



Juego de la Serpiente

Cristian Pulido Póvoa, 02290419X, 2º Grado Superior de
Desarrollo de Aplicaciones Multiplataforma 2019 - 2020

Resumen

El presente proyecto trata acerca de la creación y desarrollo de un videojuego en el cual encarnaremos a una serpiente que debe comerse todas las manzanas, y que además pueda servir para aprender y repasar conceptos de programación en Python de forma divertida y entretenida, de modo que sea una gran herramienta educativa que complemente el aprendizaje de la asignatura en las clases y las prácticas. Para llevar a cabo este videojuego, ha sido fundamental apoyarse sobre el temario de la asignatura.

Gracias a esto, se espera que este videojuego ayude a futuras generaciones de nuevos programadores.

Abstract

The purpose of the current project is the creation and development of a full snake type videogame that allows learning and reviewing Python programming concepts in a fun way. This videogame is focused in a way to be a great educational tool that complements the learning of the subject at classes and practices. For developing this videogame, it has been vital basing on the Programming course topics.

Thank this, it is hoped that this videogame helps future generations of new programmers.

Índice

Resumen.....	2
Abstract	2
Índice	4
Justificación	5
Introducción	6
Introduction.....	6
Objetivos	7
Análisis de Requerimientos	8
Requisitos Ordenador:.....	8
Herramientas para el desarrollo:	8
Diagrama de Gantt.....	9
Desarrollo	10
Diseño de interfaz:	22
Manual de Usuario – User Manual	23
Casos de Uso.....	27
Frontend:	27
Backend:	27
Conclusiones	28
Bibliografía.....	29

Justificación

La razón para la que se va a realizar este proyecto no es otra si no la de desarrollar una aplicación propia que sea funcional y así poder aprender durante todo el proceso y mejorar los conocimientos que ya se posean. Otra de las razones por la que se realiza este proyecto es la de ayudar y/o entretener a las personas, y poder formar parte de ello.

Las motivaciones principales para este proyecto son: un pequeño tributo a este clásico juego que a muchos nos hizo sentir atraídos por este tipo de entretenimiento, que, aunque en la actualidad ya se está dando este fenómeno de resurgimiento del género, con nuevos títulos y reediciones de los clásicos, sigue estando en clara desventaja tanto en ventas como en preferencias de los propios jugadores; potenciar este tipo de juegos y redirigir el género a proyectos educativos que resulten divertidos para el usuario y además, proporcionar una herramienta de repaso que apetezca jugar una y otra vez.

Para llevar a cabo este proyecto educativo, se ha decidido partir desde cero, porque lo existente no se adecuaba a lo que se quería.

Introducción

Con el paso del tiempo, los videojuegos se han ido introduciendo en el mundo de la educación de forma cada vez más clara. Aunque si bien es cierto que el terreno educacional se resiste a utilizar los videojuegos como método de enseñanza, cada vez son más colegios, institutos y universidades los que utilizan los videojuegos como pequeños experimentos y pruebas para ver cómo responden los alumnos a estos estímulos.

Puede que dentro de varios años se lleguen a utilizar los videojuegos como modo de enseñanza universal para poder recrear las condiciones necesarias que exige cada asignatura.

¿Es viable? La respuesta depende de varios factores, ya que en primer lugar se debe superar la barrera tecnológica, en la que todo el mundo tuviera un ordenador personal u otro tipo de dispositivo de acceso, como consolas y móviles. Y en segundo lugar la predisposición de padres y profesores a permitir y usar estas herramientas para el aprendizaje.

Introduction

With the passage of time, video games have entered the world of education more and more clearly. Although it is true that the educational field refuses to use video games as a teaching method, more and more schools, institutes and universities use video games as small experiments and tests to see how students respond to these stimuli.

It is possible that in several years video games will be used as a universal way of teaching to recreate the necessary conditions that each subject demands.

Is it viable? The answer depends on several factors, since in the first place the technological barrier must be overcome, in which everyone had a personal computer or another type of access device, such as consoles and mobiles. And secondly, the predisposition of parents and teachers to allow and use these tools for learning.

Objetivos

El objetivo principal de este proyecto es el diseño y la implementación de un videojuego mediante el lenguaje de programación Python. Un lenguaje cada vez más utilizado debido a su sencillez y su ayuda al programador a la hora de crear aplicaciones o programas.

El juego deberá ser capaz de funcionar sin ningún tipo de error o fallo en su código. Deberá implementar un menú principal con dos opciones: jugar y salir. También contará con el propio juego donde se moverá a una serpiente en cuatro direcciones (arriba, abajo, derecha e izquierda) que deberá comerse una manzana que se generará de forma aleatoria en la pantalla de juego. Además, contará con un menú de pausa donde se podrá seguir jugando a salirse del juego. Y por último contará con una pantalla de game over, que saldrá en el caso de que el jugador choque con una de las paredes o se coma sí mismo. Esta última pantalla dará la opción de volver a jugar o de salirse del juego.

Para hacer el juego más complejo, y a su vez más entretenido, también poseerá fondos de pantalla para sus distintas etapas y, además, todo estará acompañado por una canción como sonido de fondo.

Análisis de Requerimientos

A continuación, se analizarán los requerimientos:

Requisitos Ordenador:

Se desarrollará en un ordenador con capacidad de 4gb de RAM con un procesador Intel Celeron de 2.4 GHz y una tarjeta gráfica integrada.

Herramientas para el desarrollo:

Para desarrollar el juego se utilizará Python como lenguaje, y como programa se utilizará Thonny acompañado de visual studio code. Y para la base de datos se implementará SQLite3.

Se estima que el desarrollo del juego lleve aproximadamente 245 horas. En las cuales se dividirá en distintas fases, siendo la primera de ellas la planificación y los bocetos de la aplicación a desarrollar. La segunda fase será el desarrollo de la aplicación.

Y, por último, la fase final, que consistirá en la solución de pequeños errores y en pulir la propia aplicación dándole un aspecto más sobrio.

Durante la planificación y desarrollo del proyecto no se utilizarán ningún tipo de web y/o arquitectura de redes para su desarrollo.

Diagrama de Gantt

ACTIVIDAD	DURACIÓN(semanas)	FECHAS
Primero(Planificación)	1	1 Abril – 8 Abril
Segundo(Desarrollo)	6	8 Abril – 20 Mayo
Tercero(Finalización)	1	20 Mayo – 27 Mayo
Cuarto(Memoria)	1	27 Mayo – 2 Junio

ACTIVIDAD	1	2	3	4	5	6	7	8	9
Primero									
Segundo									
Tercero									
Cuarto									

Desarrollo

A continuación, se detallará el desarrollo del videojuego desarrollado para este trabajo final de grado.

```
import pygame
import time
import random
import sqlite3
```

Al empezar el proyecto se debe importar Pygame, pilar fundamental a la hora de crear cualquier videojuego 2D de forma sencilla. Mediante Pygame podemos utilizar sprites (objetos), cargar y mostrar imágenes en diferentes formatos, sonidos, etc. Además, al ser un módulo destinado a la programación de videojuegos se puede monitorizar el teclado o joystick de una manera bastante sencilla.

El módulo time de la biblioteca estándar de Python proporciona un conjunto de funciones para trabajar con fechas y/o horas. Además de estas funciones hay otras relacionadas en los módulos datetime y calendar que conviene conocer. En este proyecto en concreto ha sido utilizado para el tiempo que deben mostrarse las distintas ventanas del juego.

La biblioteca random contiene una serie de funciones relacionadas con los valores aleatorios. Para este proyecto es usado específicamente en la aparición de la manzana en cualquier punto de la ventana.

Y, por último, se ha importado sqlite3 para la creación y uso de una base de datos en el proyecto. La cual deberá almacenar y mostrar los datos de cada jugador, siendo su id, nombre y puntuación.

```
#Creamos la base y el cursor
miConexion = sqlite3.connect("BaseJugadores")
miCursor = miConexion.cursor()
miCursor.execute("""
    CREATE TABLE Puntuación(
        ID INTEGER PRIMARY KEY AUTOINCREMENT,
        NOM_JUG VARCHAR(50),
        PUNTOS INTEGER)
    """)
miConexion.commit()
miConexion.close()
```

El siguiente paso que se ha realizado es la creación de la base de datos que se va a utilizar en el proyecto.

Para ello se ha de crear primero la conexión con la base asignándole un nombre a la base. Después se ha de crear el puntero que nos permitirá movernos por la base y poder manejarla.

Ya creada la conexión y el cursor, pasamos a crear la sentencia SQL que creara la tabla en la cual pasaremos a almacenar los datos de los jugadores:

- El nombre de la tabla será Puntuación ya que girará en torno a la puntuación de los jugadores. Poniéndole este nombre se facilita recordarlo cuando necesitemos hacer referencia a la tabla y guardar datos en ella.
- Creamos un ID de tipo INTEGER (valores numéricos) que será clave primaria autoincrementable. Esto quiere decir que sus valores no podrán repetirse y que cada dato nuevo que sea introducido se le asignara un valor (id) automáticamente.
- La siguiente columna de la tabla se reserva para el nombre del jugador mediante un VARCHAR con tamaño para almacenar 50 bytes.
- Por último, pero no menos importante, añadimos un INTEGER denominado puntos, el cual almacenará los puntos que alcance el jugador en la partida.

El commit nos permite actualizar la tabla, en caso de haber introducido nuevos valores y/o parámetros en la tabla.

Una vez creada la base y su correspondiente tabla, cerramos la conexión.

```
#Iniciamos pygame
pygame.init()
pygame.mixer.music.load("C:/Users/Usuario/Desktop/Proyecto_Final_Serpiente/serpiente.mp3")
pygame.mixer.music.play()
```

Una vez hayamos importado todo lo necesario y hayamos creado nuestra propia base de datos, pasamos a iniciar el Pygame, el cual nos permitirá crear nuestro videojuego.

También se le ha añadido la música que ira acompañando al jugador durante su partida. Para ello se ha cargado la canción con la ruta donde se encuentra para después reproducirla en la siguiente línea de código.

```
#Declaración de constantes
WHITE = (255, 255, 255)
BLACK = (0, 0, 0)
RED = (255, 0, 0)
BLUE = (99, 166, 247)
BLUE2 = (73, 242, 189)
GOLD = (235, 158, 52)
VIOLET = (161, 79, 255)
AZULTUR = (23, 207, 133)
```

Las siguientes líneas del código están reservadas para declarar las constantes que utilizaremos más adelante para crear nuestro videojuego.

Las constantes son valores que nunca van a cambiar a lo largo del videojuego, así como de su desarrollo. En este caso serán utilizadas para asignar los colores a los textos que irán apareciendo en los distintos menús del juego.

```
#Creación de la ventana
display_ancho = 700
display_altura = 500
superficie = pygame.display.set_mode((display_ancho,display_altura))
pygame.display.set_caption("Jörmundgander")
```

Pasamos a la creación de la ventana en la cual se reproducirá nuestro juego. Para ello se le ha de asignar una altura y un ancho según el cual se acoplarán todas las imágenes, así como la propia manzana y la serpiente. Se le ha asignado el nombre de superficie a la ventana. Así como el nombre que llevara en la barra superior.

```
#Creacion de variables
reloj = pygame.time.Clock()
fondo = pygame.image.load('C:/Users/Usuario/Desktop/Proyecto_Final_Serpiente/fondoinicio.jpg').convert_alpha()
fondojuego = pygame.image.load('C:/Users/Usuario/Desktop/Proyecto_Final_Serpiente/cesped.jpg').convert_alpha()
fondogameover = pygame.image.load('C:/Users/Usuario/Desktop/Proyecto_Final_Serpiente/gameover.jpg').convert_alpha()
f_pausa = pygame.image.load('C:/Users/Usuario/Desktop/Proyecto_Final_Serpiente/fondopausa.jpg').convert_alpha()

#Ajustar tamaño de la serpiente y su velocidad
bloque_tamano = 10
FPS = 20

font = pygame.font.SysFont(None, 25)
```

Ya creadas las constantes y la propia ventana, pasamos a incluir las variables. Aquí se ha añadido un reloj el cual evitará que la ventana se cierre nada más abrirse y nos permita poder jugar.

También se han añadido aquí todas las imágenes que se usarán como fondo cargándolas previamente en el proyecto. Con el `convert_alpha` lo que estamos realizando es cambiando el formato de pieles de la imagen, adaptándolo a la ventana.

El `bloque_tamano` es utilizado para establecer un tamaño estándar a la manzana que va a aparecer, mientras que los FPS establecen la velocidad de movimiento de la serpiente.

Por último, se establece una fuente para el tipo de letra que se mostrara en el juego.

```
#Funciones
def intro_juego():
    intro = True
    while intro:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_c:
                    intro = False
                if event.key == pygame.K_q:
                    pygame.quit()
                    quit()
```

```
superficie.fill(GOLD)
superficie.blit(fondo, [0, 0]) #Una imagen que queramos
message_to_screen("Bienvenid@ a Jörmundgander", GOLD, -100)
message_to_screen("El objetivo del juego es comerse todas las manzanas,", BLUE2, -50)
message_to_screen("Mientras más manzanas comas, mas grande se hara la serpiente.", BLUE2, -30)
message_to_screen("No te choques con las paredes ni te comas a ti mismo, o perderas", BLUE2, -10)
message_to_screen("Para jugar pulse C. Para salir pulse Q", BLUE2, 100)
pygame.display.update()
reloj.tick(15)
```

Empezando con la primera función del proyecto, nos situamos en la ventana inicial del juego denominada `intro_juego`, donde establecemos que sea verdadera, es decir, que exista y que mientras estemos en esta intro nos de tres opciones, siendo:

- La primera de ellas es cerrar el juego mediante la equis de la esquina superior.
- Iniciar el juego con la pulsación del botón/letra "C"

- Cerrar el juego/ventana con la pulsación del botón/letra “Q”

Si pulsamos la letra “C” de continuar, entonces, `intro_juego` pasará a ser falsa y por lo tanto se cerrará y dará paso a la siguiente ventana.

Si pulsásemos la equis o la letra “Q” cerrará la ventana de inmediato interrumpiendo la reproducción del juego.

La siguiente parte de la función ha sido asignada a la creación e inclusión de un texto inicial con su título, además de varias líneas de texto para explicar las reglas del juego.

Todas estas líneas tienen su posición y color particular para poder distinguirlas bien, además de hacerlas llamativas.

También llamamos a una de las imágenes previamente cargada dentro de las variables.

Una vez perfilada la intro del juego se debe de actualizar para que incluya todo esto en la ventana.

```
def puntos (puntuacion):  
    text = font.render("Puntos: " + str(puntuacion), True, GOLD)  
    superficie.blit(text, [0, 0])
```

La siguiente función ha sido creada para establecer un contador de puntos en la pantalla donde los irá mostrando según vamos comiendo la manzana. Se ha situado en las coordenadas (0,0), es decir, en la esquina superior izquierda (0X, 0Y).

Además, se le ha asignado un color para que resulte más fácil de ver. También se le ha puesto como verdadero, es decir, mientras estemos en esta ventana comiendo las manzanas, los puntos se mostrarán por pantalla.


```
def pausa():
    pausado = True
    while pausado:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_c:
                    pausado = False
                elif event.key == pygame.K_q:
                    pygame.quit()
                    quit()
```

```
superficie.fill(BLUE)
superficie.blit(f_pausa, [0, 0])
message_to_screen("Pausa", BLUE, -100)
message_to_screen("Hay que descansar de vez en cuando", BLUE2, -75)
message_to_screen("Para volver a jugar, pulse C. Para terminar, pulse Q", BLUE2, 25)
pygame.display.update()
reloj.tick(5)
```

Como ya se hizo con la ventana de intro, en la ventana de pausa se le asigna como verdadera mientras esté utilizándose. Se le asignan las mismas teclas para salir y continuar, así como también se le incluye un texto y una imagen de fondo indicando que se encuentra en pausa y que puede elegir entre seguir jugando o acabar la partida y salir del juego.

Más adelante se mostrará el código utilizado para hacer funcionar la pausa, así como las opciones de continuar y salir.

```
def serpiente(bloque_tamano, listaSerp):
    for XandY in listaSerp:
        pygame.draw.rect(superficie, BLACK, [XandY[0], XandY[1], bloque_tamano, bloque_tamano])
```

Para poder crear la serpiente se ha creado una función denominada serpiente en la cual se le asigna una forma, tamaño y color. Aquí se vuelve a llamar a bloque_tamano, estableciendo así su tamaño.

```
def text_objetos(text, color):
    textSuperficie = font.render(text, True, color)
    return textSuperficie, textSuperficie.get_rect()
```

La siguiente función es utilizada para recuperar el texto de la superficie y que lo muestre por pantalla.

```
def message_to_screen(msg, color, y_displace = 0, tamano_letra = "pequeña"):
    textSur, textRect = text_objetos(msg, color)
    textRect.center = (display_ancho/2), (display_altura/2) + y_displace
    superficie.blit(textSur, textRect)
```

Esta función crea el mensaje que se va a mostrar en pantalla: message_to_scren. Siempre que se vaya a introducir un mensaje o texto en la ventana, se recurrirá a esta función.

Se establece que debe posicionarse en el centro de la pantalla, manteniendo la distancia de ambos bordes laterales.

A continuación, se describirá la función principal, aquí denominada gameLoop ():

```
def gameLoop():
    gameExit = False
    gameOver = False

    #Cabeza de la serpiente
    lead_x = display_ancho/2
    lead_y = display_altura/2

    lead_x_cambio = 0
    lead_y_cambio = 0

    listaSerp = []
    largoSerp = 1

    #Lugar y tamaño de la manzana
    azarManzana_x = round(random.randrange(0, display_ancho - bloque_tamano)/10.0)*10.0
    azarManzana_y = round(random.randrange(0, display_altura - bloque_tamano)/10.0)*10.0
```

Para empezar, establece que el gameExit (salir del juego) y gameOver (perder la partida) sean falsas y permanezcamos en el juego.

Después se establece la posición en la que aparecerá la cabeza de la serpiente, siendo esta el primer bloque de serpiente del juego. También se indica su tamaño inicial de un bloque.

Una vez establecido esto, se posiciona la manzana en la ventana, indicando que deberá aparecer aleatoriamente en el mapa con tamaño establecido y con un tamaño calculado en función del tamaño deseado. Haremos función de esto más adelante.

```

while not gameExit:
    while gameOver == True:
        superficie.fill(VIOLET)
        superficie.blit(fondogameover, [0, 0]) #Una imagen que queramos
        message_to_screen("Para volver a jugar, pulse C. Para terminar, pulse Q", BLACK, -100)

        pygame.display.update()

    for event in pygame.event.get():
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_q:
                gameExit = True
                gameOver = False
            if event.key == pygame.K_c:
                gameLoop()

```

Siguiendo dentro del gameLoop, estableceremos la pantalla de gameOver que saldrá cuando la serpiente colisione con uno de los bordes de la ventana o se coma así mismo.

Para la pantalla de gameOver se establece un texto, su color y las funciones de volver a jugar o salir del juego con las teclas "C" y "Q". Haciendo que si se presiona la "C" de volver a jugar se reinicia el gameLoop, y si presionamos la "Q" el gameExit se vuelve verdadero y cierra el juego.

```

#Movimiento de la Serpiente
for event in pygame.event.get():
    if event.type == pygame.QUIT:
        gameExit = True
    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_LEFT:
            lead_x_cambio = -bloque_tamano
            lead_y_cambio = 0
        elif event.key == pygame.K_RIGHT:
            lead_x_cambio = +bloque_tamano
            lead_y_cambio = 0
        elif event.key == pygame.K_UP:
            lead_y_cambio = -bloque_tamano
            lead_x_cambio = 0
        elif event.key == pygame.K_DOWN:
            lead_y_cambio = +bloque_tamano
            lead_x_cambio = 0
        elif event.key == pygame.K_p:
            pausa()

```

También, dentro de gameLoop, establecemos las teclas a utilizar para mover la serpiente, así como la tecla que pausara el juego.

```

#Cierre del juego si la serpiente toca uno de los bordes
if lead_x >= display_ancho or lead_x < 0 or lead_y >= display_altura or lead_y < 0:
    gameOver = True

lead_x += lead_x_cambio
lead_y += lead_y_cambio

superficie.blit(fondojuego, [0, 0]) #Una imagen que queramos

tamanoManza = 25
pygame.draw.rect(superficie, RED, [azarManzana_x, azarManzana_y, tamanoManza, tamanoManza])

cabezaSerp = []

```

Estableceremos el gameOver cuando la serpiente toque uno de los bordes en la primera línea del código.

En las dos siguientes se establece el giro de la serpiente, es decir, el cambio de sentido.

Establecemos el que será el fondo del gameLoop o fondo principal del juego llamando a una imagen previamente cargada.

Posteriormente establecemos el tamaño de la manzana que se dividirá con la función explicada pasos atrás. Además, se indica su color.

Por ultimo creamos la función cabezaserp que utilizaremos más adelante para incrementar la longitud de la serpiente.

```
#añadimos una cabeza más cuando coma la manzana
cabezaSerp.append(lead_x)
cabezaSerp.append(lead_y)
listaSerp.append(cabezaSerp)
if len(listaSerp) > largoSerp:
    del listaSerp[0]

for eachSegment in listaSerp[:-1]:
    if eachSegment == cabezaSerp:
        gameOver = True

serpiente(bloque_tamano, listaSerp)
puntos(largoSerp -1)
pygame.display.update()
```

Con estas líneas de código pedimos que la serpiente aumente en un bloque cada vez que se come la manzana y que en caso de comerse a sí mismo gameOver se convierta en verdadero y aparezca su ventana.

Cerramos con un update para que actualice.

```

#Hacemos que se genere en un nuevo sitio al azar cuando pasamos sobre ella.
if lead_x >= azarManzana_x and lead_x <= azarManzana_x + tamanoManza:
    if lead_y >= azarManzana_y and lead_y <= azarManzana_y + tamanoManza:
        azarManzana_x = round(random.randrange(0, display_ancho - bloque_tamano)/10.0)*10.0
        azarManzana_y = round(random.randrange(0, display_altura - bloque_tamano)/10.0)*10.0
        largoSerp += 1 #Agrega un bloque más a la serpiente

reloj.tick(FPS)

pygame.quit()
quit()

intro_juego()
gameLoop()

```

Como se indica en la imagen con esta función generamos la manzana en un punto aleatorio de la ventana.

Una vez indicado esto cerramos el gameLoop, así como el Pygame.

Diseño de interfaz:

Se ha elegido este diseño de la interfaz dado que se ha intentado crear un diseño sencillo de entender, pero que al mismo tiempo intente no ser pesado o complejo.

A cada línea de texto se le ha asignado su lugar en función de lo que informe esa misma línea.

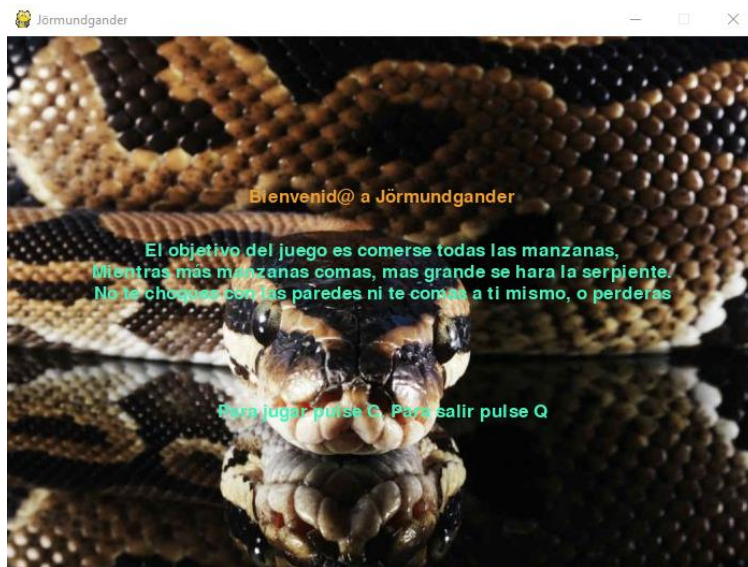
Respecto de los fondos de pantalla, se han elegido en función al contexto de la propia pantalla en cuestión, intentado seguir una misma línea de desarrollo.

Los colores de los textos fueron seleccionados en función a los fondos de pantalla para que se pudiesen distinguir sin problemas, pero también para que no desentonasen con ellos.

Manual de Usuario – User Manual

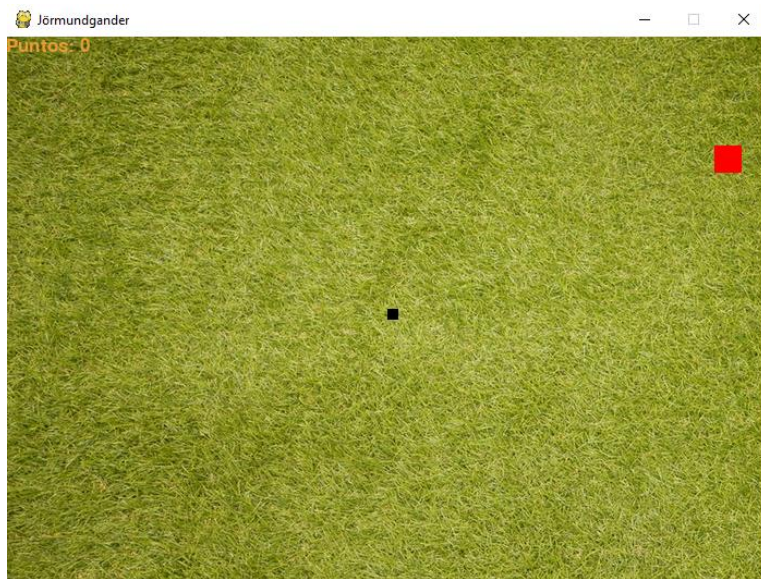
A continuación, se detallará lo que debe hacer el usuario a la hora de utilizar el juego:

Next, it will be detailed what the user must do when using the game:



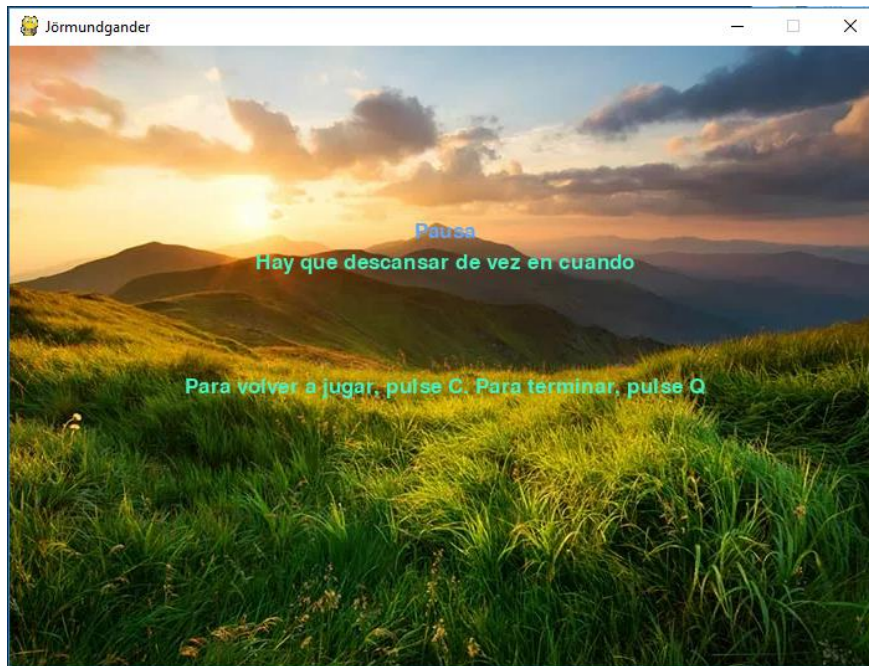
Al iniciar el juego, saldrá esta ventana, en la cual nos darán unas pequeñas instrucciones y dos opciones: "C" para jugar y "Q" para salir. Pulsamos en la "C". Y esto nos llevará a la pantalla de juego.

When starting the game, this window will appear, in which we will be given some small instructions and two options: "C" to play and "Q" to exit. Click on the "C". And this will take us to the game screen.



En esta pantalla se nos mostrara la serpiente (Centro imagen) que moveremos, la manzana que deberemos comer (bloque rojo) y los puntos en la esquina izquierda superior. Para movernos simplemente hemos de utilizar las flechas del teclado.

On this screen we will be shown the snake (Center screen) that we will move, the apple that we must eat (red block) and the points in the upper left corner. To move we simply use the arrows on the keyboard.



Si deseásemos una pausa, bastaría con pulsar la “P” y el juego se pausaría. Dándonos a elegir entre volver a jugar presionando la “C”, o salirnos del juego con la “Q”.

If we wanted a pause, just press the "P" and the game would pause. Giving us a choice between playing again by pressing the "C", or leaving the game with the "Q".

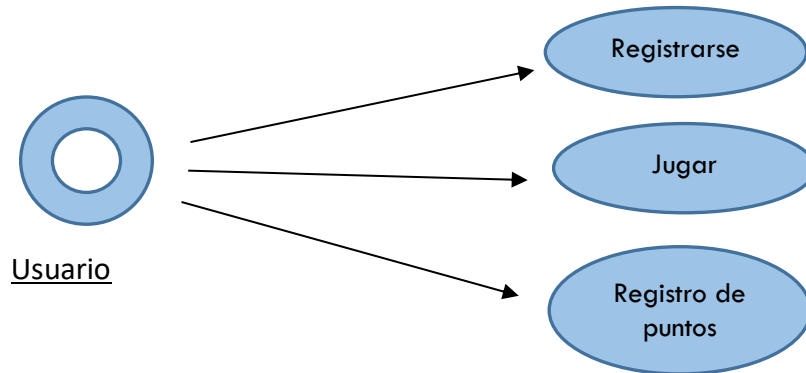


Por ultimo tendríamos la ventana de game over. Esta nos saldría en caso de que chocásemos la serpiente con uno de los límites de la pantalla o nos comiésemos a nosotros mismos. Para volver a jugar, deberíamos pulsar la "C", o, por el contrario, si deseásemos salir del juego tendríamos que pulsar la "Q".

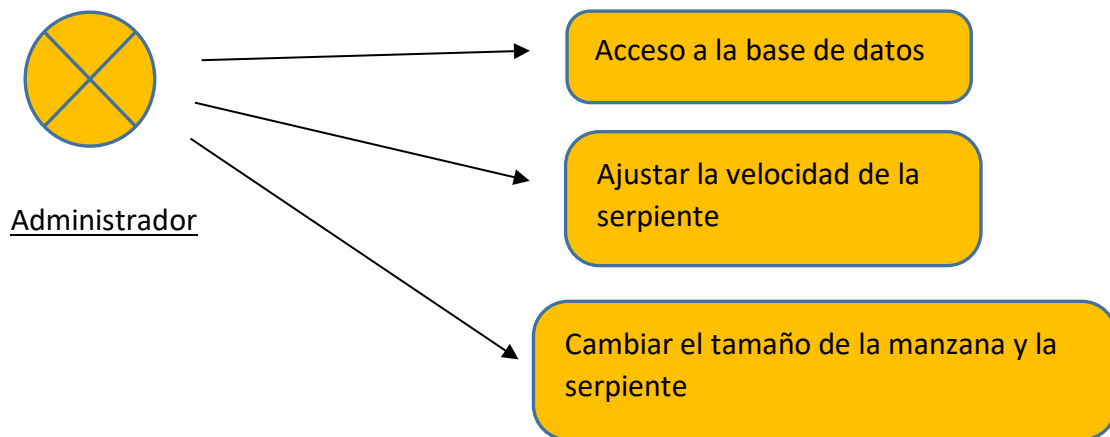
Finally, we would have the game over window. This would come out in case we hit the snake with one of the limits of the screen or we ate ourselves. To play again, we should press the "C", or, conversely, if we wanted to quit the game we would have to press the "Q".

Casos de Uso

Frontend:



Backend:



Conclusiones

Como conclusiones a este proyecto se puede extraer que durante el desarrollo de este videojuego se han mejorado los conocimientos existentes sobre Python, así como se han aprendido nuevos conocimientos y se han podido poner en práctica. Han servido para mejorar la habilidad de autoaprendizaje debido a que muchas partes del código han sido elaboradas con los conocimientos adquiridos durante el desarrollo de este proyecto. También ha servido para adquirir la capacidad de aprender de tus propios errores y solucionarlos uno mismo.

Bibliografía

- <https://www.programoergosum.com>
- <https://es.stackoverflow.com>
- <https://www.pygame.org/docs>
- <https://openwebinars.net/cursos/bases-de-datos/>
- <https://www.youtube.com/user/pildorasinformaticas>
- <https://programacionpython80889555.wordpress.com/2018/12/04/creando-login-con-python-y-tkinter/>