Projecte de Big Data - Creació d'un joc amb Pyxel

Nil Jimeno, Bernat Brucet

14 de maig de 2024

todo:

- Justificación del proyecto: explicación de las necesidades (personales y/o sociales) a las que viene a responder el proyecto.
- Explicación de la arquitectura de la aplicación desarrollada.
- Referencia a los materiales de terceros utilizados: origen de los fragmentos de código ajenos, del material multimedia incluido, de la información técnica consultada, etc.
- Lista de propuestas de mejora de la aplicación: corrección de errores (fixes), posibles nuevas funcionalidades, escalabilidad de la aplicación...

1 Justificació del projecte

El nostre projecte s'ha fet amb la intenció de posar en pràctica coneixements relacionats amb Python, per entendre millor la llògica del llenguatge de programació i aprendre i posar en pràctica diferents llibreries. A part de l'interès personal, està centrat principalment en aprendre tant Python com a llenguatge de programació com l'ús i interacció de llibreries, no té cap objectiu més enllà del coneixement obtingut.

1.1 Llógica abans d'abstracció

El projecte que hem fet està centrat en la llibreria Pyxel. Un dels punts que la distingeix per sobre d'altres llibreries, com Pygame, és la poca abstracció que utilitza. D'aquesta manera, el programa està molt més centrat en la llògica del programa i l'ús d'estructures apropiades per posar-lo en pràctica.

Encara que també hem fet ús d'altres llibreries, la base del projecte es basa molt més en el llenguatge en si que no pas en funcions de llibreries específiques, així els coneixements obtinguts no son tan situacionals.

1.2 Llibreries

Encara que pyxel no se centra en l'ús de llibreries externes, és important aprendre a utilitzar-les perque Python està molt centrat en el seu ús.

1.2.1 Pyxel

És la llibreria principal. S'encarrega de controlar que el programa s'executi en bucle de manera controlada, crear la finestra de l'aplicació i mostrar gràfics per pantalla.

Encara que sigui per Python, la llibreria té un rendiment acceptable perque ha estat programada en Rust. Requereix poc coneixement de l'abstracció de la llibreria ja que es basa en unes poques funcions simples.

1.2.2 Sqlite3

És un software d'administrador de bases de dades relacionals a nivell local, per la qual cosa s'utilitza soving en aplicacions. El programa està escrit en C. S'encarrega d'administrar algunes dades.

Sqlite és útil per aplicacions que no han d'interactuar amb un servidor. Utilitza funcions simples per executar les queries, que funcionen de manera molt similar a MariaDB.

1.2.3 Flask

Framework simple fet en Python, per projectes poc professionals/tests. S'encarrega de mostrar una pàgina amb les puntuacions màximes.

2 Arquitectura del codi

2.1 Estructura principal

La llibreria Pyxel fa molt d'ús d'estructures de classes. Aquestes se separen en tres funcions, per convenció (i, en el cas de la classe principal, perque la llibreria ho requereix):

- __init__: S'executa al instanciar la classe, es pot tornar a executar manualment. En aquesta funció s'inicialitzen totes les variables inicials amb el seu valor corresponent (en el cas de la classe principal, també s'inicia pyxel).
- update: S'executa cada torn. Per convenció, aquí s'executa tot el que té a veure amb el funcionament intern del programa i acualitzar les variables.
- draw: S'executa cada torn. Per convenció, aquí s'executa tot el que té a veure amb gráfics de Pyxel.

En el cas del nostre joc, utilitzem una classe inicial que s'encarrega d'executar les funcions update() i draw() de la classe de l'entorn que volem executar en cada cas (section).

Els entorns s'encarreguen també d'executar les funcions update() i draw() de cada classe interna que ho necessiti. Aquests estan separats perque fan funcions completament diferents, ja que, per exemple, el menú és completament independent de la secció del joc.

La majoria d'entorns ténen un funcionament simple i intuitiu, a excepció de l'entorn del joc.

2.2 Entorn: game.py

Aquest és l'entorn principal del joc. S'encarrega d'actualitzar events, gráfics i entitats del joc.

2.2.1 Llistes d'entitats

Les entitats del jugador es guarden com a variable individual, ja que només ténen una instància. Les entitats que ténen més instàncies s'emmagatzemen en llistes, les quals són iterades posteriorment per actualitzar cada una de les entitats guardades.

Hi ha dues llistes d'entitats: la d'enemics actius i la d'enemics inactius (morts). Quan un enemic mor, aquest es mou a la llista d'enemics morts fins que arribi el seu moment de reaparèixer. Cada llista executa funcions diferents de les entitats que conté.

2.2.2 Col·lisions

Per col·lisionar les entitats enemigues amb la classe del jugador o l'espasa, s'itera cada enemic actiu i es comprova la distància entre aquests. En cas de col·lisió amb la classe espasa, l'entitat enemiga va a la llista d'enemics morts, però no sense abans haver comprobat la col·lisió amb el jugador, que treuria vida.

2.2.3 Sistema d'events

Per administrar variables relacionades amb temps i contadors, s'utilitzen les classes rellotge (per calcular temps i events d'aparicions) i contador (per calcular la quantitat d'enemics actius, restants, i variables de la ronda).

2.3 Entitats

Les entitats son classes amb variables i funcions. Segueixen la convenció de pyxel en el funcionament de les funcions principals, pel que consisteixen en tres funcions principals: __init__, update i draw.

Les seves variables més importants són els vectors de posició, ja que són necessaris per saber a on imprimir la entitat i comprovar col·lisions. La majoria de la resta del funcionament és llógic i depèn únicament de la classe en si.

3 Materials de tercers utilitzats

3.1 Programes:

- Python3 :
- Pyxel
- Sqlite
- Flask

3.2 Documentació:

Per Python

- Enumerates: utilitzat en tots els usus d'enumerates.
- Printar integer en dos dígits: utilitzat per formatejar el string del contador d'enemics restants.
- Generaci'o de números pseudo-aleatoris: utilitzat pels usos de randint.

Per Pyxel

- Example 01 - hello world: per usos bàsics de la llibreria: execució de la finestra, printar text, netejar pantalla. També s'ha copiat el codi del text multicolor d'aquí.
- Example 09 - plattformer: per col·lisions (are_nearby) i ús de posicions i velocitat.

Per Sqlite3

- Tutorial d'sqlite per python: per les funcions per utilitzar sqlite a python.
- Insertar a taules: per insertar valors a les taules a sqlite.
- ROWID: per seleccionar una fila concreta per la seva posició.
- Tutorial Inner Join: per trobar l'ID dels enemics.

4 Possibles millores