**Dungeons and Cards**

Dungeons and Cards is a prototype roguelike deck-builder game. In this solo adventure you will fight with cards, build your deck and explore the mysterious dungeon. It will contain: an enemy, a card battle, 20 cards, 3 consumable objects, an inventory system, a main character with movement and a dungeon.

**MVP:**

Será un prototipo que pruebe las mecánicas básicas de un juego roguelike deckbuilder que no se centrará en el “gameplay”. Nos enfocaremos en el funcionamiento de las mecánicas. Incluirá lo siguiente:

* Crear mazmorras por procedimientos en unity
* Una pelea de cartas contra un enemigo (solo para verificar el funcionamiento de 20 cartas)
* Movimiento de un personaje (izquierda, derecha, arriba, abajo)
* Uso funcional de 3 objetos (desde el inicio de la mazmorra se tendrán esto solo para verificar su funcionamiento)
* Un tipo de enemigo disponible (para probar la pelea)

**Aprendizajes:**

* Uso práctico de la programación orientada a objetos
* Funcionamiento de algoritmos para generar mapas por procedimientos
* Uso básico de Unity
* Conocimiento básico de la programación de mecánicas de videojuegos

Clases

* Personaje
  + Protagonista
  + Enemigo
  + Jefe
* Carta
  + Cartas de ataque
  + Cartas de apoyo
* Objeto consumible
  + Pociones
    - Curación
    - Daño
    - Boost
  + A DEFINIR
* Terreno
  + Terreno caminable
  + Terreno pared
  + Trampa

UML

**Requerimientos:**

Botones para el menú que permitan iniciar el juego

Movimiento del protagonista con las teclas W, A, S, D

Consumo de objetos (pociones, etc) durante exploración y combate

Seleccionar y usar cartas

Juego 2D

Solo 5 cartas

1 jefe

1 personaje

Juego a pixelart

Moverse por un calabozo

el jugador puede administrar su inventario (revisar las cartas que tiene y objetos)

* + **Requerimientos funcionales:**
    - El jugador puede ver y administrar su inventario (revisar las cartas que tiene, ver sus objetos y puede soltar lo que no desee)
    - El jugador puede mover al personaje principal a la derecha, a la izquierda, arriba y abajo
    - El jugador puede consumir objetos durante la pelea y durante la exploración
    - El jugador puede seleccionar cartas y usarlas
    - Para entrar en pelea el jugador debe acercarse al enemigo
    - El jugador tiene que poder ver su vida actual
    - Cuando el jugador muere tiene que iniciar desde el principio la mazmorra
  + **Requerimientos no funcionales:**
    - Programar el juego utilizando C# con Unity
    - Ejecutable en Windows
    - El juego será ejecutado por un usuario a la vez

**Mecánicas del juego y contexto:**

Se contará con dos fases: la fase de exploración y la fase de pelea. La fase de pelea cuenta con dos fases. Primero eliges tus cartas y está el combate. El juego siempre inicia en la fase de exploración de la fase de exploración puedes pasar a la de pelea y de la de pelea a la de exploración.

**Fase de exploración:**

* Pierdes vida al chocar con trampas
* Puedes ver en tu inventario los objetos consumibles y las cartas. El inventario tendrá una pestaña de consumibles y otra de cartas
* Recoger objetos consumibles, doblones y cartas
* Encontrarte con enemigos y acercarte para pelear con ellos. Esto te llevará a la fase de pelea
* Cuando derrotas a un enemigo en la fase de pelea y regresas a la de exploración en el lugar donde se encontraba el enemigo encontrarás doblones

**Fase de pelea:**

1. Elegir cartas. Se mostrarán las cartas que tienes y seleccionarás 20 cartas. Puedes escoger 20 cartas del mismo tipo o de tipos diferentes.
2. Pelea de cartas.
   1. Usas cartas para derrotar al enemigo
   2. Pierdes vida si te ataca el enemigo
   3. Puedes consumir objetos consumibles
   4. Regresas a la fase de exploración una vez que derrotas al enemigo

**Sobre las cartas:**

Se contará con 5 tipos de cartas diferentes. Puedes tener varias cartas del mismo tipo. Las cartas las encuentras en paquetes de cartas.

**Sobre los objetos consumibles:**

Cuando hablamos de objetos consumibles nos referimos a objetos que solo pueden usarse una vez. Contaremos con 3 tipos de objetos consumibles. Un tipo te aumentará el daño de la carta, otro aumentará la defensa de la carta, y finalmente, otro te curará.

**Historias de Usuario**

**Durante fase de Exploración**

* Historia de usuario 1: Movimiento en el escenario
  + Como jugador, quiero poder moverme en el escenario de la fase de exploración hacia arriba, abajo, izquierda y derecha para poder explorar el escenario.
  + Criterios de aceptación:
    - El escenario tiene un tamaño de 840x840 pixeles
    - El escenario tiene una resolución de 16x16 píxeles
    - El escenario tiene 8 cuartos de 140x140 píxeles
* Historia de usuario 2: Recoger objetos del escenario de exploración
  + Como jugador, quiero poder recoger objetos (consumibles, doblones y cartas) del suelo del escenario de exploración para poder usarlos durante la exploración y la pelea.
  + Criterio de aceptación:
    - En el escenario hay objetos para recoger.
    - El jugador puede recoger los objetos.
* Historia de usuario 3: Administrar inventario del usuario
  + Como jugador, quiero poder mover y eliminar objetos de mis inventarios de cartas y objetos consumibles, para hacer espacio para otras cartas u objetos.
  + Criterio de aceptación:
    - El jugador cuenta con un inventario para cartas y otro para consumibles.
    - El jugador cuenta con objetos dentro de sus inventarios.
    - El juego permite eliminar los objetos.
* Historia de usuario 4: De fase de exploración a fase de pelea
  + Como jugador, quiero pasar de la fase de exploración a la fase de pelea al acercarme a una distancia de tres espacios (el jugador abarca un espacio) para poder pelear con cartas.
  + Criterios de aceptación:
    - El escenario cuenta con un enemigo.
    - El jugador cuenta con 20 cartas en su mazo.

**Durante fase de pelea**

* Historia de usuario 5: Seleccionar cartas antes del combate
  + Como jugador, quiero poder escoger veinte cartas antes del combate para poder pelear utilizándolas.
  + Criterio de aceptación:
    - El jugador tiene un inventario del que seleccionará cartas.
    - El jugador puede pelear con enemigos usando cartas.
* Historia de usuario 6: Pelea
  + Como jugador, quiero poder utilizar mis cartas durante la pelea con enemigos para poder hacer daño o asistirme con puntos de vida adicionales utilizando consumibles, pudiendo yo perder puntos de vida en la pelea.
  + Criterio de aceptación:
    - Durante la pelea:
      * El jugador cuenta con un mazo de cartas.
      * El jugador puede recuperar puntos de vida.
      * El jugador puede infligir daño.
      * El jugador puede recibir daño.
      * El enemigo puede recibir daño.
      * El enemigo puede infligir daño.

* Historia de usuario 7: De fase pelea a fase de exploración
  + Como jugador, quiero pasar de la fase de pelea a la de exploración tras haber derrotado al enemigo para seguir explorando y pelear con otros enemigos.
  + Criterio de aceptación:
    - La pelea debe haber terminado.
    - El jugador debe haber ganado la pelea.

**Durante ambas fases**

* Historia de usuario 8: Ver y ocultar inventario del jugador
  + Como jugador, quiero ver mis objetos disponibles en general (consumibles y cartas), pero también ocultarlos si ya no quiero verlos.
  + Criterio de aceptación:
    - El jugador cuenta con un inventario.
    - El jugador cuenta con objetos dentro de su inventario.

**Requerimientos no funcionales:**

* Programar el juego utilizando C# con Unity
* Ejecutable en Windows
* El juego será ejecutado por un usuario a la vez
* Como **<Usuario>**
* Quiero **<algún objetivo>**
* Para que **<motivo>**

**Clases del programa**

* tipoPersonaje:
  + Tipo: abstracta
  + Jerarquía: superclase
  + Atributos:
    - private int puntosDeVida;
    - private String nombrePersonaje;
    - private array[int] posicionPersonaje;
  + Métodos:
    - realizarMovimiento(array[int] posicionPersonaje);
    - imprimirVida(int puntosDeVida);
    - imprimirNombre(String nombrePersonaje);
    - utilizarObjetoConsumible;
    - utilizarCarta;
* personajeProtagonista:
  + Tipo: concreta
  + Jerarquía: hereda de tipoPersonaje
  + Atributos:
    - private int puntosDeVida;
    - private String nombrePersonaje;
    - private array[int] posicionPersonaje;
  + Métodos:
    - + realizarMovimiento(array[int] posicionPersonaje);
    - + imprimirVida(int puntosDeVida);
    - + imprimirNombre(String nombrePersonaje);
    - + utilizarObjetoConsumible;
    - + utilizarCarta;
* personajeEnemigo:
  + Tipo: concreta
  + Jerarquía: hereda de tipoPersonaje
  + Atributos:
    - private int puntosDeVida;
    - private String nombrePersonaje;
    - private array[int] posicionPersonaje;
  + Métodos:
    - + realizarMovimiento(array[int] posicionPersonaje);
    - + imprimirVida(int puntosDeVida);
    - + imprimirNombre(String nombrePersonaje);
    - + utilizarObjetoConsumible;
    - + utilizarCarta;
* pocionConsumible:
  + Tipo: abstracta
  + Jerarquía: superclase
  + Atributos:
    - int puntosVida;
    - int duracionEfecto;
    - String nombrePocion;
    - String descripcionPocion;
  + Métodos:
    - surtirEfecto(int puntosVida, int duracionEfecto);
    - imprimirNombrePocion(String nombrePocion);
    - imprimirDescripcionPocion(descripcionPocion);
* pocionCuracion:
  + Tipo: concreta
  + Jerarquía: hereda de pocionConsumible
  + Atributos:
    - int puntosVida;
    - int duracionEfecto;
    - String nombrePocion;
    - String descripcionPocion;
  + Métodos:
    - surtirEfecto(int puntosVida, int duracionEfecto);
    - imprimirNombrePocion(String nombrePocion);
    - imprimirDescripcionPocion(descripcionPocion);
* pocionRegeneracion:
  + Tipo: concreta
  + Jerarquía: hereda de pocionConsumible
  + Atributos:
    - int puntosVida;
    - int duracionEfecto;
    - String nombrePocion;
    - String descripcionPocion;
  + Métodos:
    - surtirEfecto(int puntosVida, int duracionEfecto);
    - imprimirNombrePocion(String nombrePocion);
    - imprimirDescripcionPocion(String descripcionPocion);
* cartaDeJuego:
  + Tipo: concreta
  + Jerarquía: superclase
  + Atributos:
    - private int valorAtributoAtaque;
    - private int valorAtributoDefensa;
    - private String nombreCarta;
    - private String descripcionCarta;
  + Métodos:
    - imprimirNombreCarta(String nombrePocion);
    - imprimirDescripcionCarta(String descripcionCarta);