

# Hoja de ruta y conceptos

## DeadBeans

### Índice

INTRODUCCIÓN.....	2
CONCEPTO.....	3
DISEÑO.....	4
1 Jugabilidad.....	4
2 Personajes / Enemigos / NPCs.....	4
3 Entorno y niveles.....	5
4 UI / UX.....	5
Herramientas y tecnologías.....	6
Base de datos.....	7
ENTIDADES.....	7
players.....	7
enemies.....	8
items.....	8
Especialización de items (Herencia Is_a).....	8
weapons.....	8
armors.....	8
consumables.....	8
materials.....	9
RELACIONES.....	9
drops (N:M entre enemies e items).....	9
sells (N:M entre players e items).....	9
Planes.....	10

El enlace al repositorio de github es:

<https://github.com/DFran49/DeadBeans.git>

# INTRODUCCIÓN

DeadBeans es un videojuego estilo roguelike, género el cual consiste en entrar a mazmorras o ciertos lugares y explorarlos varias veces, siendo que cada vez que se entra, la mazmorra es ligeramente distinta que las demás veces. En general todo es lo mismo excepto, la disposición, el número y el tamaño de las salas, los enemigos de cada sala, las recompensas de cada sala y la posición de ciertos puntos de interés.

La idea que tengo en mente es un estilo de juego tipo RPG, en el cuál podrás equipar a tu personaje con ciertas armas o armaduras, las cuales darán ciertos atributos o ataques al jugador para que pueda defenderse.

También quiero añadir un sistema de tienda en la que el jugador pueda vender ciertos objetos conseguidos en las mazmorras para conseguir dinero y mejorar su equipamiento.

El público objetivo de este videojuego si fuera un videojuego profesional y pensado para su publicación sería principalmente para programadores debido a que la ambientación se va a basar en conceptos de programación e informática.

Principalmente el videojuego va a estar disponible para Windows, aunque también intentaré darle soporte en sistemas basados en Linux.

El objetivo de este proyecto es introducirme al mundo de desarrollo de videojuegos y dar mis primeros pasos, aprendiendo nuevas tecnologías, acostumbrarme a nuevos entornos y en general, divertirme, porque he elegido un género de videojuegos que en lo personal me encantan.

# CONCEPTO

Todavía no tengo una historia clara para el juego, pero si tengo algunas ideas para ir definiendo la historia. Por otro lado tengo un documento en el cual voy escribiendo la historia para el juego, pero por el momento no voy a compartirlo porque no está completa.

Lo que si puedo decir es que se seguirá al protagonista, quien sin saberlo procede de una familia que lleva luchando contra la corrupción del mundo, la cual se manifiesta en forma de bugs, virus, malas prácticas y todo tipo de cosas negativas relacionadas con la programación y la informática.

Los enemigos, las habilidades, las armas, las estadísticas y el diseño de nivel va a sacar inspiración de ciertas herramientas y elementos del mundillo, tal y como puede ser por ejemplo que un escudo sea un firewall o que uno de los enemigos importantes sea Netbeans (Si, hasta ese nivel llega lo poco que nos ha gustado ese IDE).

Las mecánicas principales incluyen:

- Combate estilo RPG “en vivo”, sin seguir un sistema de turnos.
- Venta de objetos en la tienda del jugador.
- Exploración de mazmorras, las cuales contendrán objetos para vender en la tienda, para crear o mejorar el equipamiento u objetos de equipamiento adicionales que solo se podrán usar en esa mazmorra, lo cual implica que al morir o terminarla, se desechan.

Los géneros son los ya mencionados RPG y RogueLike, junto con RogueLite, Simulación, Exploración y Aventura.

Las influencias que me han hecho decantarme por este videojuego han sido videojuegos como “Moonlighter”, “Vampire Survivors”, “Risk of Rain” y más juegos de ese estilo. También han influido medios como ciertas novelas de fantasía, ciencia ficción y aventuras entre otras.

# DISEÑO

## 1 Jugabilidad

Los controles quiero que se basen en teclado con wasd y algunos otros botones para por ejemplo usar algún consumible o usar alguna habilidad especial. También quiero implementar control con gamepad, ya que este estilo de juegos suelen beneficiarse mucho de un gamepad, y puede ser interesante intentarlo.

Las fases de juego serán un bucle, siguiendo el siguiente formato:

*Entrar a la mazmorra > Explorar > Combatir > Conseguir objetos > Derrotar al jefe / Morir / Huir > Vender > Mejorar > Repetir*

Se sigue el formato RogueLike, siendo la mecánica principal la de explorar las mazmorras, pero hay cierta libertad de configuración y de disfrute.

La dificultad va a sufrir un aumento gradual y otro exponencial.

El aumento gradual se basa en que las diferentes mazmorras son cada una más difícil que la anterior, por lo cual si se entra a la segunda mazmorra, los enemigos serán más difíciles que en la primera.

El aumento exponencial se basa en el tiempo que se pasa dentro de una mazmorra. Hay un contador de tiempo que servirá como multiplicador de dificultad. Irán apareciendo enemigos en cualquier sala que no sea de jefe para atacar al jugador. Conforme más tiempo se pase en la mazmorra, más enemigos irán apareciendo, y tanto estos enemigos como los específicos de cada sala (Que se definen al generarse la sala) se harán más fuertes.

## 2 Personajes / Enemigos / NPCs

El diseño de todas estas entidades será en pixel art, siguiendo la estética futurista basada en programación e informática.

Solo habrá un personaje jugador, cuyo diseño será siempre el mismo, pero lo que cambiará será el diseño de su equipamiento cuando este cambie.

El diseño de los enemigos será variado, en cada mazmorra habrá una serie de enemigos predefinidos. Habrá enemigos generales cuyo diseño variará en función de la mazmorra en la que se encuentren.

El comportamiento de los enemigos será variado, habiendo ciertos enemigos que simplemente anden hacia el jugador, otros enemigos tendrán ciertas habilidades como cargar contra el jugador o disparar proyectiles.

Los NPCs serán también algo variados. En el caso de la tienda, una serie de NPCs la visitarán para comprar los objetos que se encuentren a la venta, y en la ciudad, habrá otros NPCs que darán diálogos genéricos junto con los NPCs de las otras tiendas, los cuales venderán o mejorarán equipamiento, pociones u objetos exóticos que puedan hacer falta.

### **3 Entorno y niveles**

La zona principal será una ciudad futurista, donde el jugador principal dispondrá de una tienda, y habrá tiendas de NPCs donde poder comprar ciertos servicios y/o productos.

En esa ciudad habrá una sub zona con la entrada a todas las mazmorras, comenzando con solo la primera desbloqueada. Cada mazmorra dispondrá de alrededor de 4 pisos, cada uno con sus salas específicas. En el caso de completar una mazmorra, se desbloquea la siguiente.

Cada mazmorra tendrá su propia inspiración para el diseño de nivel y de los enemigos, por ejemplo con algunas etapas de los avances de la tecnología.

En cada piso de cada mazmorra, se generará cierto número de salas, las cuales podrán tener recompensas, enemigos ambas o nada. Todo eso definido cuando se entra a ese piso y se generan las salas. En todos los pisos habrá al menos una sala vacía la cual será la entrada, una sala para curarse, y una sala de jefe con la entrada al siguiente piso.

### **4 UI / UX**

Habrà un menú principal para cargar la partida o volver a empezarla.

Dentro del juego habrá una interfaz estilo RPG que indicará la salud actual, el arma equipada y ciertas otras cosas pequeñas como la cantidad de dinero disponible.

También habrá menús para cada tienda, para la propia tienda del jugador y para el inventario de este.

Finalmente, habrá una pantalla de carga para cargar la partida, iniciar una nueva, para entrar a una mazmorra o avanzar de piso y para cuando se sale de una mazmorra.

# Herramientas y tecnologías

El motor de juego va a ser Unity, con un proyecto en 2D para adoptar la estética pixelart.

Se usará el lenguaje C#, utilizado por Unity y quizás se utilice Rust si decido hacer una api para el acceso a BBDD.

Lo ideal sería tener los datos del juego en local en archivos json por ejemplo o como objetos serializados, pero teniendo en cuenta la limitación mínima de tablas para la base de datos, habría demasiadas tablas para hacerlo de forma eficiente, y no quiero tener que instalar una base de datos cuando se instale el juego, por lo cual estoy barajando la opción de una api que me permita recibir los datos de los objetos y almacenarlos en json para que el propio juego pueda usarlos.

Se está usando control de versiones con git, trabajando cada característica en ramas diferentes para evitar tener conflictos entre los cambios.

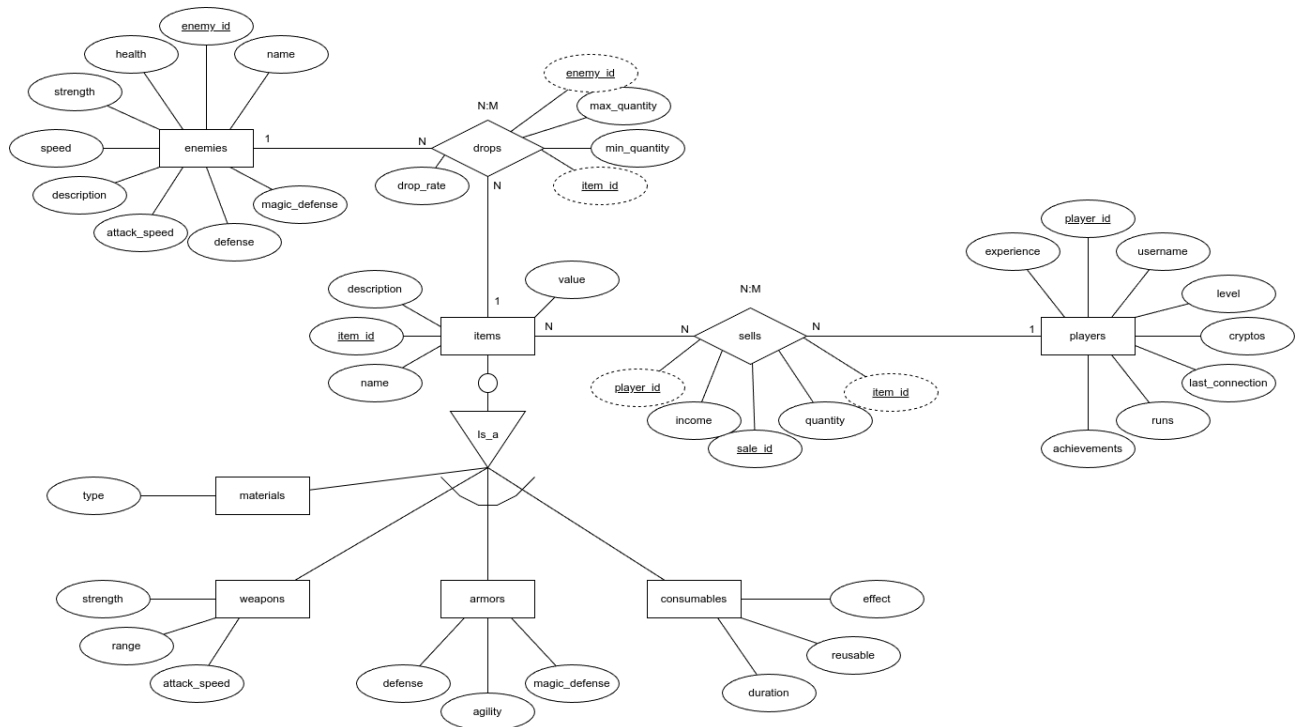
Se usarán herramientas gráficas como Aseprite para hacer el diseño de los sprites, las spritesheets, los tiles que formarán las salas, las tilesheets y en general todo el diseño gráfico del juego.

Para audio, principalmente se usará Audacity, tanto para grabar como para editar y preparar los archivos.

Finalmente, se usará una base de datos, aunque todavía no he decidido si usar una base de datos relacional (MariaDB) o una no relacional (MongoDB), esto debido a que al tener que hacer mínimo 8 tablas para la BBDD, hace que no todas tengan relación entre si, dando lugar a esa duda.

# Base de datos

La base de datos será una relacional con el siguiente diagrama EER:



Toda esta información será cargada mediante API en formato JSON a unos ficheros locales para cargar ciertas actualizaciones de algunas cosas específicas (Como el daño de un arma).

El paso a tablas es el siguiente:

## ENTIDADES

### players

- player\_id (PK)
- username - String
- experience - int
- level - int
- cryptos – int (Este campo es el dinero)
- last\_connection - date
- runs - int
- achievements - int

### **enemies**

- enemy\_id (PK)
- name - String
- health – int
- defense - int
- magic\_defense - int
- strength - float
- speed - float
- attack\_speed - float
- description - String

### **items**

- item\_id (PK)
- name - String
- description - String
- value - int

## **Especialización de items (Herencia Is\_a)**

### **weapons**

- item\_id (PK, FK to items)
- strength - float
- range - float
- attack\_speed - float

### **armors**

- item\_id (PK, FK to items)
- defense - int
- magic\_defense - int
- agility - float

### **consumables**

- item\_id (PK, FK to items)
- effect – float (Este campo es el valor del efecto)
- reusable - bool
- duration - float



### **materials**

- item\_id (PK, FK to items)
- type - String

## **RELACIONES**

### **drops (N:M entre enemies e items)**

- enemy\_id (PK, FK to enemies)
- item\_id (PK, FK to items)
- min\_quantity - int
- max\_quantity - int
- drop\_rate - float

### **sells (N:M entre players e items)**

- sale\_id (PK)
- player\_id (FK to players)
- item\_id (FK to items)
- income - int
- quantity - int

## AVANCES EN LA 2a ENTREGA

Para esta segunda entrega he hecho bastantes avances. Lo principal ha sido la decisión de dividir en pequeños componentes reutilizables el código, haciendo que si distintos elementos necesitan código similar, se reutilice.

Por ejemplo, en la entidad jugador y en la entidad enemigo se necesita el sistema de combate (Que está dividido en el componente de salud y en el componente de estadísticas), entonces se les asignan los scripts pertinentes y ya está listo para usarse.

También he creado un menú de inicio que permite salir del juego, empezar una partida o acceder a unas opciones.

Al comenzar el juego, se spawnea al jugador y conforme pasa el tiempo se van spawneando enemigos. Cuando se elimina a un enemigo, se consiguen Cryptos, la moneda de este juego. Si el jugador muere, se espera 5 segundos y se le vuelve a hacer aparecer, con toda la salud y sin Cryptos.

También he hecho una GUI que refleja el estado de salud actual del jugador y su cantidad de Cryptos.

# Planes

Los principales planes son hacer unas 4 mazmorras con unos cuantos pisos cada una. Los pisos se generarán automáticamente al entrar a estos a partir de una “pool” de salas pregeneradas a las cuales tras eso se les asignarán enemigos y botín.

Una api en rust para recibir los datos de la BBDD sobre los objetos y enemigos, y actualizar las estadísticas de forma sencilla en caso de necesitar hacer un cambio de balance.

Diseñar mi propia arte para las tiles y los sprites, dándole un toque personalizado y más consistente al juego.

Completar la línea de historia y aplicarla de forma atractiva.

Sistema de combate con diferentes armas.

Lo principal que he investigado han sido las animaciones, cosa que ha sido muy fructífera.

## Nuevos planes

Los planes anteriores siguen en pie, pero ahora es muy posible que no haga mi propio arte ya que es muy posible que no me de tiempo.

Hacer un sistema de inventario para que cuando se elimine a un enemigo pueda soltar objetos y que estos se puedan recoger.

Hacer diferentes armas.

Hacer que los enemigos tengan ataques variados.

Hacer algunas habilidades para el jugador y un sistema de subida de nivel.

Hacer que se puedan conseguir ciertos objetos que potencien ciertos aspectos del jugador.

Hacer presets de más enemigos para dar variedad.

Hacer presets de salas para al menos hacer una mazmorra.