# DOCUMENTACIÓN TÉCNICA – PLAYER VS ENEMIGO

## 1. Descripción del juego

**Nombre del juego:** Player vs Enemigo  
**Objetivo:** El jugador debe moverse dentro del área de juego para evitar que los enemigos lo alcancen. Cada vez que un enemigo toca al jugador, este cambia de color.

## ANÁLISIS

**Historia de Usuario 01[HU01]:**

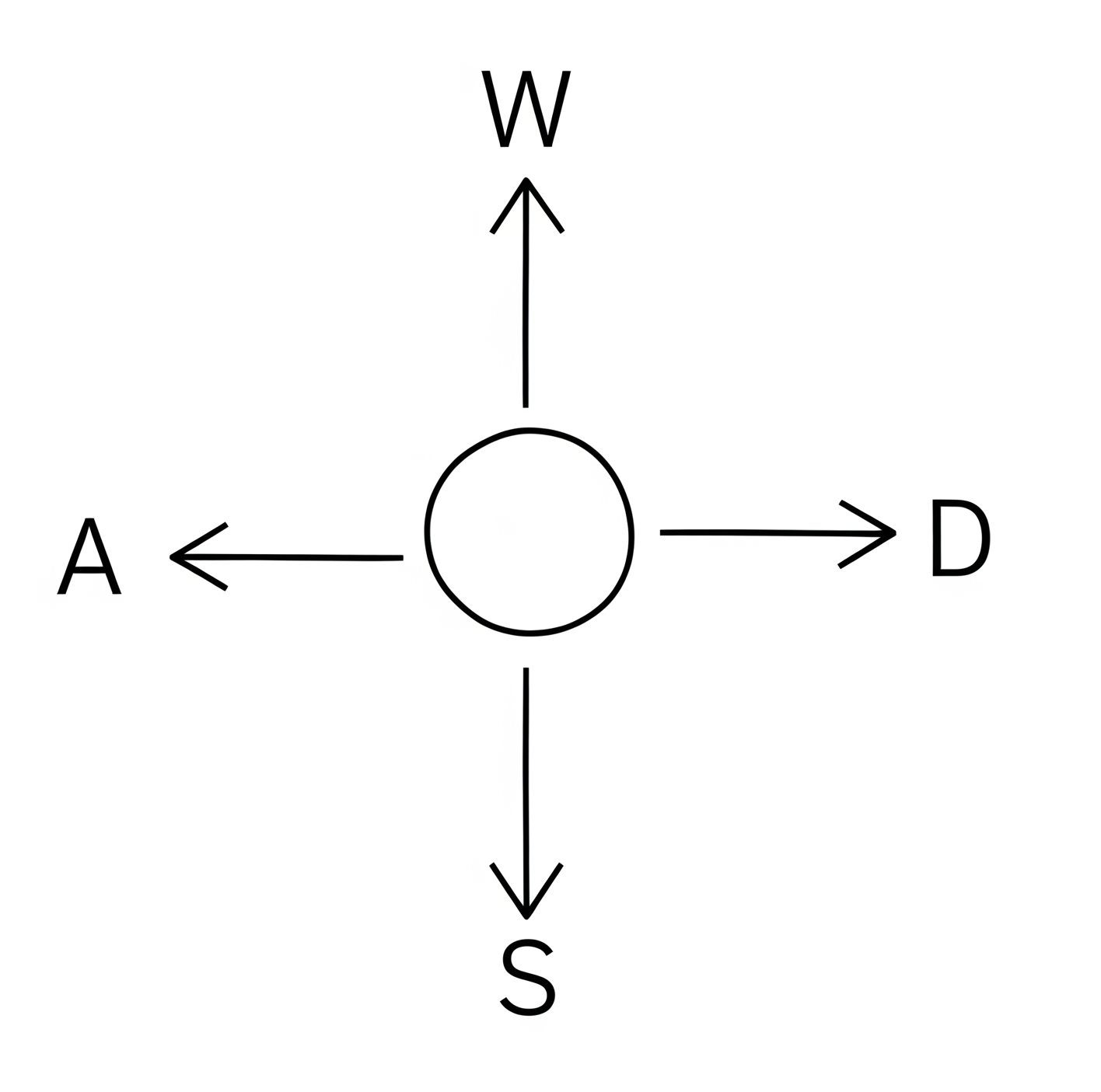
**Como** jugador **quiero** que el player se pueda mover en todas las direcciones **para** que pueda escapar del enemigo.

**Precondición:** Antes de moverse debe dibujarse el player.

**PostCondición:** Luego de moverse debe dibujarse el player.

| Historia | Entradas | Procesos | Salida |
| --- | --- | --- | --- |
| **HU01** | Teclas W, A, S, D | Detectar teclado y actualizar posición del jugador | Nueva posición del jugador |

## 



**Lista de Tareas H01:**

| **T01** | Crear clase Player con posición, diámetro, velocidad y color. |  |
| --- | --- | --- |
| **T02** | Implementar la lógica de movimiento del jugador según la tecla presionada. |  |
| **T03** | Visualizar al player con circle(). |  |

**Criterio de Aceptación HU01:**

| **T01** | Al presionar W/A/S/D, el jugador se mueve correctamente. |  |
| --- | --- | --- |
| **T02** | Al soltar cualquier tecla de movimiento, el jugador se detiene inmediatamente. |  |
| **T03** | Visualización correcta. |  |

DISEÑO

| Modelo con Lenguaje Natural(HU01) | Modelo usando clases(HU01) |
| --- | --- |
| Objetos que necesito:   * Nombre:Player(Es el jugador) * Operaciones(Que hace):   + Moverse a todas las direcciones.   + Dibujarse en la pantalla. * Atributos   + Diametro   + Velocidad   + Color   + posicion |  |

## **Modelo y documentación del algoritmo(los métodos de las clases involucradas):**



## ANÁLISIS

**Historia de Usuario 02[HU02]:**

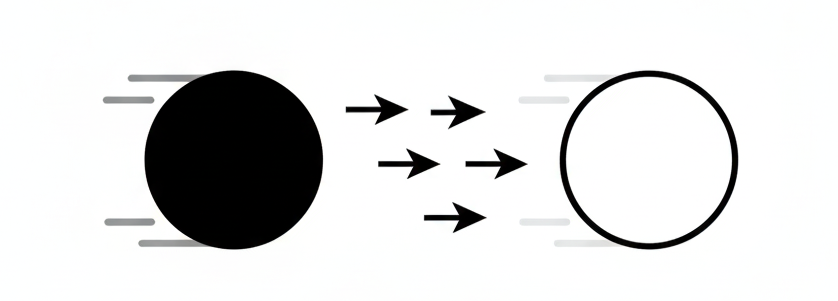
**Como** jugador **quiero** que el enemigo se pueda mover a la dirección del player **para** tocarlo.

**Precondición:** Antes de moverse debe dibujarse el enemigo.

**PostCondición:** Luego de moverse debe dibujarse el enemigo.

| Historia | Entradas | Procesos | Salida |
| --- | --- | --- | --- |
| **HU02** | Posición del jugador | Calcular vector hacia el jugador y mover enemigo | Enemigo se acerca al jugador |

**Boceto de pantalla:**



**Lista de Tareas H02:**

| **T01** | Crear clase Enemigo con posición, diámetro, velocidad. |  |
| --- | --- | --- |
| **T02** | Visualizar al enemigo con circle(). |  |
| **T03** | Crear el método mover(player): acercarse al jugador. |  |

**Criterio de Aceptación HU02:**

| C01 | El enemigo siempre se acerca al jugador. |  |
| --- | --- | --- |
| C02 | Visualización correcta. |  |

## 

## DISEÑO

| Modelo con Lenguaje Natural(HU02) | Modelo usando clases(HU02) |
| --- | --- |
| Objetos que necesito:   * Nombre:Player(Es el jugador) * Operaciones(Que hace):   + Moverse a la dirección del player   + Dibujarse en la pantalla. * Atributos   + Diametro   + Velocidad   + posicion |  |

## **Modelo y documentación del algoritmo(los métodos de las clases involucradas):**



## ANÁLISIS

**Historia de Usuario 03[HU03]:**

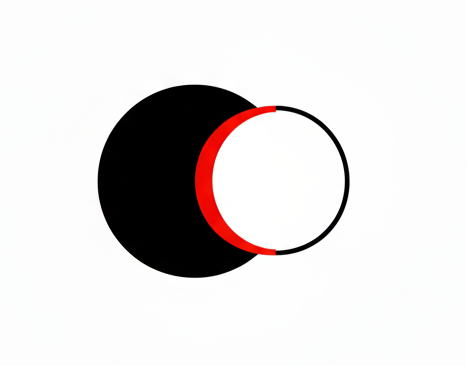
**Como** jugador **quiero** que el enemigo colisione con player **para** que suceda un evento, que el player cambie de color.

**Precondición:** Antes el enemigo y el player deben estar dibujados en la pantalla.Debe existir un sistema de detección de colisiones activo.

**PostCondición:** Si el enemigo colisiona con el player, el color del player cambia.

| Historia | Entradas | Procesos | Salida |
| --- | --- | --- | --- |
| **HU03** | Posición del jugador y enemigo | Detectar distancia entre jugador y enemigo; cambiar color si colisionan | Jugador cambia de color |

### 



## 

**Lista de Tareas H03:**

| **T01** | Aplicar deltaTime a las acciones del jugador y enemigo. |  |
| --- | --- | --- |
| **T02** | En Player, crear el método verificarColision(enemigo): cambiar color de player si colisiona. |  |

**Criterio de Aceptación HU03:**

| **C01** | Movimiento suave independiente de FPS. |  |
| --- | --- | --- |
| **C02** | Cuando un enemigo toca al jugador, este cambia de color. |  |

## 

## DISEÑO

| Modelo con Lenguaje Natural(HU03) | Modelo usando clases(HU03) |
| --- | --- |
| Objetos que necesito:   * Nombre:Player(Es el jugador) * Nombre:Enemigo * Operaciones(Que hace):   + Calcular la distancia, para la colicion |  |

## **Modelo y documentación del algoritmo(los métodos de las clases involucradas):**

