Objetivo: crear un bullet-hell (un protagonista contra muchísimos enemigos que disparan en su contra)

Fase de análisis y diseño orientado a objetos: Como jugador quiero poder moverme en un amplio escenario

Historia de usuario

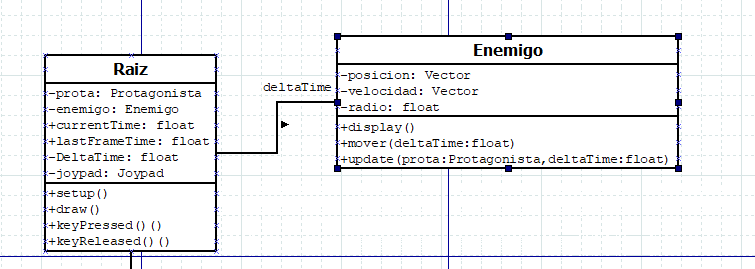
|  |  |
| --- | --- |
| Código: 001 | Usuario: Lautaro |
| Nombre de la Historia de Usuario: construcción de escenario y ubicación de game objects | |
| Prioridad: Alta | Riesgo de desarrollo: Alto |
| Estimación: 3 dias | iteración Asignada: 1 |
| Responsable: Espinosa Lautaro Eduardo | |
| Descripción: Como jugador quiero observar un escenario abierto | |
| Criterios de aceptación:  CA1: El jugador puede moverse en 8 direcciones  (poner imagen del mapa con protoenemigos)  CA2: Se crea una clase enemigo y se generan una cantidad de enemigos fija repartida aleatoriamente por el mapa  CA3: Los enemigos en pantalla pueden o no moverse por el escenario pero respetan los limites del mapa  CA4: Se generan dentro del lienzo del mapa (2400x1600) una cantidad finita esparcida aleatoriamente de cofres de la suerte  (poner imagen del mapa con protocofres) | |
| Observaciones:  CA1: El protagonista puede moverse hasta los limites establecidos del mapa (2400x1600)  CA3: Para mantener la consistencia del juego en diferentes equipos se utilizará DeltaTime | |

CA1:

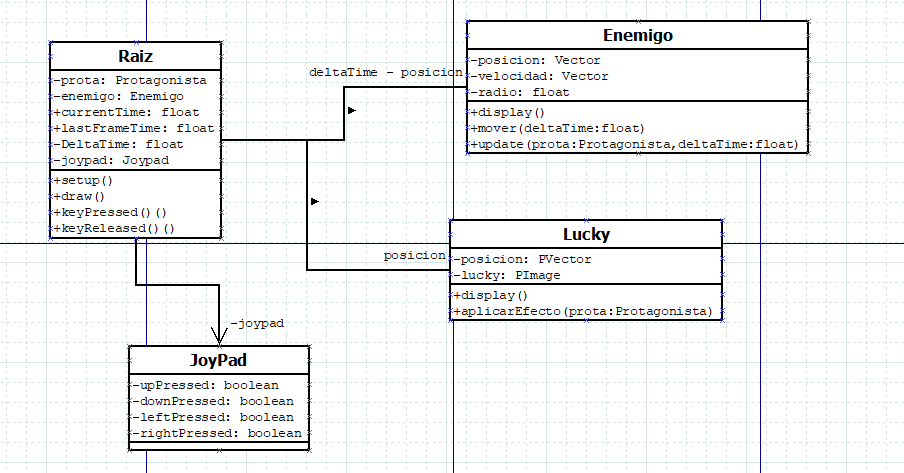
Diagrama

El contenido generado por IA puede ser incorrecto.

CA2 y CA3:

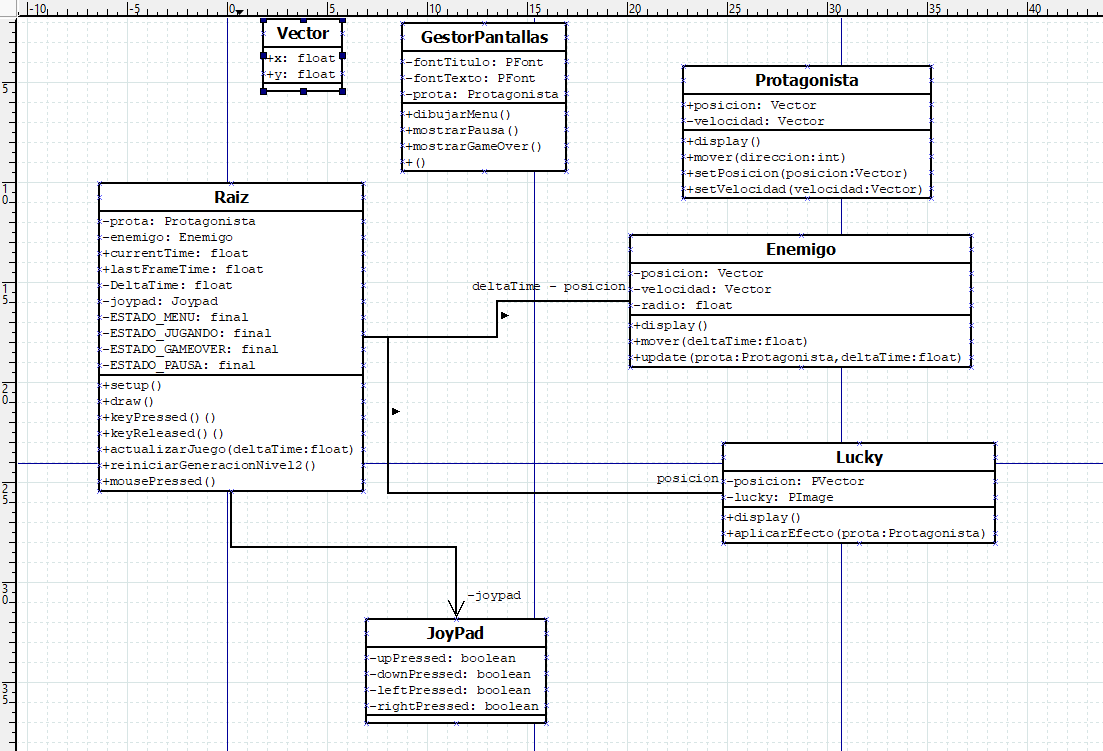


CA4:



Historia de usuario

|  |  |
| --- | --- |
| Código: 002 | Usuario: Lautaro |
| Nombre de la Historia de Usuario: construcción de una maquina de estado para el manejo del juego | |
| Prioridad: Alta | Riesgo de desarrollo: Alto |
| Estimación: 3 dias | iteración Asignada: 1 |
| Responsable: Espinosa Lautaro Eduardo | |
| Descripción: Una maquina de estado que permitirá moverse en distintas instancias del videojuego | |
| Criterios de aceptación:  CA1: Se debe crear el estado menu  CA2: Se debe crear el estado nivel 1  CA3: Se debe crear el estado Game Over  CA4: Se debe crear el estado pausa  (poner imagen) | |
| Observaciones:  El estado menú,gameover,pausa se dibujaran en una clase aparte llamada GestorPantallas | |



Historia de usuario

|  |  |
| --- | --- |
| Código: 003 | Usuario: Lautaro |
| Nombre de la Historia de Usuario: Creacion de proyectiles | |
| Prioridad: Alta | Riesgo de desarrollo: Alto |
| Estimación:1 dia | iteración Asignada: 1 |
| Responsable: Espinosa Lautaro Eduardo | |
| Descripción: Crear un sistema que permita tanto a enemigos como protagonista dispararse entre si | |
| Criterios de aceptación:  CA1: Se debe crear una clase proyectil  CA2: El protagonista puede disparar  CA3: Los enemigos deben disparar al protagonista apuntandole  CA4: Si el proyectil del protagonista choca con el enemigo, el enemigo desaparece  CA5: Si el proyectil del enemigo choca con el protagonista, el protagonista pierde 1 vida  (poner imagen) | |
| Observaciones:  El protagonista como los enemigos crean un arraylist de la clase proyectiles por cada disparo que ejecutan  La lógica de la colisión se almacena en la clase ManejadorNivel  Cada vez que el protagonista elimina un enemigo gana 100 puntos  Si el enemigo reduce las vidas del protagonista a 0, se acaba el juego | |

Imagen que contiene Tabla

El contenido generado por IA puede ser incorrecto.

Historia de usuario

|  |  |
| --- | --- |
| Código: 004 | Usuario: Lautaro |
| Nombre de la Historia de Usuario: Manejador de nivel | |
| Prioridad: Alta | Riesgo de desarrollo: Alto |
| Estimación: 5 dias | iteración Asignada: 1 |
| Responsable: Espinosa Lautaro Eduardo | |
| Descripción: Crear una que maneje la lógica de los niveles | |
| Criterios de aceptación:  El manejador de nivel debe dibujar el escenario  El manejador de nivel debe gestionar el comportamiento del enemigo  El manejador de nivel debe gestionar las colisiones de proyectiles del protagonista y enemigo  El manejador de nivel debe aplicar los efectos de los luckyblocks y eliminarlos  En caso de no completar la condición especial no se podra avanzar de nivel  (poner imagen) | |
| Observaciones: Cambiar el Estado\_Jugando a Estado\_GameOver cuando las vidas del protagonista sean igual a 0  Los proyectiles que se salgan del lienzo deben eliminarse para cuidar el rendimiento  Por cada enemigo que el protagonista elimina su puntaje aumenta en 100  En el caso de estar en el modo horda el protagonista gana 5 vidas  En el caso de estar en el modo horda por cada enemigo eliminado se generan de 1 a 3 enemigos nuevos de las diferentes variantes existentes  Existe un máximo de 100 enemigos en el campo | |

Interfaz de usuario gráfica, Tabla

El contenido generado por IA puede ser incorrecto.

Historia de usuario

|  |  |
| --- | --- |
| Código: 005 | Usuario: Lautaro |
| Nombre de la Historia de Usuario: Nivel 1 | |
| Prioridad: alta | Riesgo de desarrollo: alta |
| Estimación: 3 dias | iteración Asignada: 1 |
| Responsable: Espinosa Lautaro Eduardo | |
| Descripción: Primer nivel del videojuego | |
| Criterios de aceptación:  Se debe acceder mediante el menú interactivo  El protagonista cuenta con 500 vidas fijas  Se generan una cantidad de 24 enemigos (8 de cada tipo) se generan y se mueven suavemente por el lienzo del juego  Se generan una cantidad de 15 cofres de la suerte (la suerte se decide aleatoriamente y se genera a lo largo del lienzo del juego  Se alcanza al nivel 2 si y solo si se consigue una puntuación de 500 puntos  El juego acaba si el protagonista no tiene vidas (vidas = 0) o no hay enemigos en pantalla  (poner imagen) | |
| Observaciones:  Se puede tomar pausa en cualquier momento del videojuego  El manejo del nivel se realiza mediante la clase manejador de nivel, pero el cambio de estado yace en la función -actualizarJuego() | |

Historia de usuario

|  |  |
| --- | --- |
| Código: 006 | Usuario: Lautaro |
| Nombre de la Historia de Usuario: Nivel 2 | |
| Prioridad: alta | Riesgo de desarrollo: alta |
| Estimación: 3 dias | iteración Asignada: 1 |
| Responsable: Espinosa Lautaro Eduardo | |
| Descripción: Ultimo nivel de la prueba técnica del videojuego | |
| Criterios de aceptación:  El protagonista accede mediante obtener 500 o más puntos en el nivel anterior  El protagonista conserva las vidas del anterior nivel  Se generan una cantidad de 30 enemigos (10 de cada tipo) se generan y se mueven suavemente por el lienzo del juego  Se generan una cantidad de 12 cofres de la suerte (la suerte se decide aleatoriamente y se genera a lo largo del lienzo del juego  Se alcanza el final de la demo si y solo si se consigue una puntuación de 2500 puntos  El juego acaba si el protagonista no tiene vidas (vidas = 0) o no hay enemigos en pantalla  (poner imagen) | |
| Observaciones: | |

Historia de usuario

|  |  |
| --- | --- |
| Código: 007 | Usuario: Lautaro |
| Nombre de la Historia de Usuario: Modo Horda | |
| Prioridad: alta | Riesgo de desarrollo: alta |
| Estimación: 3 dias | iteración Asignada: 1 |
| Responsable: Espinosa Lautaro Eduardo | |
| Descripción: | |
| Criterios de aceptación:  (poner imagen) | |
| Observaciones: Se puede tomar pausa en cualquier momento del videojuego  El manejo del nivel se realiza mediante la clase manejador de nivel, pero el cambio de estado yace en la función -actualizarJuego()  En el estado\_creditos se puede volver a acceder al menú de juego donde se podrá reiniciar las partidas o jugar el modo horda | |

Historia de usuario

|  |  |
| --- | --- |
| Código: 008 | Usuario: Lautaro |
| Nombre de la Historia de Usuario: | |
| Prioridad: media-baja | Riesgo de desarrollo: bajo |
| Estimación: 1 dia | iteración Asignada: 1 |
| Responsable: Espinosa Lautaro Eduardo | |
| Descripción: Se establecerán la imagen de fondo de los niveles, menú de inicio,pausa,final y de los diferentes enemigos | |
| Criterios de aceptación:  Cada nivel deberá tener una imagen propia  El protagonista como los diferentes enemigos deben contar con una imagen que los caracterice  El menú principal debe contar con un logo e imágenes para acceder a diferentes modos  (poner imagen) | |
| Observaciones:  Según el tipo de enemigo el tamaño del mismo puede variar  Se precargan las imágenes y fuentes en el setup  El diseño de las pantallas yace en la clase gestorPantallas | |