

PROGRAMACIÓN EN PYTHON

TRABAJO FINAL



Nombre del trabajo:	Plataforma de minijuegos
Identificador de equipo:	1.B
Componentes: Número de matrícula - nombre	Carola Adrados Herrero 18001 Juan Berenguer Triana 19036 Sergio Gómez Montes 18150
Fecha entrega:	21/01/2023

Índice

1.	OBJETIVO DE LA APLICACIÓN	2
2.	REQUISITOS FUNCIONALES	2
3.	ESTRUCTURA DE LA APLICACIÓN	2
4.	MANUAL DE USUARIO	2
5.	PRESUPUESTO	2

1. OBJETIVO DE LA APLICACIÓN

El principal objetivo de esta aplicación se trata del desarrollo de una plataforma de minijuegos que permita la selección de distintos modos de juego. Puedes elegir entre jugar de forma individual y seleccionando la dificultad de estos o una partida más larga que salte de minijuego en minijuego de forma aleatoria. Se guardará también un historial de las partidas.

Los juegos implementados son: Runner, Snake, Pong y Catch.

2. REQUISITOS FUNCIONALES

Al ejecutar el programa debe:

- Generar una ventana con un menú que permita elegir el modo de juego y consultar las puntuaciones previas.
- Al elegir el modo de juego debe de aparecer una ventana nueva con el modo.
- En el menú de *Juego Individual* deben presentarse los posibles minijuegos (Snake, Pong, Runner o Catch), y sus correspondientes dificultades (fácil, normal, difícil o imposible), con botones para su selección.
- Al pulsar el botón jugar aparecerá la ventana del minijuego correspondiente y comenzará la partida.
- Una vez finalice la partida se guardará la puntuación en el historial.
- Respecto a la opción *Super Party*: se sucederán partidas de los minijuegos de forma que o bien sus dificultades (si se elige la opción *Juego*) o los juegos (si se elige la opción *Dificultad*) vayan cambiando según se superan niveles. En la opción *Juego* las dificultades avanzan desde fácil a imposible escalonadamente, mientras que en la opción *Dificultad* el orden de los juegos es aleatorio.
- Para consultar el historial al hacer click sobre el botón en el menú principal se abrirá una ventana donde se podrán consultar:
 - El número de partidas jugadas y ganadas en cada juego y dificultad
 - La mejor puntuación obtenida en cada juego y dificultad.
 - Resetear todo el historial.
- En cuanto a los juegos se espera la correcta implementación de Runner, Snake, Pong y Catch. Es necesario que en cada minijuego funcione:
 - En el Runner: Movimiento correcto del personaje y de los obstáculos, colisión correcta entre el personaje y los obstáculos, si se produce, que la barra de vida baje en consecuencia. Además las animaciones y texturas deberán funcionar correctamente.
 - En el Snake: Movimiento de la cabeza de la serpiente y del cuerpo, el cual deberá seguir a la cabeza y aumentar de longitud según se vayan consiguiendo puntos,

aparición de naranjas y veneno en lugares aleatorios del mapa, detectar colisiones entre la serpiente y los objetos varios del mapa, además deberán oírse diversos sonidos implementados en el juego.

- o En el **Catch**: Movimiento del personaje de lado a lado de la pantalla (pasando al extremo opuesto cuando se sale de la pantalla), movimiento vertical de los objetos y colisiones, donde se actualizará la puntuación en función de los objetos que hayan colisionado con el personaje
- o En el **Pong**: El movimiento de la pala del jugador por teclado; el correcto movimiento del pong; detectar colisiones entre bola y pala y bola y pared; el conteo de la puntuación; el control de la pala de la computadora; y la regulación del nivel de dificultad.

3. ESTRUCTURA DE LA APLICACIÓN

La aplicación está estructurada en un código principal, las librerías y los módulos de los juegos. En el código principal se encuentran: la clase **VentanaPrincipal** con las funciones individual, aleatorio y records; la clase **VentanaReset** y su función resets; la clase **VentanaRecords** con *regJuego*, *regDif*, *record* y *reset* ; la clase **VentanaIndividual** con *regJuego*, *regDif* y *jugar*; **VentanaParty** y sus funciones *juego* y *dificultad*; **VentanaJuego** con *regJuego* y *jugar*; y **VentanaDificultad** con *regDif* y *jugar* . Además de las funciones: *update_highscores*, *read_highscores* y *reset_highscores* Dentro de cada módulo de juego se encuentra definida la función del juego que es importada al código principal, en esta quedan definidas las clases necesarias para cada juego:

- En el **Runner** las clases son: **Runner**, **BasicObstacle**, **MovingObstacle** y **Trofeo**, sin haber relación jerárquica entre ellas.
- En el **Snake** las clases son: **Serpiente**, **Comida** y **Veneno**. Se define la función bucle que contiene el bucle del juego.
- En el **Catch** las clases son: **Leprechaun** y **BasicCoin**, sin haber relación jerárquica entre ellas.
- En el **Pong** las clases son: **Pala** y **Bola** y son subclases de la clase de la librería pygame: **Sprite**. Además, se definen la función redraw para el refresco de la ventana.

Todos los módulos tienen aparte una función para pasar al menú el resultado del juego: *men_juego* siendo juego el nombre del que sea.

Las librerías utilizadas son **pygame**, **random**, **time** y **sys**.

4. MANUAL DE USUARIO

MENÚ: Nos dejará elegir entre los dos modos de juego y el historial.

MODO INDIVIDUAL: Deberemos elegir entre uno de los cuatro juegos disponibles y una dificultad, posteriormente pulsaremos sobre el botón Jugar para dar comienzo a la combinación escogida.

Una vez acabado el juego, se nos notificará la puntuación obtenida, guardará también un registro de la puntuación y de la partida.

MODO PARTY: Primero hay que elegir entre *Juego* o *Dificultad*. En la primera opción, se elige un juego (en otra ventana) y se juegan las cuatro dificultades disponibles, hasta que no se supere un nivel. En la segunda opción, se elige una dificultad y se juegan los cuatro juegos en esa dificultad, en orden aleatorio y hasta que no se supere un nivel.

HISTORIAL: Deberemos elegir la dificultad y el juego, al pulsar sobre el botón *Ver Historial* se nos mostrará un registro de las partidas así como nuestra puntuación máxima.

Un botón nos permitirá resetear el historial si lo pulsamos, teniendo que confirmar nuestra elección en otra ventana.

RUNNER: Se iniciará el juego con una pantalla en la que nos indica el objetivo del mismo, esquivar los obstáculos.

- Cada vez que toquemos un obstáculo se pierde vida, y al quedar aplastado entre el límite izquierdo de la pantalla y un obstáculo se pierde toda la vida
- Si se pierde toda la vida, se finaliza el nivel, sin superarlo
- Usaremos las teclas de dirección para movernos y llegar al trofeo, momento en el que se supera el nivel

SNAKE: Comenzará el juego con la cabeza de la serpiente en el centro de la ventana y una fruta y, en el modo imposible, una fruta envenenada en posiciones aleatorias del mapa.

- El objetivo del juego es que el primer bloque de la serpiente colisione con los bloques naranjas, si lo consigues el cuerpo de la serpiente crecerá en un bloque. Si chocas contra tu cuerpo, una pared o un bloque morado perderás.
- Para mover la cabeza de la serpiente podrás usar o bien las teclas de dirección o las teclas A, W, S, D.

CATCH: Empezaremos en el centro inferior de la ventana, de la parte superior caerán monedas de cobre, plata y oro y carbón a diferentes velocidades.

- Usaremos las flechas de dirección para movernos de izquierda a derecha. El objetivo es desplazarse para recoger las monedas al colisionar con ellas.
- Recoger las monedas aumentará la puntuación y tocar el carbón tendrá el efecto contrario.

- La partida tiene una puntuación objetivo sin embargo no termina hasta que caigan todos los objetos de la partida, pudiendo así puntuar por encima del objetivo para establecer un récord.

PONG: Al abrirse la ventana comenzará el juego, la pala del jugador se trata de la pala de la izquierda.

- Para mover arriba y abajo la pala utilice las teclas W y S respectivamente. Una vez el pong colisione con una pala cambiará su dirección.
- Para puntuar es necesario conseguir que el contrario no alcance el pong a tiempo.
- Una vez se alcance la puntuación máxima: si gana el jugador se continuará la partida hasta que el jugador decida salir o la CPU alcance la puntuación objetivo. Si gana la CPU se saldrá automáticamente. En caso de salir en mitad de la partida(haciendo click sobre la cruceta de la ventana) se habrá ganado si se ha alcanzado la puntuación objetivo.

5. PRESUPUESTO

En conjunto las horas de trabajo han sido alrededor de 60 horas.