Proyecto Semestral Programación I
Red Bounce Ball

Luciano Ignacio Revillod Jerez

Ingeniería Civil Informática

Programación I

Mayo 2022

Índice:

Introducción

Para crear un videojuego se necesitan múltiples conocimientos y herramientas, entre ellas, un lenguaje de programación.

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general. La sintaxis de python es bastante intuitiva lo que hace a este lenguaje muy amigable al aprendizaje.

Python cuenta con múltiples librerías, estas facilitan e incorporan múltiples funciones dentro del lenguaje.

> "*En informática, una librería es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca .*"
>
> *Wikipedia.org*

Para programar "Red Bounce Ball" se utilizó principalmente la librería de python pygame, esta permite trabajar con imágenes, objetos, sonidos, etc.

Manual de usuario

1) Ejecución y Requisitos:

a. Para ejecutar correctamente el código se necesita:

b. Un editor de texto capaz de ejecutar Python.
Se recomienda **Visual Studio Code.**

c. Una versión actualizada de **Python (3.x)**.

d. La librería de Python **“Pygame”**
en una versión compatible con Python3.

e. Un teclado que posea las teclas de posición
(arriba, abajo, izquierda y derecha).

2) Objetivo y Gameplay:

a. El código consiste en un juego de dos dimensiones
basado en plataformas, un jugador y colisiones.

b. El objetivo principal del jugador es llegar al final del mapa.

c. El jugador posee 3 vidas (intentos).

3) Movilidad:

a. El jugador tiene las mismas capacidades de movimiento que los clásicos juegos en dos dimensiones (2D).

b. Dentro del mapa existen múltiples obstáculos, estos buscarán impedir el avance del jugador, al tocar uno de estos obstáculos el jugador perderá una vida equivalente a un intento.

c. Los movimientos tienen asignados las siguientes teclas:

1. **Saltar:** Tecla *“ K_UP ”* (Flecha arriba).
2. **Derecha:** Tecla *“ K_RIGHT ”* (Flecha derecha).
3. **Izquierda**: Tecla *“ K_LEFT ”* (Flecha izquierda).
4. **Salto a la derecha:** Combinación de teclas:
““K_UP ” + “ K_RIGHT ”.
5. **Salto a la izquierda:** Combinación de Teclas:
““K_UP ” + “ K_LEFT ” .

Diagrama de casos de uso

Un diagrama de casos de usos muestra de manera visual las distintas funciones que puede realizar un usuario en un sistema de información, en este caso un programa.

Para construir este diagrama se utilizó la herramienta de modelado online “Lucidchart”, con el diagrama se busca visualizar las funcionalidades y casos de uso que posee quien ejecute el programa (Jugador) y quien haya creado el programa (Programador).

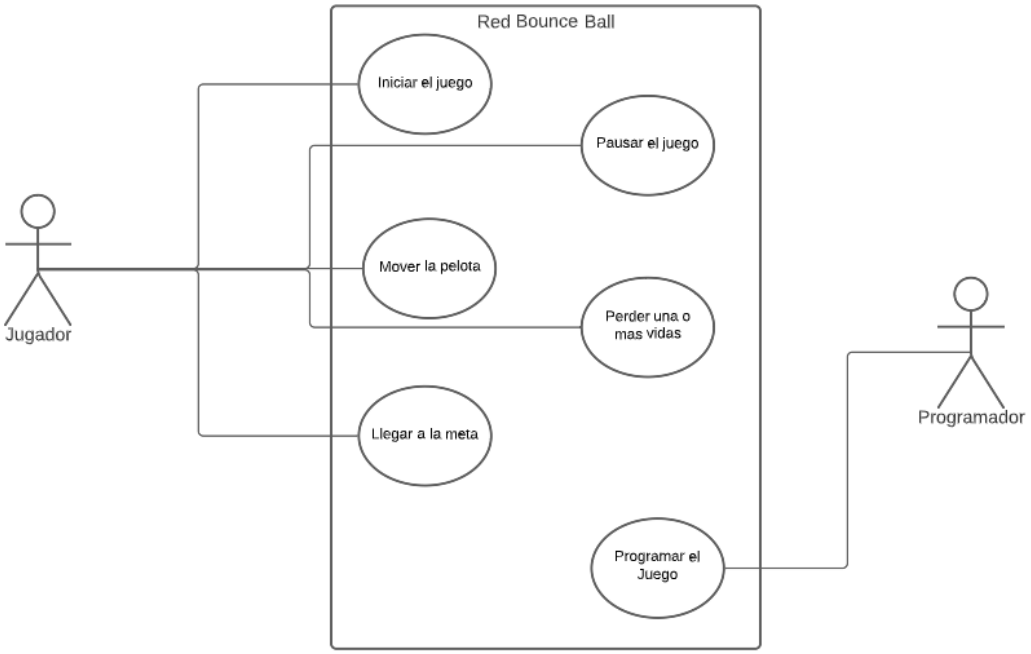
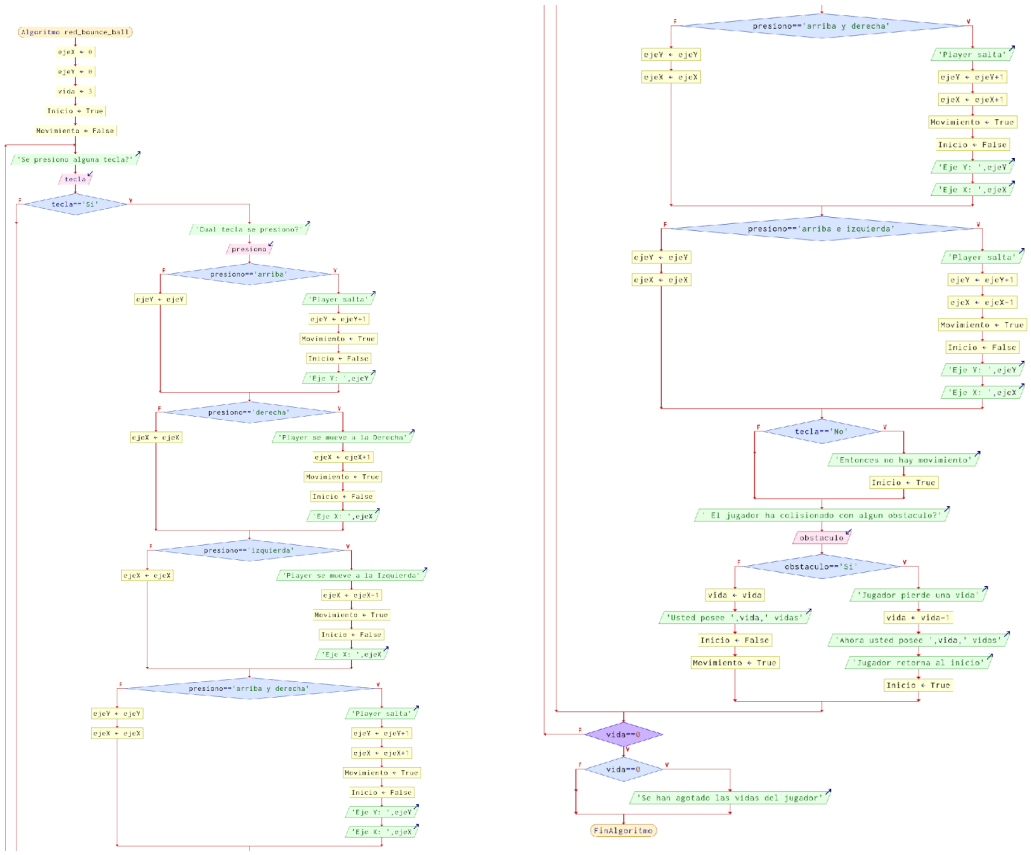


Diagrama de flujo

¿Qué es un diagrama de flujo?

> *Un diagrama de flujo es un diagrama que describe un proceso, sistema o algoritmo informático. Se usan ampliamente en numerosos campos para documentar, estudiar, planificar, mejorar y comunicar procesos que suelen ser complejos en diagramas claros y fáciles de comprender.*
>
> *Lucidchart.com*

Para Red Bounce Ball el diagrama consiste en los distintos casos que se pueden presentar en la ejecución del programa, por ejemplo: presionar una tecla, esta acción provocara un cambio en el programa.



Imágenes referenciales del diagrama

Archivo .psc original adjuntado como “red_bounce_ball.psc”

Librerías

El proyecto “Red Bounce Ball” consiste en un código basado en Python y su librería Pygame.

> *“Pygame es un conjunto de módulos de Python diseñados para escribir videojuegos. Pygame añade funcionalidad a la excelente biblioteca de SDL. Esto le permite crear juegos con todas las funciones y programas multimedia en el lenguaje Python”.*
>
> *pygame.org*

Para utilizar esta librería y todas sus funciones se debe realizar el siguiente procedimiento:

```
Import pygame

From pygame.locals import *
```

Este texto debe ir, de preferencia al inicio del código. Cabe recalcar para importar cualquier otra librería se recomienda seguir el orden indicado.

Ejemplo:

main.py > ...

```
1   import pygame,world,Redball
2   from world import *
3   from pygame.locals import *
4   from Redball import Redball
5
6   #=============================================================================================#
7   #                                  Definiciones Globales                                      #
8   #=============================================================================================#
9
10  minX = 0      ; maxX = 800                    ;         xd = 0           ;          yd = 0
11  #Valor minimo y maximo del eje X en el mapa.    #Valor asignado a 0 en x       #Valor asignado a 0 en Y
12  minY = 0      ; maxY = 800                    #se puede usar para sumar      #se puede usar para sumar
13  #Valor minimo y maximo del eje Y en el mapa.    #un numero y asi cambiar       #un numero y asi cambiar
14  #                                             #el valor del eje x.           #el valor del eje y.
15
16  ToF = True
17  #Valor True, usado principalmente para crear un ciclo while infinito.
18
19  res = (maxX,maxY)
20  #Resolucion de la ventana.
21
22  tileX = 128           ;             tileY = 128
23  #Valores en X e Y del tile principal, a futuro se reescalara.
24
25  tileSize = 20
26  #valor rescalado de todos los sprt.
27
28  mouX = 0                    ; mouY = 0
29  #Valor asignado a 0 en x       #Valor asignado a 0 en Y
30  #se puede usar para sumar      #se puede usar para sumar
31  #un numero y asi cambiar       #un numero y asi cambiar
32  #el valor del eje x(MOUSE).    #el valor del eje y(MOUSE).
33
34  Timer1 = 10
35
```

Funciones:

Para el proyecto se utilizaron múltiples funciones incorporadas en pygame, tales como:

- **pygame.init ( ):** Función que se encarga de iniciar todos los módulos de la librería pygame.
- **pygame.mixer.init ( ):** Inicia el reproductor de audio de pygame.
- **pygame.display.set_caption ( ):** Esta función recibe un str de entrada. El string ingresado será el nombre de la ventana de nuestro juego.
- **pygame.mouse.set_visible(True, False):** Visibilidad del mouse en la ventana principal.
- **pygame.display.set_mode(**resolución**):** Función que recibe la resolución de la ventana a crear.
- **pygame.event.get ( ):** Identificador de eventos dentro del ciclo, un ejemplo de evento sería clickear la X (QUIT).
- **pygame.time.delay ( ):** Esta función se utilizó para solucionar un problema de velocidad y tiempo, básicamente ralentiza las acciones en milisegundos, en el caso del códigos son 25 ms, de esta forma el jugador se mueve un poco menos fluido, pero es más controlable.
- **pygame.display.flip ( ):** Actualiza la pantalla.
- **pygame.transform.scale ( ):** Función que reescala imágenes, recibe dos parámetros de entrada, una imagen/sprite y el tamaño en píxeles a reescalar. Se utilizó en las clases Redball y World.
- **pygame.key.get_pressed ( ):** Función que verifica si se presiona alguna tecla. En el código se usó para mover al jugador.
- **get_rect ( ):** Función de pygame que captura la “hitbox” de un determinado objeto/ sprite, en el código se usó en las clases world y Redball, esto para obtener las hitbox de los sprt y poder asignarles colisión en el mapa.

Conclusión

Python es un lenguaje muy versátil, ya que mediante su librería pygame nos permite crear programas con interfaz gráfica e interacciones con el usuario, si bien no es la forma más óptima de crear un videojuego, nos ayuda a conocer y entender a grandes rasgos cómo funcionan los distintos elementos de un videojuego.

En el proyecto se utilizaron distintas funciones, con ellas se buscó optimizar y realizar de la mejor manera todas las características del juego. Además el código del programa se encuentra ampliamente comentado, de esta forma se busca que cualquier persona con conocimientos en programación pueda entender cada sección del código.