

PROYECTO FINAL PYTHON

CÓDIGO: Te he quitado la parte de los dibujos para que no ocupe tanto espacio.

```
import json
import random
import time

# Archivo para guardar el progreso
Guardado = "guardados_juego.json"

# Clases de personajes
Personajes = {
    "Sonic": {"vida": 100, "ataque": 15, "defensa": 10, "inventario":
["Espada", ]},
    "Mario": {"vida": 100, "ataque": 20, "defensa": 5, "inventario":
["Armadura", "Puño americano"]},
    "Kratos": {"vida": 90, "ataque": 18, "defensa": 7, "inventario":
["Espada"]},
    "Pac Man": {"vida": 120, "ataque": 20, "defensa": 12, "inventario":
[ "Armadura"]},
    "Crash Bandicoot": {"vida": 110, "ataque": 18, "defensa": 8,
"inventario": ["Espada", "Puño americano"]}
}

# Lista de ubicaciones
Localizacion = ["isla de las muñecas", "Capilla de los huesos", "Casa
de las caras", "Hacienda Ibarburu", "Pantano Oscuro", "Catacumbas de
Paris", "Cementerio Dos Hermanas"]

# Enemigos posibles
enemigos = [
    {"nombre": "Creeper", "vida": 60, "ataque": 10},
    {"nombre": "Kuro ", "vida": 80, "ataque": 12},
    {"nombre": "Dr. Wily", "vida": 100, "ataque": 8},
    {"nombre": "Dr. Neo Cortex ", "vida": 70, "ataque": 11},
    {"nombre": "Ivo Robotnik", "vida": 100, "ataque": 15},
    {"nombre": "Browser", "vida": 120, "ataque": 18},
    {"nombre": "Ganondorf ", "vida": 90, "ataque": 15}
]
```

```

#Funcion para guardar partida
def guardar_partida(jugador):
    try:
        # Abrimos el archivo guardado en modo lectura
        with open(Guardado, "w") as file:
            # Convertimos el diccionario del jugador en formato JSON y
            lo escribimos en el archivo
            json.dump(jugador, file)
            # Mostramos mensaje de éxito al guardar el progreso
            print("\nProgreso guardado exitosamente!\n")
            # Capturamos cualquier error durante el proceso
    except Exception as e:
        # Mostramos el error al guardar el progreso
        print(f"Error al guardar: {e}")

# Funcion para cargar la partida
def cargar_partida():

    try:
        # Abrimos el archivo guardado en modo lectura
        with open(Guardado, "r") as file:
            # Cargamos el JSON del archivo en un diccionario en Python
            jugador = json.load(file)
            # Mostramos mensaje de éxito al cargar el progreso
            print("\nProgreso cargado!\n")
            # Retornamos el diccionario del jugador cargado
            return jugador

    # Si el archivo no existe capturamos el error
    except FileNotFoundError:
        # Mostramos el mensaje de que no se encontró una partida
        guardada
        print("No se encontró una partida guardada.")
        # Si hay un error al interpretar el json
    except json.JSONDecodeError:
        # Mostramos el mensaje de que no se pudo interpretar el JSON
        del archivo guardado
        print("Error al cargar la partida guardada.")
        # Retornamos None si no se pudo cargar el progreso
        return None

#Funcion para crear el personaje
def crear_personaje():
    #Pedimos el nombre del personaje
    nombre = input("Introduce el nombre de tu personaje: ")

```

```

print("\nElige una personaje:")
#Mostramos la lista de clases de personajes disponibles
for i, cls in enumerate(Personajes.keys(), 1):
    print(f"{i}. {cls}")

while True:
    try:
        #Pedimos la selección del personaje
        choice = int(input("Selecciona el número del personaje: "))
        if 1 <= choice <= len(Personajes):
            break
        else:
            print("Número fuera de rango. Intenta nuevamente.")
    except ValueError:
        print("Entrada inválida. Ingresa un número válido.")

#Obtenemos la clase seleccionada
selected_class = list(Personajes.keys())[choice - 1]
#Creamos el personaje
jugador = {"nombre": nombre, "clase": selected_class,
**Personajes[selected_class], "inventario": []}
print("\nTu personaje es un", selected_class , "llamado ", nombre)
print(Personajes_ASCII[selected_class])
return jugador

#Funcion para huir del combate
def huir(jugador):
    #Se recibe un daño aleatorio por huir
    daño = random.randint(5, 15)
    jugador["vida"] -= daño
    print(f"Has huido del combate y perdiste {daño} puntos de vida por
cagon. La vida no es facil pero hay que luchar siempre. Vida actual:
{jugador['vida']}")

# Funcion para usar la pocion de escudo
def usar_pocion(jugador):
    #Verificamos si el jugador tiene la pocion de escudo en el
inventario
    if "Poción de escudo" in jugador["inventario"]:
        # Recuperamos vida
        jugador["vida"] += 10
        # Eliminamos la pocion de escudo del inventario
        jugador["inventario"].remove("Poción de escudo")

```

```

        print(f"Has usado una Poción de escudo y recuperado 10 puntos
de vida. Tu vida: {jugador['vida']}")
    else:
        print("No tienes una Poción de escudo disponible.")

# Funcion para explorar
def exploracion(jugador):
    print("\nExplorando...")
    # Tiempo de espera para explorar
    time.sleep(2)
    # Se genera una ubicación aleatoria para el enemigo
    location = random.choice(Localizacion)
    print(f"Has llegado a {location}.")

    if random.random() < 0.5:
        #Posibilidad de encontrar un enemigo
        enemigo = random.choice(enemigos)
        print(f";Un {enemigo['nombre']} aparece en {location}!")
        combate(jugador, enemigo)
    else:
        # Posibilidad de encontrar un item
        item = random.choice(["Poción de escudo", "Espada", "Armadura",
"Puño americano"])
        jugador["inventario"].append(item)
        print(f"Has encontrado un {item}")

        if item == "Poción de escudo":
            decision = input(";Quieres guardarla en el inventario (G) o
usarla ahora (U)? ").lower()
            if decision == "u":
                jugador["vida"] += 10
                print(f"Has usado una Poción de escudo y recuperado 10
puntos de vida. Tu vida: {jugador['vida']}")
            else :
                jugador["inventario"].append(item)
                print("Has guardado la Poción de escudo en tu
inventario.")

        elif item == "Armadura":
            #Aumentamos defensa
            jugador["defensa"] += 5

```

```

        print(f"Has aumentado tu defensa en 5 puntos.")
    elif item in ["Puño americano", "Espada"]:
        #Aumentamos ataque
        jugador["ataque"] += 10
        print(f"Has aumentado tu ataque en 10 puntos.")

# Funcion para el combate
def combate(jugador, enemigo):
    print(f"\n;Combate contra {enemigo['nombre']}!")
    #Mientras ambos tengan vida el combate sigue
    while jugador["vida"] > 0 and enemigo["vida"] > 0:
        accion = input("¿Atacar (A), Huir (H) o usar poción (P)??")
        .lower()

        # Si el jugador elige atacar
        if accion == "a":
            # Reducimos la vida al enemigo segun el ataque del jugador
            enemigo["vida"] -= jugador["ataque"]
            print(f"Golpeas al {enemigo['nombre']} por {jugador['ataque']} de daño. Vida enemiga: {enemigo['vida']}")
            #Si el enemigo es derrotado
            if enemigo["vida"] <= 0:
                print(f";Has derrotado al {enemigo['nombre']}!")
                #Llamamos a la funcion despues del combate
                despues_del_combate(jugador)
                break

            #El enemigo ataca al jugador
            jugador["vida"] -= enemigo["ataque"]
            print(f"El {enemigo['nombre']} te golpea por {enemigo['ataque']} de daño. Tu vida: {jugador['vida']}")
            # Si el jugador elige huir
        elif accion == "h":
            huir(jugador)
            break

        # Si el jugador elige usar poción
        elif accion == "p":
            usar_pocion(jugador)
        else:
            print("Comando inválido.")

    #Si el jugador pierde toda su vida se termina el juego
    if jugador["vida"] <= 0:
        print("Has sido derrotado. Fin del juego.")
        exit()

```

```

# Funcion para despues del combate
def despues_del_combate(jugador):
    while True:
        print("\n¿Qué quieres hacer ahora?")
        print("1. Seguir explorando")
        print("2. Usar poción si tienes")
        choice = input("Selecciona una opción: ")
        if choice == "1":
            break
        elif choice == "2":
            usar_pocion(jugador)
        else:
            print("Opción inválida.")

# Funcion principal del juego
def main():
    print("Bienvenido al juego de aventura!")
    #Inicio del juego
    while True:
        print("\n1. Nueva Partida")
        print("2. Cargar Partida")
        print("3. Salir")
        choice = input("Selecciona una opción: ")
        if choice == "1":
            jugador = crear_personaje()
            break
        elif choice == "2":
            jugador = cargar_partida()
            if jugador:
                break
        elif choice == "3":
            print("Saliendo del juego...")
            return
        else:
            print("Opción inválida.")

    while True:
        #Menu de opciones
        print("\nOpciones:")
        print("1. Explorar")
        print("2. Usar poción")
        print("3. Guardar Partida")
        print("4. Salir")

```

```

choice = input("Selecciona una opción: ")
if choice == "1":
    exploracion(jugador)
elif choice == "2":
    usar_pocion(jugador)
elif choice == "3":
    guardar_partida(jugador)
elif choice == "4":
    print("Saliendo del juego...")
    break
else:
    print("Opción inválida.")

if __name__ == "__main__":
    main()

```

Prueba caja negra:

Para esto vamos a ejecutar el código y verificar que funciona correctamente las entradas y las salidas que nos da una vez escogida las diferentes opciones.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS
2. Cargar Partida
3. Salir
Selecciona una opción: 2

Progreso cargado!

Opciones:
1. Explorar
2. Usar poción
3. Guardar Partida
4. Salir
Selecciona una opción: 1

Explorando...
Has llegado a Cementerio Dos Hermanas.
Has encontrado un Espada
Has aumentado tu ataque en 10 puntos.

Opciones:
1. Explorar
2. Usar poción
3. Guardar Partida
4. Salir
Selecciona una opción: 1

Explorando...
Has llegado a Hacienda Ibarburu.
Has encontrado un Poción de escudo
¿Quieres guardarla en el inventario (G) o usarla ahora (U)? u
Has usado una Poción de escudo y recuperado 10 puntos de vida. Tu vida: 47

Opciones:
1. Explorar
2. Usar poción
3. Guardar Partida
4. Salir
Selecciona una opción: 1

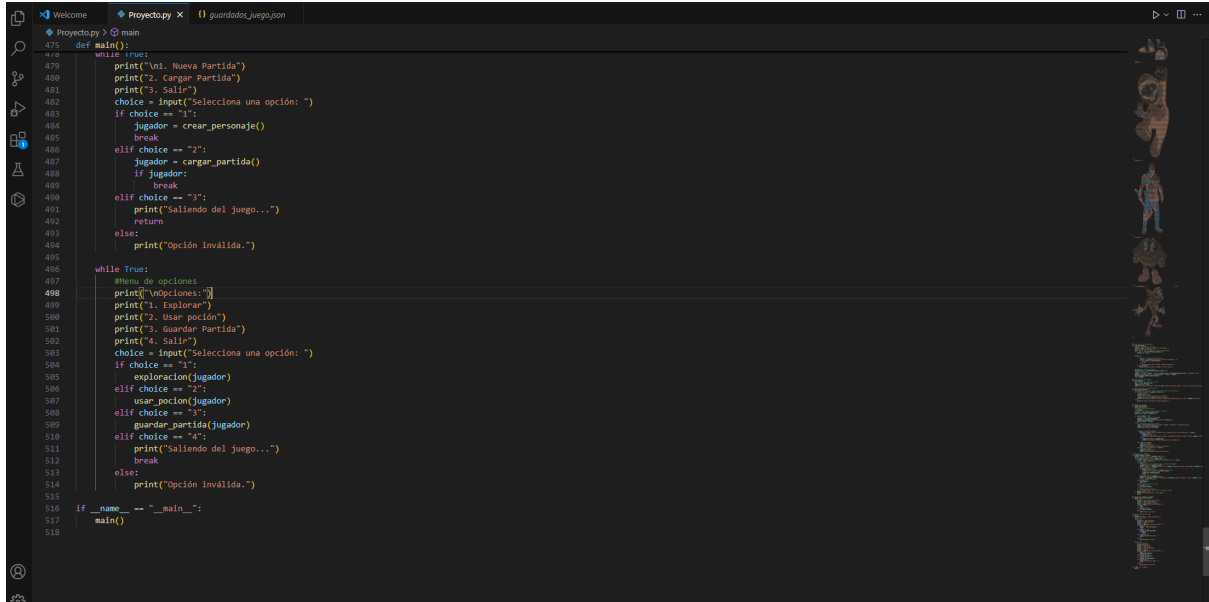
Explorando...
Has llegado a Casa de las caras.
Has encontrado un Poción de escudo
¿Quieres guardarla en el inventario (G) o usarla ahora (U)? g
Has guardado la Poción de escudo en tu inventario.

Opciones:
1. Explorar
2. Usar poción

```

Prueba caja blanca:

Como podemos ver en la siguiente captura el código no nos da ningún error, ejecutando todas las sentencias y estructuras del código.



```
475 def main():
476     while True:
477         print("\n1. Nueva Partida")
478         print("2. Cargar Partida")
479         print("3. Salir")
480         choice = input("Selecciona una opción: ")
481         if choice == "1":
482             jugador = crear_personaje()
483             break
484         elif choice == "2":
485             jugador = cargar_partida()
486             if jugador:
487                 break
488         elif choice == "3":
489             print("Saliendo del juego...")
490             return
491         else:
492             print("Opción inválida.")
493
494     while True:
495         #Menu de opciones
496         print("\nOpciones:")
497         print("1. Explorar")
498         print("2. Usar poción")
499         print("3. Guardar Partida")
500         print("4. Salir")
501         choice = input("Selecciona una opción: ")
502         if choice == "1":
503             exploracion(jugador)
504         elif choice == "2":
505             usar_pocion(jugador)
506         elif choice == "3":
507             guardar_partida(jugador)
508         elif choice == "4":
509             print("Saliendo del juego...")
510             break
511         else:
512             print("Opción inválida.")
513
514 if __name__ == "__main__":
515     main()
516
517
518
```