

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Curso Laboratorio de

Introducción a la

Programación y

Computación 1

Sección: F

Ing.

Tutor Académico: Zenaida Irazema Chacón García



Proyecto 1

Nombre juego: Juego por turnos

Nombre: Gustavo Alejandro Chip Aguilar

Carné: 202308450

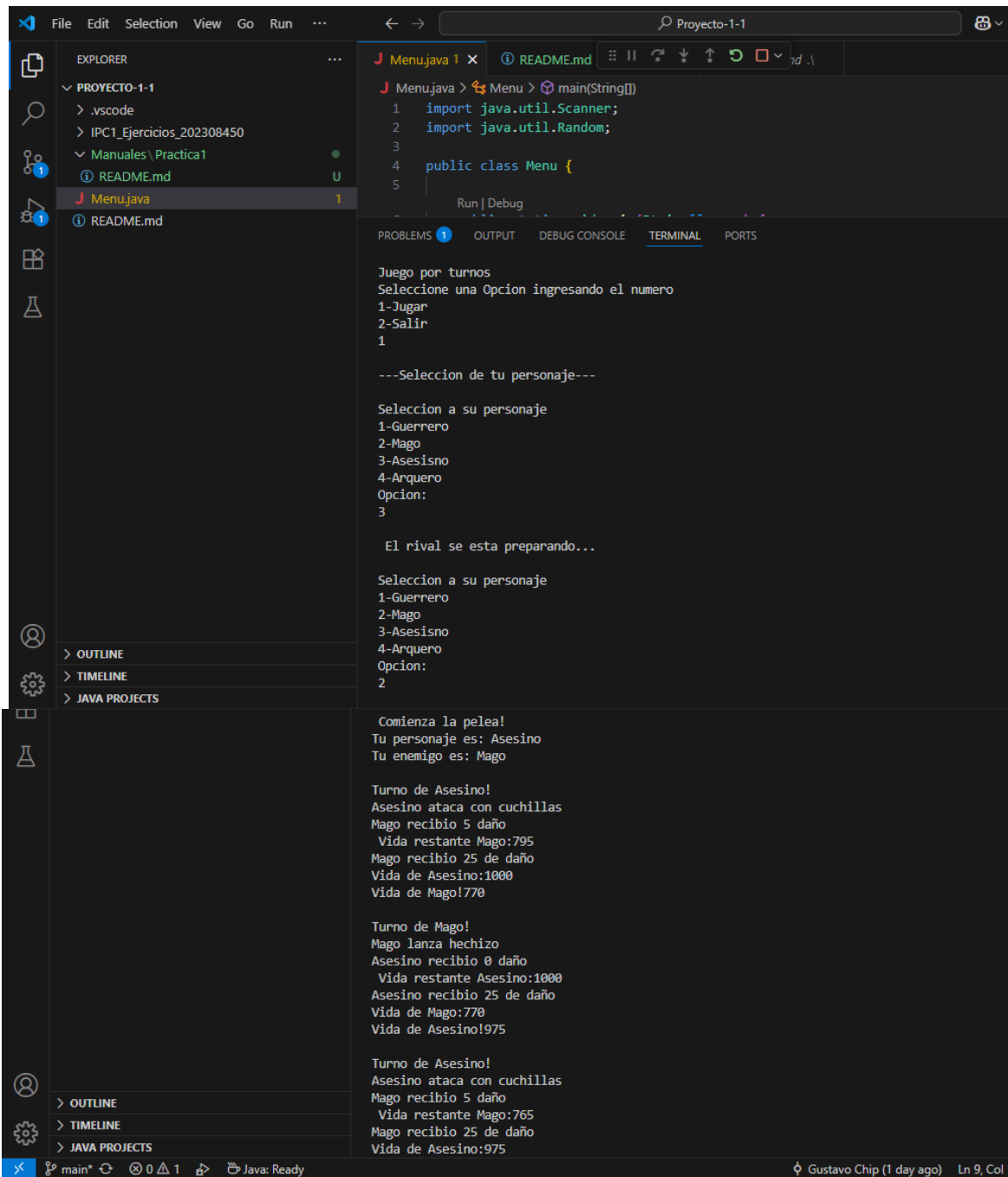
Fecha: agosto 2025

Desarrollo de

Introducción:

Es un juego por turnos el cual podemos seleccionar diferentes personajes y armas para combatir contra un rival en consola del editor de código utilizando el código de programación java.

Plataforma de Ejecución:



El código de java es ejecutado en la consola del editor de código Visual Studio Code

Líneas de Códigos mas Importante:

```
Menu.java > personaje > personaje(String, int, int, int)
4 public class Menu {
42 // Logica principal del juego
43 public static void jugar(Scanner scanner){
44     System.out.println(x:"\n---Seleccion de tu personaje---");
45     personaje jugador = elegirpersonaje(scanner);
46
47     System.out.println(x:"\n El rival se esta preparando...");
48     personaje rival = elegirpersonaje(scanner);
49
50     System.out.println(x:"\n Comienza la pelea!");
51     System.out.println("Tu personaje es: " + jugador.getclase());
52     System.out.println("Tu enemigo es: " + rival.getclase());
53
54 // clase = jugador, se modifiko porque empezo a hacer confuso para identificar al jugador de la clase
55 // de personaje seleccionado.
56
57     while (jugador.vivo() && rival.vivo()) {
58         turno(jugador, rival);
59         // El bucle comienza y finaliza hasta que el jugador derroto al rival
60         if (!rival.vivo()){
61             break;
62         }
63         turno(rival, jugador);
64     }
65
66     System.out.println(x:"\n Fin de la partida");
67     if (jugador.vivo()) {
68         System.out.println("Has ganado :) " + jugador.getclase() );
69     } else {
70         System.out.println("Has sido Derrotado :/ " + rival.getclase());
71     }
72 }
```

Opción de menú de selección de personajes, bucle de si los personajes están vivo (tiene vida) que finaliza hasta que uno de los dos tenga una vida menor a 0. Mensaje de victoria o derrota de la partida.

```
125 }
126 }
127
128 // Como funciona los turnos
129 private static void turno(personaje atacante, personaje defensor){
130     System.out.println("\nTurno de " + atacante.getclase() + "!");
131     atacante.atacar(defensor);
132     if (defensor.getdot() > 0) {
133         defensor.aplicardot();
134     }
135     System.out.println("Vida de " + atacante.getclase() + ":" + atacante.gethp());
136     System.out.println("Vida de " + defensor.getclase() + ":" + defensor.gethp());
137 }
```

Líneas de código que establecen el orden de acción del usuario y el rival con mensajes de vida restante de cada después de cada ataque recibido

Diccionario de Métodos:

Método: `elegirpersonaje(scanner)` = Lee el personaje seleccionado (la opcion)

Método: `turno(personaje atacante, personaje defensor)` = Lee los dos personajes que va a pelear y los separa como el atacante y el defensor

Método: `personaje(String clase, int hp, int atk, int def, int dot)` = estamos diciendo que es lo que contiene el personaje, el orden y que valores son si son caracteres o números enteros.

Método: `recibirdano(int dano)` = toma el dano (daño) como un valor numérico

Método: `Guerrero(String clase)` = guerrero es el nombre de personaje o en este caso la clase que selecciona el jugador, String lee lo que este en “clase” en caracteres que se le mostrar la información de la acción al jugador en la consola.

Método: `Mago(String clase)` = mago es el nombre de personaje o en este caso la clase que selecciona el jugador, String lee lo que este en “clase” en caracteres que se le mostrar la información de la acción al jugador en la consola.

Método: `Asesino(String clase)` = asesino es el nombre de personaje o en este caso la clase que selecciona el jugador, String lee lo que este en “clase” en caracteres que se le mostrar la información de la acción al jugador en la consola.

Método: `Arquero(String clase)` = arquero es el nombre de personaje o en este caso la clase que selecciona el jugador, String lee lo que este en “clase” en caracteres que se le mostrar la información de la acción al jugador en la consola.

Método: `atacar(personaje rival)` = se le dice que al atacar va a ser daño al enemigo .

Manual de Usuario

- Al iniciar el Código el usuario tendrá la opción de jugar o salir
- Al colocar el numero de la opción de jugar aparecerá un mensaje de selección de personajes disponibles
- También tendrá que seleccionar a su rival
- El juego empieza.

Implementación de la practica

Para mayor comodidad se utilizo el editor de Código de Visual Studio Code, antes de empezar con la programación del juego se busco material de estudio que en este caso fue videos de introducción a la programación utilizando java para conocer el código luego se pensó en que es lo que se quería realizar con el juego que acciones tendría, que personajes, como funcionaria, si tendrían habilidades alguna clasificación, buffs durante la partida que le darían tomar ventajas sobre le enemigo o le jugaría en su contra, diferentes armas, un historial y duración de juego, clasificación de enemigos.

Teniendo la idea de que es lo que se quería lograr se empezó con la programación del juego.

Problemas encontrados durante el desarrollo

- Problemas con la instalación de java
- Problemas con el primer editor de código a utilizar que es IntelliJ IDEA ya que no reconocía la versión de java instalado por lo que se intentó de diferentes formas y no era capaz de reconocer el código.
- Errores al momento de programar con los bucles.
- Errores de problemas de reconocimiento de variables.
- Errores con las { } que se mezclaba o perdía.
- No tener un orden con el codigo.

Soluciones Adoptadas

- Se cambio de editor de código por Visual Studio Code.
- Corrección de errores con bucles y métodos que no eran reconocidos.
- Mejora con la sintaxis de código un mejor orden al momento de programar.
- Implementación de comentarios para saber donde comienza un nuevo procedimiento.

Conclusión

A pesar de no poder cumplir con todos los objetivos propuestos el juego es funcional con grandes aspectos de mejora que se pueden realizar en un futuro, dándole actualizaciones al juego como una mayor cantidad de personajes un mejor equilibrio en estadísticas de cada personaje, haciendo que el jugador disfrute de una buena experiencia y un rato de diversión.

