

Creación de Juego en python Y JAVASCRIPT

Programación orientada a objetos



11 de January de 2023

Integrantes:

Ignacio mamani

Milovan sorich

Josue vilches

Contenido

Tabla de contenido

[1. Introducción 0](#_Toc9800)

[2. Desarrollo 1](#_Toc11552)

[LENGUAJE PYTHON 1](#_Toc2639)

[Características: 1](#_Toc32141)

[Librerías utilizadas: 1](#_Toc9124)

[LENGUAJE JAVASCRIPT 3](#_Toc4444)

[Características: 3](#_Toc18924)

[Librerías utilizadas: 3](#_Toc30207)

[3. Conclusión 4](#_Toc8111)

# Introducción

En este informe lo que se pretende conseguir es mostrar el uso y la aplicación de las herramientas impartidas en las clases de Programación Orientada a Objetos, por lo cual se creo un juego que incluyera ciertos pilares de la programación orientada a objetos, en los siguientes lenguajes:

* Python
* JavaScript

Entonces, siguiendo con la creación del juego, primero se hace en Python y luego se pasa a JavaScript.

También se debe investigar el uso de librerías para un mejor desarrollo del juego y esto también sirve para una mejor presentación al momento de ejecutar el juego. Si en el caso de que ocurriera errores o bugs al momento de ejecutar el juego está permitido el uso de la IA conocida como ChatGPT, la cual nos ayudara a optimizar el código y mejorar el rendimiento de dicho proceso, si bien la IA puede ser muy útil al momento de solucionar errores, hay que recordar que al momento de programar todos pensamos de diferentes maneras o caminos para llegar a la solución, por lo cual la ayuda del CHATGPT solo debe tomarse como una sugerencia y no como un hecho irrefutable.

# Desarrollo

## LENGUAJE PYTHON

### Características:

* Lenguaje interpretado
* Tipeado dinámico
* Lenguaje de alto nivel
* Parcialmente orientado a objetos

### Librerías utilizadas:

* Random
* Time

El juego a desarrollar en este caso va a ser de enfrentamiento jugador 1 vs jugador 2 y el primer lenguaje a ocupar en este caso fue python dado que es el que mas dominamos, partimos utilizando la clase padre”character” y le pasamos sus respectivos parámetros como: “name”,”hp”,”attacke”,”defense”, y desde ahi empezamos a añadir métodos segun las parámetros que utilizamos, como: método “attacke”, “showcase”, “life” y por ultimo el método “combate” que es el mas importante para el procesamiento del juego, hay que mencionar que también añadimos la librería random en python para hacer los métodos como “attacke” y “combate” mas dinámicos y también la librería time para que de esta manera el juego no sea tan lineal. Luego de añadir la clase padre y sus métodos comenzamos a añadir mas pilares de POO, empezamos incluyendo herencia dado que debíamos crear personajes para que el jugador 1 y el jugador 2 escogieran sus respectivos personajes, creamos los siguientes personajes:

-Warrior

-Mage

-God

-Spartan

A estos personajes les dimos sus respectivos parámetros como: “name”, “hp”, “attacke”, “defense”

Por ultimo, se instancia los objetos y con sus respectivos parámetros pero con un extra incluido que seria utilizando la librería random en cada parámetro excepto el nombre para lograr un enfrentamiento mucho mas dinámico.

Aquí se adjuntan algunas lineas de código importantes del código.

  def attacke(self, other\_character):

        rdattack = random.randint(self.attack-40,self.attack+40)

        rddefense = random.randint(other\_character.defense-30,other\_character.defense+30)

        damage = rdattack - rddefense

        other\_character.hp -= damage

        print(f"{self.name} lanza un ataque de {rdattack}")

        time.sleep(2)

        print(f"{other\_character.name} se defendio con {rddefense}")

        time.sleep(2)

        print(f"{self.name} le hizo {damage} de daño a {other\_character.name}!")

        time.sleep(3)

    def combate(self, player2):

        i=0

        vida1=self.hp

        vida2=player2.hp

        while self.hp > 0 and player2.hp > 0:

            i=i+1

            print("\n--------------------------------------------------------------------------------------------")

            print("\nMinuto", i)

            time.sleep(1)

            random\_number = random.randint(0, 3)

            print(f"\n{self.name} Intenta atacar a {player2.name}")

            time.sleep(3)

            if random\_number == 0:

                print(f"{self.name} fallo")

                time.sleep(3)

            else:

                self.attacke(player2)

            if player2.hp > 0:

                print(f"\n{player2.name} Coontraataca!\n")

                time.sleep(3)

                random\_number2 = random.randint(0, 3)

                if random\_number2 == 0:

                    print(f"{player2.name} fallo")

                    time.sleep(3)

                else:

                    player2.attacke(self)

                if self.hp > 0 and player2.hp > 0:

                    print("\nResultados:")

                    time.sleep(1)

                    self.vida(vida1)

                    time.sleep(1)

                    player2.vida(vida2)

                    time.sleep(1)

                    input("\nPresione enter para pasar al siguiente round")

        if self.hp > 0:

            print(f"\n{self.name} GANOO!")

        else:

            print(f"\n{player2.name} GANOOO lol!")

Aquí se adjuntan partes del código en el cual fueron indispensables las sugerencias de la IA.

    def attacke(self, other\_character):

        rdattack = random.randint(self.attack-40,self.attack+40)

        rddefense = random.randint(other\_character.defense-30,other\_character.defense+30)

        damage = rdattack - rddefense

        other\_character.hp -= damage

        print(f"{self.name} lanza un ataque de {rdattack}")

        time.sleep(2)

        print(f"{other\_character.name} se defendio con {rddefense}")

        time.sleep(2)

        print(f"{self.name} le hizo {damage} de daño a {other\_character.name}!")

        time.sleep(3)

En esta parte del código la ayuda del Chat GPT fue MUY útil para hacer el ataque random, como fue una de las varias formas que sugería la IA resulto ser muy útil al momento de ejecutar el código y utilizar este algoritmo para los enfrentamiento

## LENGUAJE JAVASCRIPT

### Características:

* Lenguaje orientado a objetos
* Tipado debil o no tipado
* Alto nivel
* Lenguaje interpretado

### Librerías utilizadas:

* Math js

El Segundo lenguaje a utlizar y traspasar todo lo que hicimos en python es Javascript, empezamos a hacer el enfrentamiento jugador 1 vs jugador 2 y partimos utilizando la primera clase llamada ”character” y le pasamos sus respectivos parámetros como: “name”,”hp”,”attack”,”defense”, se hizo un función llamada doDamage donde está el daño a hacer del personaje y utilizando el math.random para hacerlo más de suerte.ocupamos la herencia para así crear distinto personajes como: Warrior,Mage, God, Spartan y que tengan los mismos atributos, se hizo una nueva clase que seria como el motor del juego llamada “engine” y se le añadieron metodos como “combat” , “playerislive”,”showroundstatus”, “showstatus”, y otros metodos ocupados como comentarios donde se ocupara el <div donde en el archivo HTML se mostrar unos diseños al usuario con efectos visuales

Al final instaciamos los objetos con sus atributos, en el “hp”, “attack”, “defense” se le agrego el math.random y el math.round para que se lograra un enfrentamiento mas dinamico

Se adjuntan algunas lineas de código importantes del código

![Texto

Descripción generada automáticamente]()

![Texto

Descripción generada automáticamente]()

![Texto

Descripción generada automáticamente]()

# Conclusión

Para concluir, obtenemos un gran conocimiento de la programación orientada a objetos tanto en python como en javascript, si bien ambos lenguajes pueden ser un poco diferentes, cabe destacar que los pilares de programacion que se utilizan no varían en su complejidad o entendimiento, si no que se complementan ya sea en python o en javascript o en cualquier otro lenguaje orientado a objetos, por lo tanto el conocimiento adquirido en este proyecto es muy útil para otros lenguajes u otros proyectos ya sean laborales o académicos, también destacar que el uso de las IA en proyectos puede ser muy eficiente al momento de salir de dudas ya que provee de información concisa para solucionar el problema que estemos tratando, si bien la lógica en la cual todos nos basamos para solucionar distintos problemas puede variar de persona en persona, la IA nos da una base para poder empezar desde ahi a planificar el camino hacia la solución.