

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

UNIDAD DE SERVICIOS DE CÓMPUTO ACADÉMICO

FUNDAMENTOS DE PROGRAMACIÓN EN PYTHON

SNAKE (DOCUMENTACIÓN)

CREADO POR:

CANTÚ DE GABRIEL JOSUÉ FERNANDO

COLÍN SOFÍA LOPEZ

MARTÍNEZ MARTÍNEZ ANDRES

SEMESTRE 2019-2

Documentación

Introducción

Snake (a veces también llamado la serpiente) es un videojuego lanzado a mediados de la década del 70 que ha mantenido su popularidad desde entonces, convirtiéndose en un clásico. En 1998, el Snake obtuvo una audiencia masiva tras convertirse en un juego estándar pre-grabado en los teléfonos Nokia.

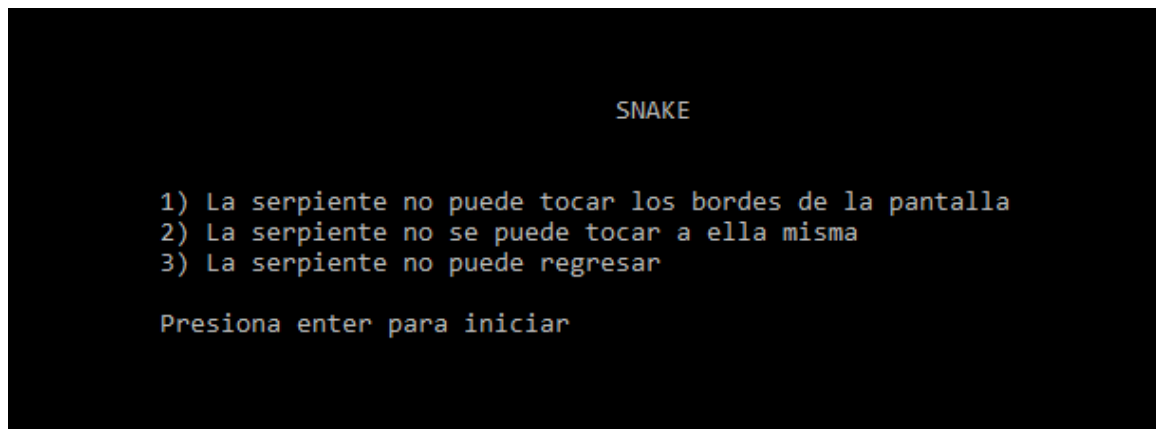
Descripción del juego

Funcionamiento

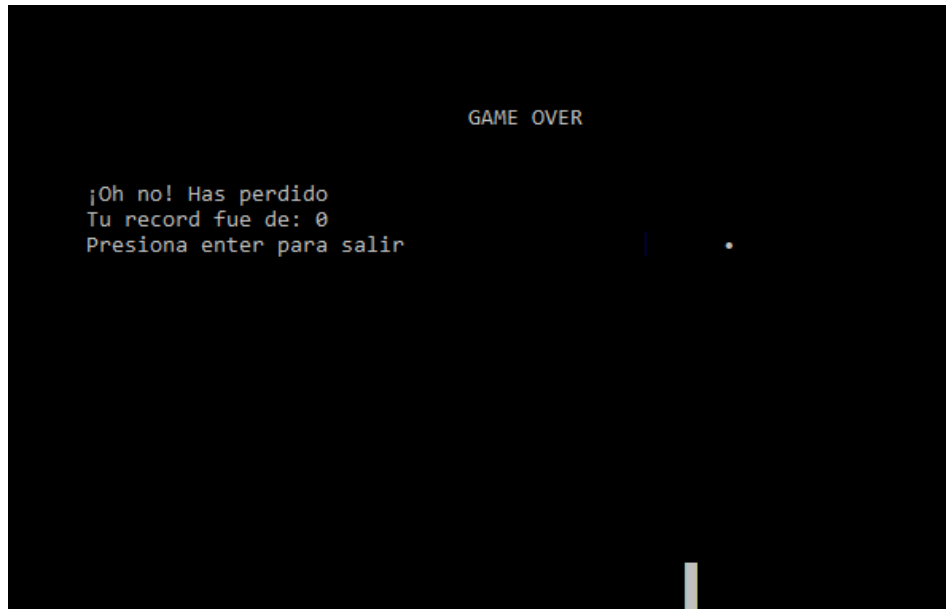
Snake consiste en una línea (análoga a una serpiente o a un gusano) conformada por una serie de puntos (o pequeños conjuntos) que siempre están en movimiento. La serpiente sólo puede moverse en 4 direcciones (90° entre cada dirección) que el jugador es capaz de controlar comúnmente a través de un teclado, esto implica que sólo se necesitan cuatro teclas para jugar. El jugador debe “comer” comidas que aparecen dentro del juego, para ello deberá desplazarse hasta su ubicación y “pasar” por encima de ellas. Cada vez que la serpiente coma, crecerá de tamaño una unidad. El objetivo del juego es conseguir que la serpiente crezca el máximo tamaño posible sin que choque contra ella misma o con los límites del mapa en donde se desplaza.

Reglas del juego

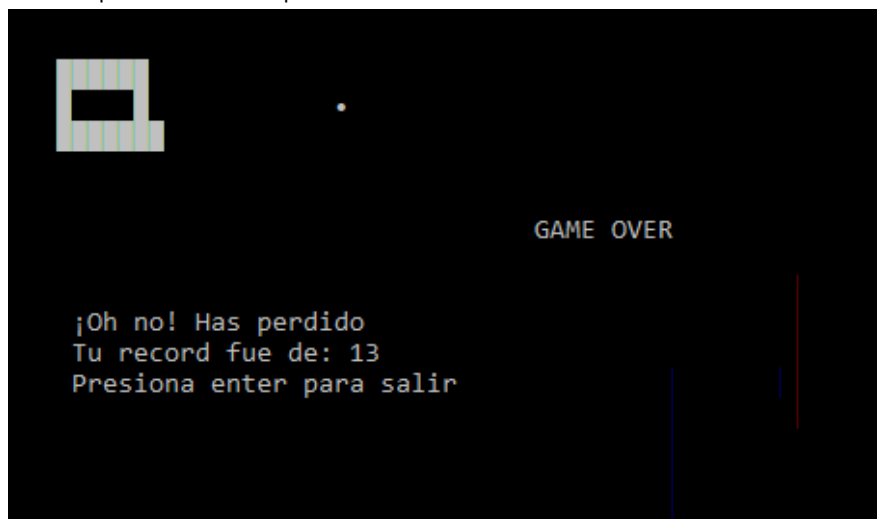
Una vez abierto el programa, se muestran en pantalla tres reglas a seguir para evitar perder el juego:



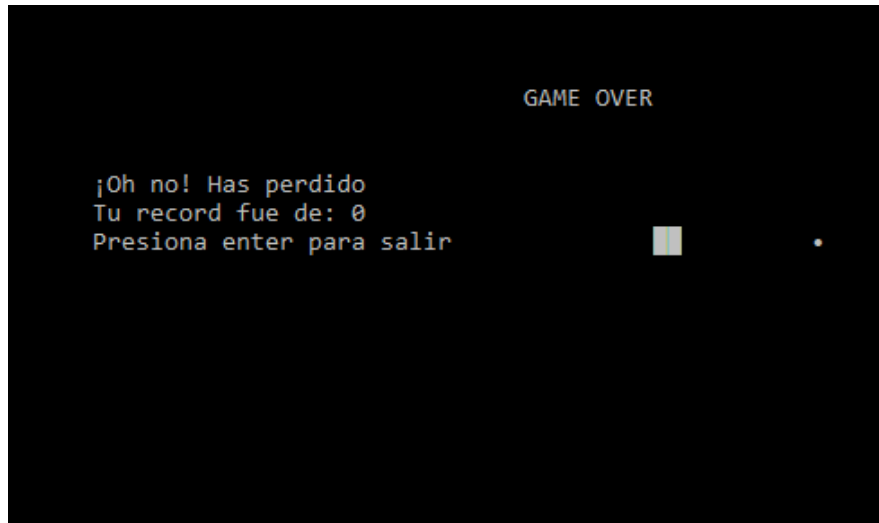
- 1) La serpiente no puede tocar los bordes de la pantalla.



- 2) La serpiente no se puede tocar a sí misma.



- 3) La serpiente no puede retroceder (p.e. Se intenta ir hacia la izquierda cuando la serpiente avanza hacia la derecha.).



Además, el usuario sólo podrá presionar las teclas de movimiento para direccionar a la serpiente, puesto que si intenta presionar alguna otra tecla, perderá.

Para poder ganar el juego el usuario debe “comer” o conseguir la mayor cantidad de “comidas” o puntos. Cada vez que la serpiente consiga un punto crecerá de tamaño una unidad (separadas entre sí por una línea verde). Por lo tanto, el jugador debe conseguir que la serpiente obtenga el máximo tamaño posible.

Implementación del programa

Descripción del algoritmo empleado

Considerando que la serpiente es una estructura de tamaño variable y elementos de dos coordenadas, se decidió utilizar una lista cuyos elementos fueran también listas para poder almacenar cada coordenada de cada elemento. Inicialmente la serpiente comienza en a moverse en un punto y con una dirección por defecto, definido gracias a las propiedades del módulo *curses*.

Para el elemento que representa a la comida, se utilizó una lista que también guardara las dos coordenadas de ésta.

El funcionamiento del juego se basa en iteraciones, por lo que se decidió utilizar un ciclo `while`.

Dentro del ciclo se ejecutan varias instrucciones:

- Leer la tecla que el usuario presiona (si es que presiona): si la tecla presionada corresponde a una dirección diferente a la que en ese

momento lleva la serpiente, se modificará su curso. En caso de que sean iguales la serpiente lleva la misma dirección.

- Verificar que ambas coordenadas de la cabeza de la serpiente no sean iguales a las coordenadas de cualquier punto de los límites del mapa, o a las coordenadas de cualquier elemento de la serpiente. Esto implica que la serpiente no debe retroceder porque entonces la cabeza tendría las mismas coordenadas que el elemento inmediato anterior. En el caso de que algunas de las condiciones anteriores se cumpliera, se muestra el score que logró el jugador y se termina el ciclo.
- Asignar una nueva dirección insertando un nuevo elemento al principio de la lista Snake: se crea un nuevo elemento que tenga las mismas coordenadas que la cabeza de la serpiente y dependiendo de la tecla presionada, se suma o resta una unidad a la coordenada horizontal (dirección hacia la derecha o hacia la izquierda) o a la coordenada vertical (dirección hacia arriba o hacia abajo).

Finalmente, se inserta el elemento al principio de la serpiente mediante la función *insert()*, que inserta un elemento al principio de una lista si no se especifica la posición de éste.

- Asignar nuevas coordenadas a la comida en el caso de que la serpiente se la haya comido: si las coordenadas de la cabeza son iguales a las coordenadas de la comida, aparte de sumarle una unidad al score del jugador, se generan aleatoriamente dos nuevas coordenadas para la nueva posición de la comida. No obstante se debe utilizar un ciclo para evitar problemas, puesto que al ser aleatorias las coordenadas pueden coincidir con las coordenadas de los elementos de la serpiente. También se incrementa un elemento el tamaño de la serpiente. Finalmente, se muestra en pantalla la comida.

En caso de que la serpiente no coma, se expulsa el último elemento de la serpiente mediante la función *pop()* que devuelve el último elemento de una lista si no se especifica la posición de éste.

El ciclo termina mostrando en pantalla el último elemento que se le agregó a la serpiente.

Descripción de las funciones

Para mostrar en pantalla, saber que tecla presiona el usuario, modificar el tamaño de la ventana, entre otras cosas, se utilizan funciones del módulo *curses*.

Por medio de la función *getmaxyx()* se obtiene el tamaño de la ventana de la consola.

Con la función *newwin(alto_max, ancho_max, alto_min, ancho_min)* se crea una ventana.

La función *randint()* devuelve un número entero incluido entre los valores indicados. Los valores de los límites inferior y superior también pueden aparecer entre los valores devueltos.

Para iniciar *curses* se utiliza la función *initscr()* que devuelve un objeto de ventana que representa la pantalla completa.

Habilitando el modo de teclado *keypad(1)* se pueden obtener teclas especiales, como las teclas de cursor o las teclas de navegación, como *Av Pág e Inicio*, como una secuencia de escape multibyte, devolviendo un valor especial como por ejemplo *curses.KEY_LEFT*.

Para revertir la configuración de terminal de *curses* se utiliza la función *endwin()* para restaurar el terminal a su modo de operación original.

addstr() permite mostrar en la ventana cadenas de caracteres, para ello se le pasan a la función las coordenadas y la cadena.

Mediante *getch()* se lee la tecla presionada por el usuario.

Descripción de las bibliotecas de Python

Curses: La biblioteca curses proporciona una herramienta de manejo de pantalla y pintura de pantalla independiente de terminal para terminales basadas en texto. Los terminales de pantalla admiten varios códigos de control para realizar operaciones comunes, como mover el cursor, desplazar la pantalla y borrar áreas.

Random: El módulo random de la librería estándar de Python incluye un conjunto de funciones que permiten obtener de distintos modos números pseudoaleatorios.

Conclusiones

Uno de los mayores retos en los juegos son las interacciones con el usuario. Pensar en todas las posibles entradas que pueden hacer colapsar el programa y restringirlas es un proceso que tarda un poco de tiempo.

Para poder tener un mejor control de un programa siempre es mejor utilizar ciclos combinados con condicionales puesto que extienden las posibilidades y restringen la libertad del usuario cuando es necesario.