformas y trampas

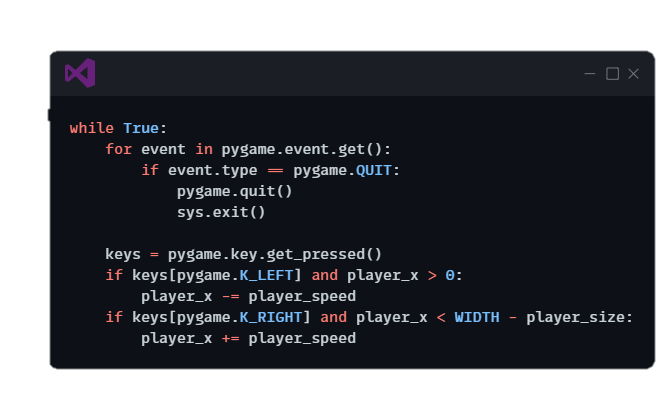
Este apartado maneja lo que es al cerrar el juego, detecta que lo quiere cerrar y cierra el scrip, también tenemos los controles del personaje, moviéndose de izquierda y derecha, también configurando que el jugador no pueda salir de la ventana.

Ilustración ,movimiento, cerrar ventana

Aquí vamos asegurarnos si el jugador esta saltando, toda esta línea de código maneja lo que es el salto del personaje, tanto como la altura del salto, y si este ya termino de saltar.

Ilustración , colisiones de plataformas, gravedad



Ilustración , salto del jugador

Aquí ya se van generando de manera aleatoria las plataformas y se simula el movimiento de ellas de manera que parece que están ascendiendo.

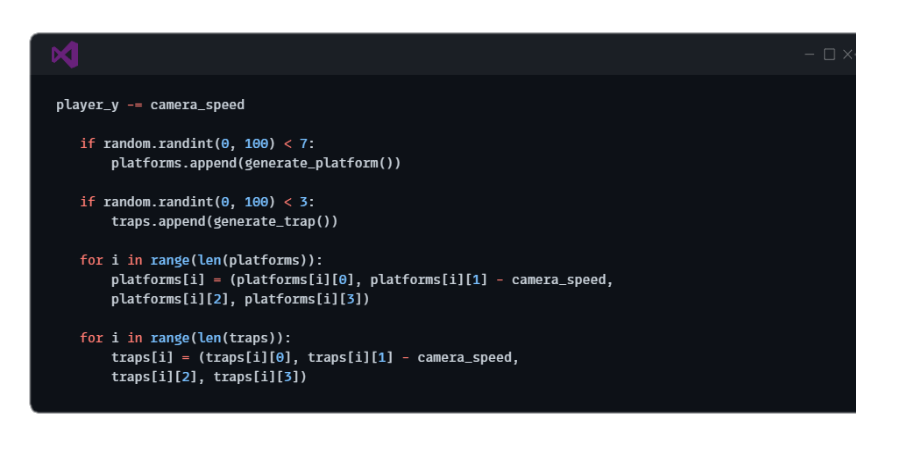
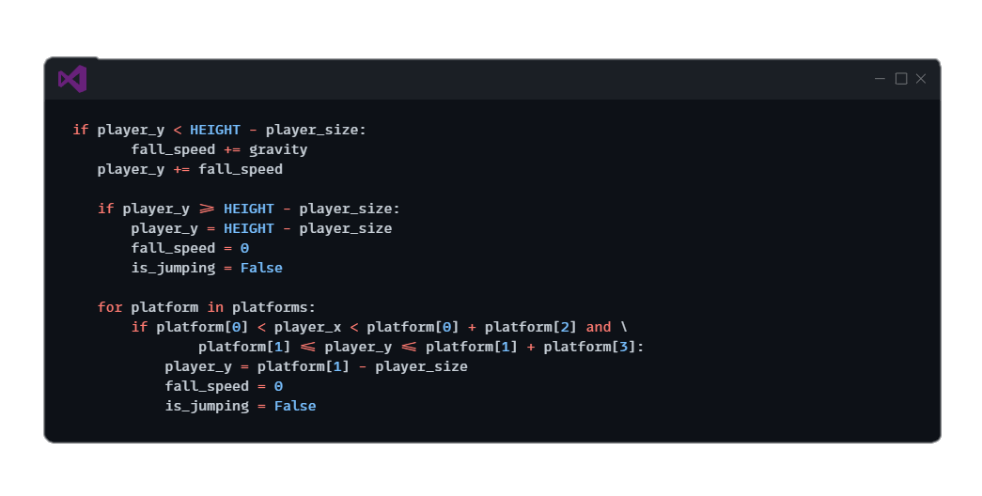


Ilustración , plataformas, movimiento de cámara

Aquí se maneja las colisiones, la gravedad, la actualización de la posición del jugador, todo lo que es el proceso lógico las plataformas y al tocarlas se controla desde aquí, tanto cuando detecta que toco o no.



Se sigue trabajando con las colisiones, tanto para que detecte cuando toca una plataforma y el algoritmo sepa que no esta saltando y lo que pasa cuando llegamos a colisionar con una trampa, la cual te dará un mensaje y perderás el juego, haciendo que salgas del mismo.

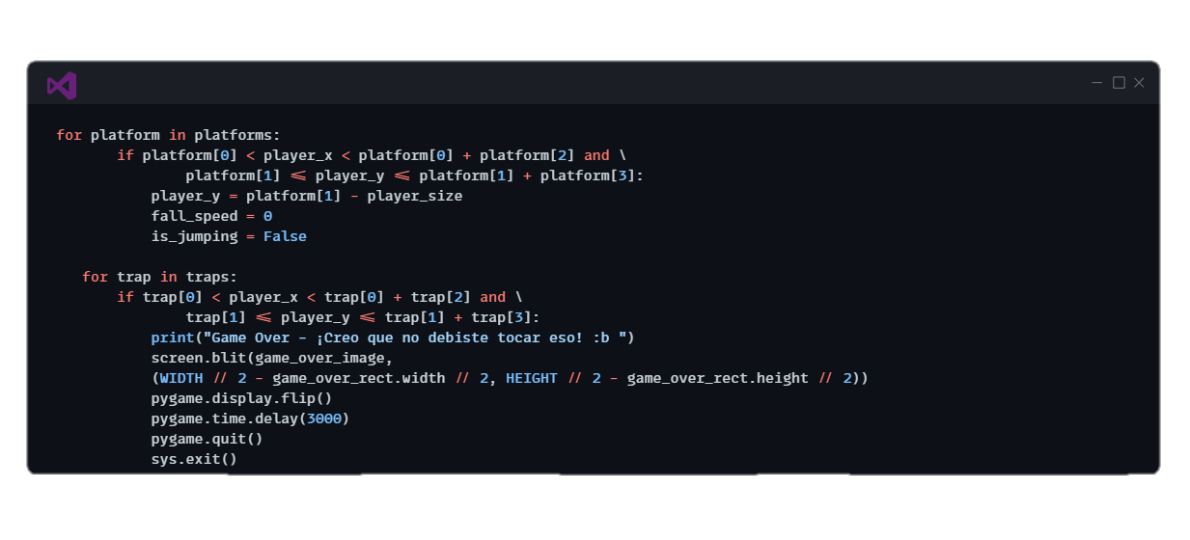


Ilustración , colisiones y colisión trampa

Todo lo que son estas secciones de código trabajan distintos apartados del mismo, aquí ya se dibujan o representan las plataformas o trampas, como rectángulos siendo las plataformas azules y trampas rojas, se hace limpieza de las plataformas y trampas fuera de la pantalla para que no se sature y se van generando nuevas, se actualiza la pantalla para que se noten los cambios generados, y se limitas los cuadros por segundo, también se representa al jugador con un rectángulo rojo.

Ilustración , jugador, limpieza de pantalla, plataformas

IMÁGENES DEL LOS PROGRAMAS EJECUTADOS



Ilustración , menú

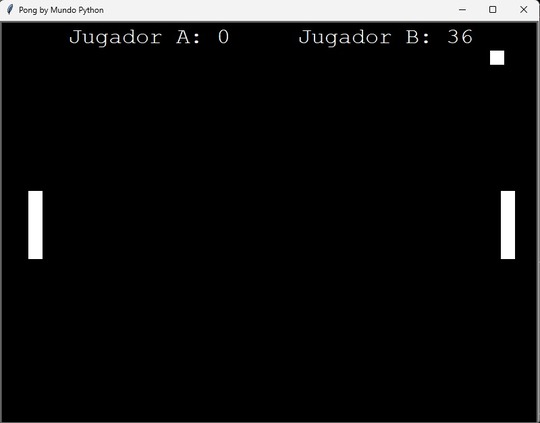


Ilustración , ping pong

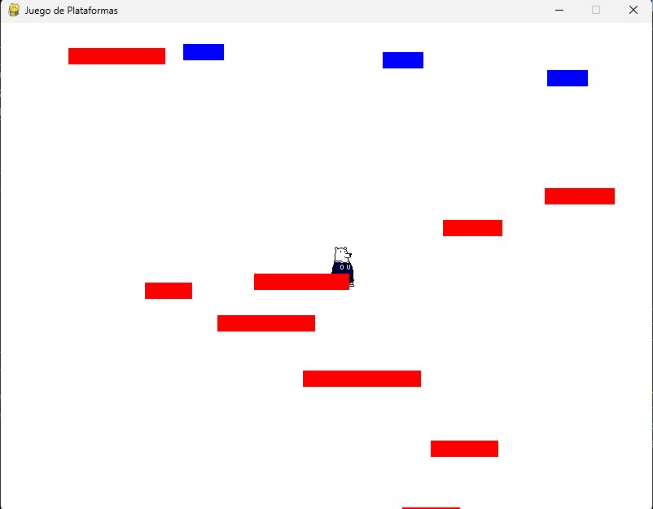


Ilustración , plataformero



Ilustración , game over

REPOSITORIO

Para más información acerca del proyecto les compartimos el enlace del repositorio en GitHub.

<https://github.com/Fercam12/Arcade>

<https://github.com/Fercam12/Arcade/blob/main/README.md>