<!--IE-0117 Programación bajo plataformas abiertas-->

Proyecto Final {

<Por=
"Bryan Cortés"/n
“Danny Solórzano (coordinador)”/n
“Adrián Angulo”>

}

Objetivo General {

Crear un juego de “Buscaminas”

La idea es recrear el conocido juego Buscaminas con las características clásicas que distinguen a este videojuego utilizando el lenguaje de programación de Python y el uso de la biblioteca pygame

El objetivo fue cumplido

Utilizando la biblioteca pygame se pudo recrear la idea BÁSICA del juego

}

Objetivos Específicos {

Generar una interfaz gráfica para el juego

Se buscará crear una interfaz que le permita al usuario revelar las casillas vacías y el número de minas adyacentes en cada casilla

Además se le permitirá elegir el tamaño del tablero y la cantidad de minas en el mismo

El objetivo se modificó

Se decidió olvidar la idea de permitirle al usuario elegir el tamaño del tablero y la cantidad de minas

Ahora, en el juego el usuario solo puede elegir entre 3 niveles de dificultad: fácil, medio, difícil

}

Objetivos Específicos {

Crear un código para generar el tablero

La idea es que en cada partida se genere un tablero con minas en posiciones aleatorias para luego contar la cantidad de minas adyacentes a cada casilla

El objetivo fue cumplido

Se crearon dos clases: Casilla y Botón con las cuales se generó una tercera clase llamada Tablero

A esta clase Tablero se le agregaron diferentes métodos para poder obtener las casillas adyacentes con minas, entre otras funciones

}

Justificación {

El juego clásico de Buscaminas

Se eligió hacer este proyecto porque el Buscaminas es un juego que es muy accesible y fácil de utilizar

No tiene un funcionamiento muy complejo y la idea del juego es simple y concreta, sin embargo, con las pocas opciones que ofrece puede ser muy entretenido

Es una actividad que además de ser divertida permite ejercitar el cerebro

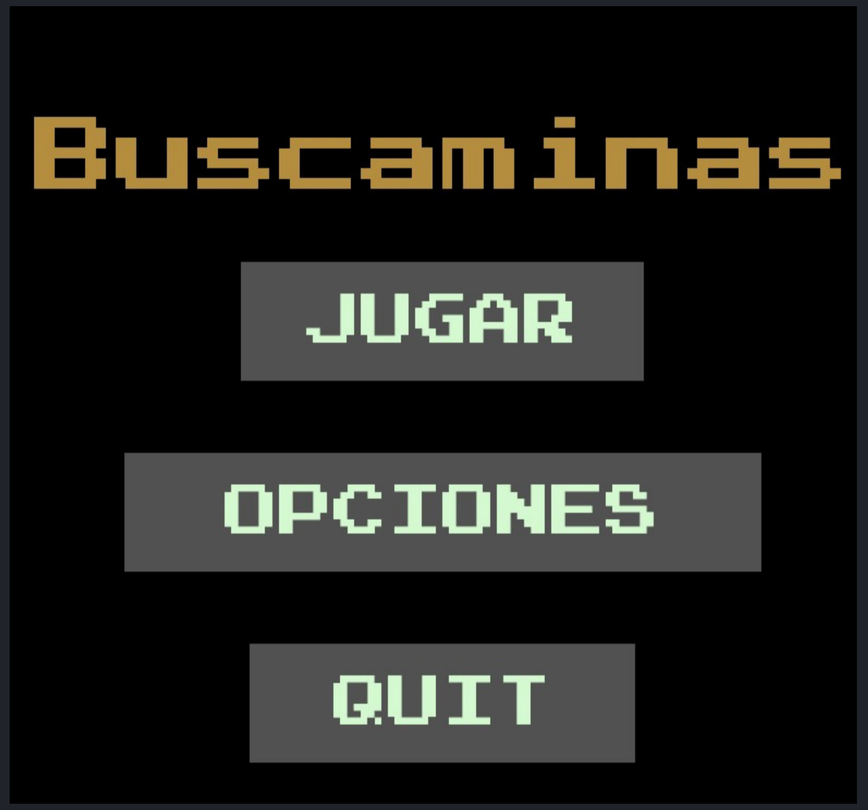
}

Interfaz gráfica {

Inicio del juego

Al iniciar el juego ofrece tres opciones principales:

Jugar
Opciones (Dificultad)
Quit (Salir)



}

Interfaz gráfica {

Opciones de dificultad

Al ingresar a las opciones se ven tres dificultades principales:

Facil (tablero 9x9)
Medio (tablero 12x12)
Dificil (tablero 15x15)

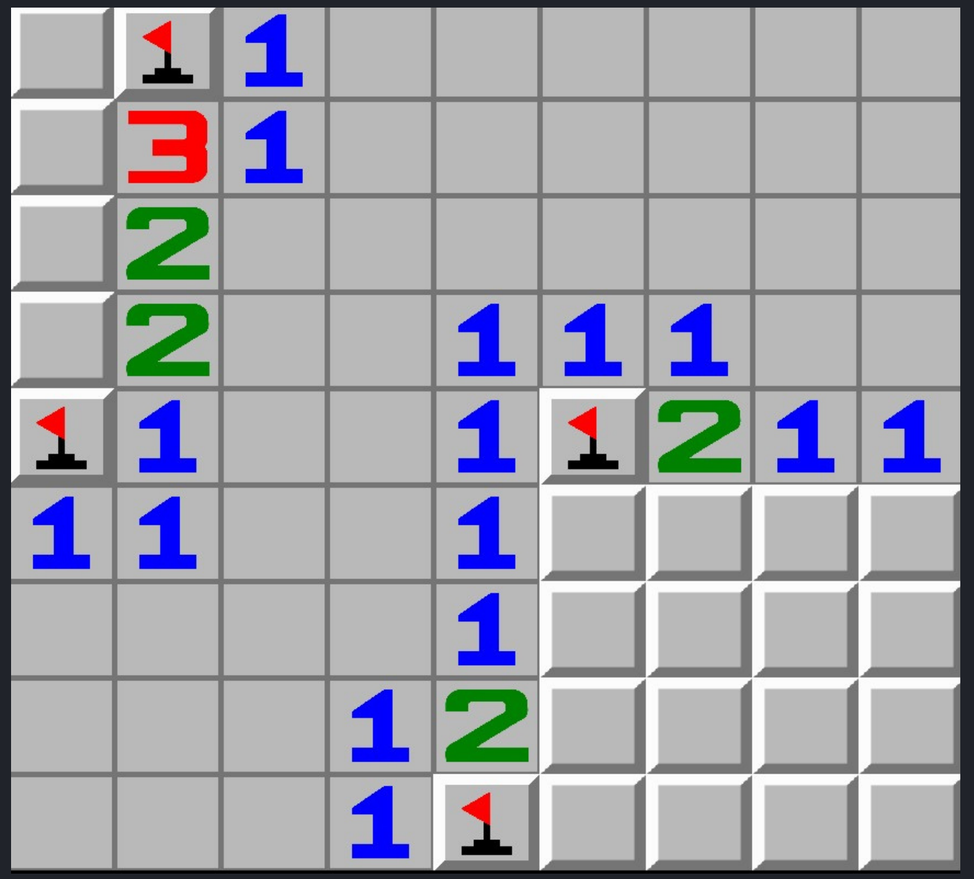
}

Interfaz gráfica {

Interfaz del juego

Al seleccionar un nivel y empezar a jugar la interfaz se vería de la siguiente forma:

Las casillas se revelan con clic derecho y con clic izquierdo se colocan banderas



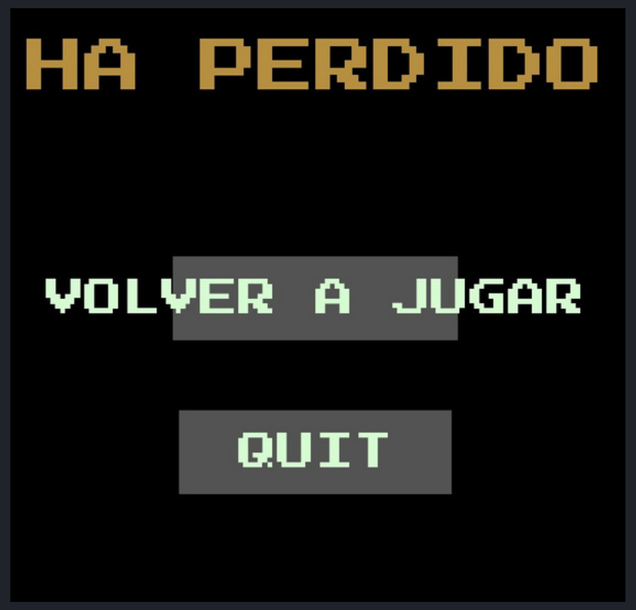
}

Interfaz gráfica {

Perder o ganar

El jugador gana si revela todas las casillas sin bomba y pierde si toca una bomba

Los mensajes de perdedor o ganador se verían de la siguiente forma:



}

pygame {

Facilita el desarrollo de videojuegos y aplicaciones multimedia en Python

Permite gestionar gráficos y entradas del usuario

Permite la creación de ventanas, manipulación de imágenes y manejo de eventos, como clics de ratón y de teclas

Uso en el programa

Crear la interfaz gráfica

Gestionar eventos del usuario

Actualizar la pantalla

Cargar imágenes desde archivos

}

random {

Permite trabajar con números aleatorios

Permite realizar muchas operaciones basadas en el azar

Uso en el programa

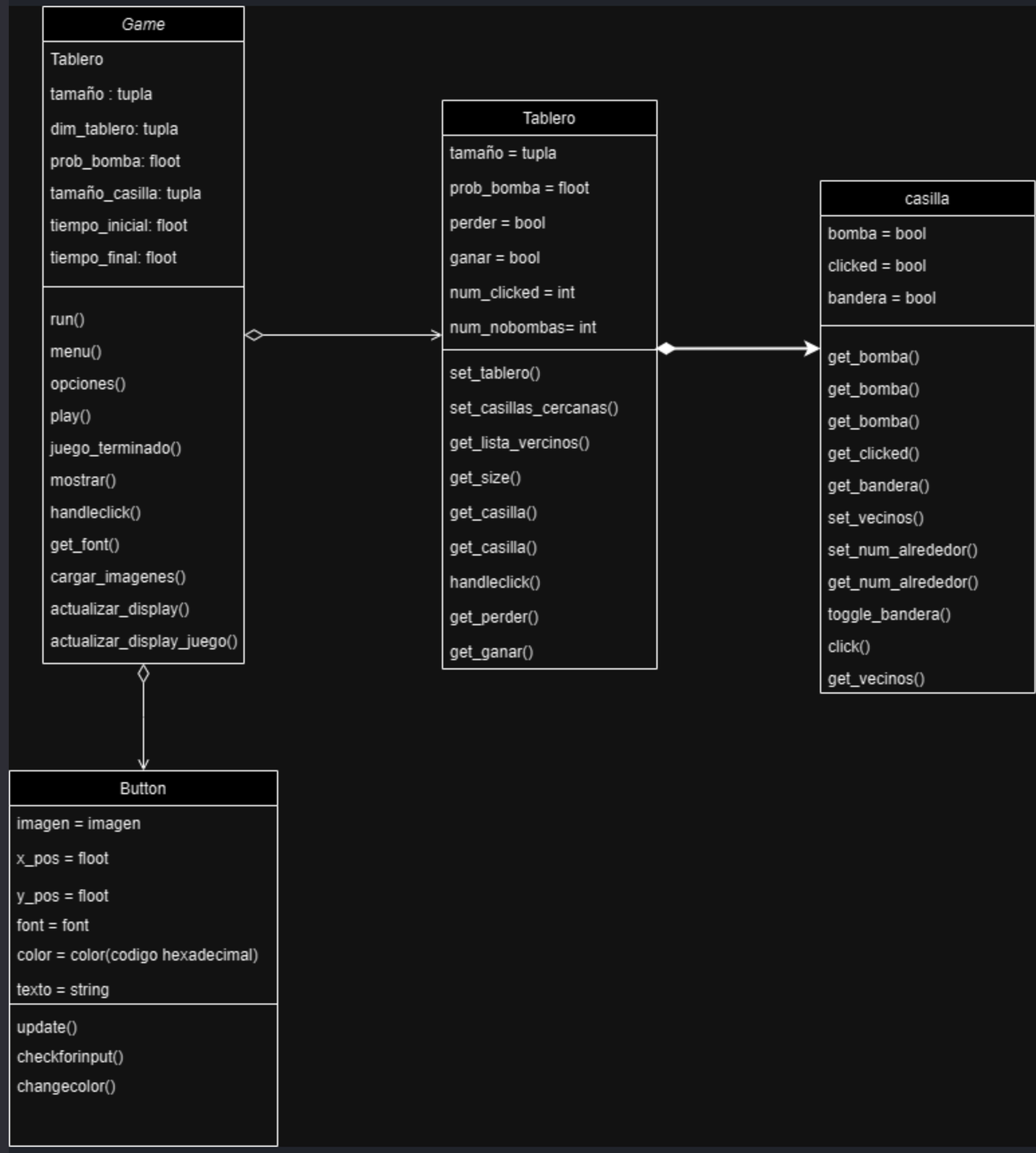
Introducir aleatoriedad en la distribución de bombas

Decidir si una casilla específica contendrá una bomba o no, basándose en una probabilidad

}

Descripción del diseño {

El siguiente diagrama muestra de forma resumida el funcionamiento del código:



Descripción del diseño {

El programa está dividido en cinco archivos, cada uno responsable de una parte específica del juego

casilla.py:

Contiene la clase Casilla. Esta clase representa una casilla en el tablero

Tiene atributos como la presencia de una bomba, si la casilla ha sido clicada, si tiene una bandera, el número de casillas vecinas con bombas, etc

Métodos como como hacer clic, colocar banderas, etc.

Módulos del código

- casilla.py
- boton.py
- tablero.py
- game.py
- main.py

}

Descripción del diseño {

El programa está dividido en cinco archivos, cada uno responsable de una parte específica del juego

boton.py:

Define la clase Button. Esta clase representa un botón en la interfaz

Atributos como tener imágenes, texto y colores que cambian al pasar el mouse sobre ellos

Métodos para actualizar y verificar la interacción del usuario con el botón

Módulos del código

- casilla.py
- boton.py
- tablero.py
- game.py
- main.py

}

Descripción del diseño {

El programa está dividido en cinco archivos, cada uno responsable de una parte específica del juego

tablero.py:

Contiene la clase Tablero. Esta clase se encarga de crear y gestionar el tablero del juego

Genera un tablero con bombas distribuidas aleatoriamente y maneja las interacciones del jugador, como hacer clic en una casilla

Módulos del código

- casilla.py
- boton.py
- tablero.py
- game.py
- main.py

}

Descripción del diseño {

El programa está dividido en cinco archivos, cada uno responsable de una parte específica del juego

game.py:

Contiene la clase Game, que actúa como el controlador principal del juego

Utiliza instancias de las clases anteriores para gestionar la lógica del juego, la interfaz gráfica, y la interacción con el usuario

Métodos para ejecutar el juego, manejar eventos y mostrar mensajes de juego terminado

Módulos del código

- casilla.py
- boton.py
- tablero.py
- game.py
- main.py

}

Descripción del diseño {

El programa está dividido en cinco archivos, cada uno responsable de una parte específica del juego

main.py:

Crea una instancia de la clase Game y usa sus métodos para iniciar el juego y gestionar los eventos del mismo

Módulos del código

- casilla.py
- boton.py
- tablero.py
- game.py
- main.py

}

Conclusiones {

La implementación del código fue difícil

A pesar de que el juego parece ser muy simple, la implementación del mismo no fue fácil ya que existen muchos casos y situaciones a considerar

Se tuvieron que invertir varias horas y días de trabajo para dar con el resultado final

Se cumplieron los objetivos

El objetivo principal se cumplió. Se pudo recrear el videojuego en su versión más clásica

No obstante, algunos objetivos se tuvieron que modificar e incluso reducir para disminuir la complejidad del programa

No se pudieron implementar tantas funciones como se quería en un inicio, pero estamos satisfechos

}

Gracias {

<Por="su atención"/>

}