

Informe de Proyecto Final

Tema: Soul Knight Fan Remake

Nota

Estudiante	Escuela	Asignatura
Carlos Orson Peralta Marreros Bran James Paz Patatingo Heydy Anyely Castillo Arocutipa Angelica Giovana Huaman Cuba cperaltam@ulasalle.edu.pe bpazp@ulasalle.edu.pe hcastilloa@ulasalle.edu.pe ahuamanc@ulasalle.edu.pe	Carrera Profesional de Ingeniería de Software	Lenguaje de Programación II Semestre: III Código: carlos : 78288004 angelica : 60864535

Laboratorio	Tema	Duración
Proyecto Final	Soul Knight Fan Remake	24 días

Semestre académico	Fecha de inicio	Fecha de entrega
2025 - B	Del 1 noviembre 2025	Al 24 de noviembre 2025

1. Tarea

- Desarrollar un videojuego 2D basado en Soul Knight, exclusivamente para computadoras o laptops, con skins divertidas, un mapa sencillo e intuitivo. Agregar el modo multiplayer local. Se creará el videojuego en Unity usando tecnologías vistas en clase como threads para el modo multiplayer o para los enemigos que el personaje debe enfrentar.
- Como base conceptual inicial, se implementó una versión de consola del juego en Java llamada "Mini Dungeon Console", que permite entender la mecánica básica del juego antes de desarrollar la versión completa en Unity.

2. Equipos, materiales y temas utilizados

2.1. Herramientas de Desarrollo

- Control de versiones:
 - Git

- GitHub
- Unity Hub + Motor Unity

■ **IDE y Editores:**

- Visual Studio Code
- Eclipse (para prototipo en Java)
- Apache NetBeans (para prototipo en Java)

■ **Diseño de personajes:**

- Photoshop
- Canva
- GIMP
- Krita
- DragonBones
- Aseprite

■ **Recursos Unity:**

- 2D Animation Package
- Cinemachine
- DOTween
- Unity Asset Store

■ **Sonido:**

- Murf IA
- Audio Studio
- Pixabay
- EpidemicSound

2.2. Conceptos Técnicos

- Uso de hilos (threads)
- Programación Orientada a Objetos
- Control de versiones con Git
- Cuenta institucional GitHub

3. URL de Repositorio GitHub

- URL del Repositorio GitHub para clonar o recuperar:
- `https://github.com/rescobedoulasalle/pd.git`
- URL para el proyecto final en el Repositorio GitHub:
- `https://github.com/rescobedoulasalle/pd/tree/main/Juego%20Base%20para%20la%20consola`

4. Descripción del Proyecto

4.1. Concepto General

Soul Knight Fan Remake es un videojuego 2D basado en el popular Soul Knight original. A diferencia del juego original diseñado para móviles, esta versión está optimizada exclusivamente para computadoras y laptops, con controles intuitivos de teclado y mouse.

4.2. Características Principales

- **Skins divertidas:** Variedad de personajes con diseños creativos
- **Mapa sencillo e intuitivo:** Diseño de niveles fácil de navegar
- **Modo multijugador local:** Cooperación entre dos jugadores en la misma máquina
- **Sistema de habitaciones:** Cada nivel consta de 5 habitaciones conectadas
- **5 niveles totales:** Progresión con dificultad incremental
- **Jefe final:** Boss con alta cantidad de vida en el nivel 5

4.3. Entorno del Juego

La historia del juego es directa y enfocada en la acción:

1. El personaje jugable aparece en una habitación inicial equipado con un arma
2. Las habitaciones están conectadas por caminos
3. Al entrar a una nueva habitación, las puertas se cierran automáticamente
4. Aparecen enemigos que el jugador debe derrotar
5. Una vez eliminados todos los enemigos, las puertas se abren
6. Cada nivel tiene 5 habitaciones
7. El juego cuenta con 5 niveles en total
8. En el nivel 5 (última habitación) aparece un jefe final con mucha vida

4.4. Elementos del Juego

4.4.1. Items principales

- Personajes jugables
- Enemigos variados
- Jefe final (boss)
- Mapas/Habitaciones
- Objetos consumibles (curas o bebidas)
- Variedad de armas

5. Prototipo en Java - Mini Dungeon Console

5.1. Propósito del Prototipo

Con el fin de tener una base inicial con conceptos claros y detallados del videojuego, se creó un prototipo simple para ejecutar desde la terminal (consola). Este prototipo en Java permite entender y probar las mecánicas básicas del juego:

- Sistema de movimiento del jugador
- Sistema de movimiento aleatorio de enemigos
- Sistema de combate básico
- Detección de colisiones
- Sistema de vida para jugador y enemigo
- Visualización del mapa en consola

5.2. Código Fuente Completo

5.2.1. Archivo: Jugador.java

Listing 1: Clase Jugador

```
// Clase base para el personaje jugador
class Jugador {
    int vida = 20;
    int dano = 5;
    int x = 0, y = 0;
}
```

5.2.2. Archivo: Enemigo.java

Listing 2: Clase Enemigo

```
// Clase para el enemigo
class Enemigo {
    int vida = 10;
    int dano = 3;
    int x, y;

    public Enemigo(int x, int y) {
        this.x = x;
        this.y = y;
    }
}
```

5.2.3. Archivo: MiniDungeon.java (Versión Completa)

Listing 3: Clase Principal del Juego - MiniDungeon.java

```
import java.util.Random;
```

```
import java.util.Scanner;

// Clase base para el personaje jugador
class Jugador {
    int vida = 20;
    int dano = 5;
    int x = 0, y = 0;
}

// Clase para el enemigo
class Enemigo {
    int vida = 10;
    int dano = 3;
    int x, y;

    public Enemigo(int x, int y) {
        this.x = x;
        this.y = y;
    }
}

// Clase principal del juego
public class MiniDungeon {
    static final int TAM = 5;
    static Jugador jugador = new Jugador();
    static Enemigo enemigo = new Enemigo(3, 3);
    static Scanner sc = new Scanner(System.in);

    public static void main(String[] args) {
        System.out.println("=== MINI DUNGEON CONSOLE ===");
        System.out.println("Controles: w=arriba, s=abajo, a=izquierda, d=derecha, x=atacar\n");

        // Bucle principal del juego
        while (jugador.vida > 0 && enemigo.vida > 0) {
            mostrarMapa();
            System.out.print("Mover > ");
            char mov = sc.next().charAt(0);
            moverJugador(mov);
            moverEnemigo();

            if (enemigo.vida <= 0) break;
        }

        // Resultado final
        if (jugador.vida <= 0)
            System.out.println("\n Has sido derrotado...");
        else
            System.out.println("\n Ganaste la batalla!");

        sc.close();
    }

    // Muestra el mapa en la consola
    static void mostrarMapa() {
        System.out.println("\n--- MAPA ---");
    }
}
```

```
for (int i = 0; i < TAM; i++) {
    for (int j = 0; j < TAM; j++) {
        if (jugador.x == i && jugador.y == j)
            System.out.print("P "); // Jugador
        else if (enemigo.x == i && enemigo.y == j)
            System.out.print("E "); // Enemigo
        else
            System.out.print(". "); // Espacio vacio
    }
    System.out.println();
}
System.out.println("    Vida Jugador: " + jugador.vida +
    " |    Vida Enemigo: " + enemigo.vida);
}

// Controla el movimiento y ataque del jugador
// By Carlos Peralta
static void moverJugador(char mov) {
    switch (mov) {
        case 'w':
            if (jugador.x > 0) jugador.x--;
            break;
        case 's':
            if (jugador.x < TAM - 1) jugador.x++;
            break;
        case 'a':
            if (jugador.y > 0) jugador.y--;
            break;
        case 'd':
            if (jugador.y < TAM - 1) jugador.y++;
            break;
        case 'x':
            // Sistema de ataque
            if (jugador.x == enemigo.x && jugador.y == enemigo.y) {
                enemigo.vida -= jugador.dano;
                System.out.println("    Buen ataque (-" + jugador.dano + " HP)");
            } else {
                System.out.println("    No hay enemigo aqui");
            }
            return;
    }
}

// Controla el movimiento aleatorio del enemigo
static void moverEnemigo() {
    Random r = new Random();
    int mov = r.nextInt(4);

    switch (mov) {
        case 0: if (enemigo.x > 0) enemigo.x--; break;
        case 1: if (enemigo.x < TAM - 1) enemigo.x++; break;
        case 2: if (enemigo.y > 0) enemigo.y--; break;
        case 3: if (enemigo.y < TAM - 1) enemigo.y++; break;
    }

    // Si el enemigo alcanza al jugador, lo ataca
}
```

```
        if (enemigo.x == jugador.x && enemigo.y == jugador.y) {  
            jugador.vida -= enemigo.dano;  
            System.out.println("        Te atacan! (-" + enemigo.dano + " HP)");  
        }  
    }  
}
```

5.3. Explicación del Código

5.3.1. Clase Jugador

Define las propiedades básicas del personaje jugable:

- **vida**: Puntos de vida del jugador (inicia con 20)
- **dano**: Daño que inflige al atacar (5 puntos)
- **x, y**: Posición en el mapa (inicia en 0,0 - esquina superior izquierda)

5.3.2. Clase Enemigo

Define las propiedades del enemigo:

- **vida**: Puntos de vida del enemigo (inicia con 10)
- **dano**: Daño que inflige al jugador (3 puntos)
- **x, y**: Posición inicial (3,3 - cerca del centro del mapa)
- Constructor que permite definir la posición inicial

5.3.3. Método main()

Contiene el flujo principal del juego:

1. Muestra el título y controles
2. Inicia el bucle principal que se ejecuta mientras ambos estén vivos
3. En cada iteración:
 - Muestra el mapa actual
 - Lee el input del jugador
 - Mueve al jugador según el input
 - Mueve al enemigo aleatoriamente
4. Al finalizar, muestra el resultado (victoria o derrota)

5.3.4. Método mostrarMapa()

Dibuja el estado actual del juego en la consola:

- Recorre una matriz de 5x5
- Muestra "P" donde está el jugador
- Muestra "E" donde está el enemigo
- Muestra ".." espacios vacíos
- Muestra las vidas actuales de ambos

5.3.5. Método moverJugador()

Procesa el input del jugador:

- w: Mueve arriba (decrementa x)
- s: Mueve abajo (incrementa x)
- a: Mueve izquierda (decrementa y)
- d: Mueve derecha (incrementa y)
- x: Ataca si está en la misma posición que el enemigo
- Valida que no se salga de los límites del mapa (0 a TAM-1)

5.3.6. Método moverEnemigo()

Controla el comportamiento del enemigo:

- Genera un número aleatorio entre 0 y 3
- Mueve al enemigo en una dirección aleatoria
- Valida límites del mapa
- Si está en la misma posición que el jugador, lo ataca automáticamente

6. Compilación y Ejecución del Prototipo Java

6.1. Requisitos Previos

- Java Development Kit (JDK) 8 o superior instalado
- Variable de entorno JAVA_HOME configurada
- Terminal o línea de comandos

6.2. Verificar instalación de Java

Listing 4: Verificar versión de Java

```
$ java -version  
$ javac -version
```

La salida debe mostrar la versión instalada, por ejemplo:

```
java version "11.0.12" 2021-07-20 LTS  
Java(TM) SE Runtime Environment 18.9 (build 11.0.12+8-LTS-237)
```


6.3. Pasos para Compilar y Ejecutar

6.3.1. Paso 1: Clonar el repositorio

Listing 5: Clonar repositorio desde GitHub

```
$ cd $HOME/lasalle
$ git clone https://github.com/rescobedoulasalle/pd.git
$ cd pd/Juego\ Base\ para\ la\ consola
```

6.3.2. Paso 2: Verificar archivos

Listing 6: Listar archivos del proyecto

```
$ ls -la
```

Debe mostrar el archivo `MiniDungeon.java`

6.3.3. Paso 3: Compilar el código

Listing 7: Compilar MiniDungeon.java

```
$ javac MiniDungeon.java
```

Si no hay errores, se generarán los archivos `.class`:

- `MiniDungeon.class`
- `Jugador.class`
- `Enemigo.class`

6.3.4. Paso 4: Ejecutar el juego

Listing 8: Ejecutar el juego

```
$ java MiniDungeon
```

6.4. Ejemplo de Ejecución

Listing 9: Salida en consola del juego

```
=== MINI DUNGEON CONSOLE ===
Controles: w=arriba, s=abajo, a=izquierda, d=derecha, x=atacar

--- MAPA ---
P . . . .
. . . .
. . . .
. . . E .
. . . .
      Vida Jugador: 20 |      Vida Enemigo: 10
Mover > d
```

```
--- MAPA ---
. P . . .
. . . . .
. . . . .
. . . E .
. . . . .
      Vida Jugador: 20 |      Vida Enemigo: 10
Mover > s

--- MAPA ---
. . . . .
. P . . .
. . . . .
. . E . .
. . . . .
      Vida Jugador: 20 |      Vida Enemigo: 10
Mover > s

--- MAPA ---
. . . . .
. . . . .
. P . . .
. . . E .
. . . . .
      Te atacan! (-3 HP)
      Vida Jugador: 17 |      Vida Enemigo: 10
Mover > d

--- MAPA ---
. . . . .
. . . . .
. . P E .
. . . . .
. . . . .
      Vida Jugador: 17 |      Vida Enemigo: 10
Mover > x
      No hay enemigo aqui

--- MAPA ---
. . . . .
. . . . .
. . P E .
. . . . .
. . . . .
      Vida Jugador: 17 |      Vida Enemigo: 10
Mover > d

--- MAPA ---
. . . . .
. . . . .
. . . P .
. . . E .
. . . . .
      Te atacan! (-3 HP)
      Vida Jugador: 14 |      Vida Enemigo: 10
```

```
Mover > s

--- MAPA ---
. . . . .
. . . . .
. . . . .
. . . E P
. . . . .
    Vida Jugador: 14 |    Vida Enemigo: 10
Mover > a

--- MAPA ---
. . . . .
. . . . .
. . . . .
. . . P .
. . . E .
    Te atacan! (-3 HP)
    Vida Jugador: 11 |    Vida Enemigo: 10
Mover > s

--- MAPA ---
. . . . .
. . . . .
. . . . .
. . E . .
. . . P .
    Vida Jugador: 11 |    Vida Enemigo: 10
Mover > a

--- MAPA ---
. . . . .
. . . . .
. . . . .
. . . E .
. . P . .
    Vida Jugador: 11 |    Vida Enemigo: 10
Mover > w

--- MAPA ---
. . . . .
. . . . .
. . . . .
. . P E .
. . . . .
    Vida Jugador: 11 |    Vida Enemigo: 10
Mover > d

--- MAPA ---
. . . . .
. . . . .
. . . . .
. . . P .
. . E . .
    Vida Jugador: 11 |    Vida Enemigo: 10
Mover > s
```

```
--- MAPA ---
. . . . .
. . . . .
. . . . .
. . . . .
. . E P .
    Vida Jugador: 11 |      Vida Enemigo: 10
Mover > a

--- MAPA ---
. . . . .
. . . . .
. . . . .
. . E . .
. . P . .
    Te atacan! (-3 HP)
    Vida Jugador: 8 |      Vida Enemigo: 10
Mover > w

--- MAPA ---
. . . . .
. . . . .
. . . . .
. . P . .
. . E . .
    Vida Jugador: 8 |      Vida Enemigo: 10
Mover > w

--- MAPA ---
. . . . .
. . . . .
. . P . .
. . . . .
. E . . .
    Vida Jugador: 8 |      Vida Enemigo: 10
Mover > a

--- MAPA ---
. . . . .
. . . . .
. P . . .
. E . . .
. . . . .
    Vida Jugador: 8 |      Vida Enemigo: 10
Mover > s

--- MAPA ---
. . . . .
. . . . .
. . . . .
. P . . .
. E . . .
    Te atacan! (-3 HP)
    Vida Jugador: 5 |      Vida Enemigo: 10
Mover > s
```

```
--- MAPA ---
. . . . .
. . . . .
. . . . .
. . . . .
. P . . .
    Vida Jugador: 5 |      Vida Enemigo: 10
Mover > d

--- MAPA ---
. . . . .
. . . . .
. . . . .
. E . . .
. . P . .
    Vida Jugador: 5 |      Vida Enemigo: 10
Mover > a

--- MAPA ---
. . . . .
. . . . .
. . . . .
. E . . .
. P . . .
    Vida Jugador: 5 |      Vida Enemigo: 10
Mover > w

--- MAPA ---
. . . . .
. . . . .
. . . . .
. P . . .
E . . . .
    Te atacan! (-3 HP)
    Vida Jugador: 2 |      Vida Enemigo: 10
Mover > s

--- MAPA ---
. . . . .
. . . . .
. . . . .
. . . . .
P . . . .
    Vida Jugador: 2 |      Vida Enemigo: 10
Mover > d

--- MAPA ---
. . . . .
. . . . .
. . . . .
E . . . .
. P . . .
    Vida Jugador: 2 |      Vida Enemigo: 10
Mover > a
```

```
--- MAPA ---
. . . . .
. . . . .
. . . . .
. . . . .
P E . . .
    Vida Jugador: 2 |      Vida Enemigo: 10
Mover > x
    Buen ataque (-5 HP)

--- MAPA ---
. . . . .
. . . . .
. . . . .
. E . . .
P . . . .
    Vida Jugador: 2 |      Vida Enemigo: 5
Mover > w

--- MAPA ---
. . . . .
. . . . .
. . . . .
P . . . .
. E . . .
    Vida Jugador: 2 |      Vida Enemigo: 5
Mover > s

--- MAPA ---
. . . . .
. . . . .
. E . . .
P . . . .
    Vida Jugador: 2 |      Vida Enemigo: 5
Mover > s

--- MAPA ---
. . . . .
. . . . .
. . . . .
. E . . .
P . . . .
    Te atacan! (-3 HP)

    Has sido derrotado...
```