

CUBE MADNESS

How to Play

Basic Controls

Movement

- A & D o Izquierda & Derecha para movimiento direccional.

Goal

- Evadir los obstáculos
- Conseguir la mayor cantidad de píldoras posibles con el uso de tus habilidades matemáticas (eligiendo la pared más ventajosa).
- Derrotar a los virus con las píldoras curativas.
- Llegar a la meta para seguir avanzando en los niveles.

Scripts

Hazard handler

- Heredacion de logica base de tipos de hazard, para simplificar clases y callings de logica
- Implementacion de interfaz de hazards para que se ejecute la logica base desde el dot controller o player segun corresponda

Monobehaviour decrease

- Refactor de clases para decrementar la cantidad de monobehaviours requeridos en escena

Arquitectura del proyecto

- Hazards:
 - Base Hazard para manejar la arquitectura base
 - Cada Hazard luego overridea la funcion con su logica especifica
 - Implementa una interfaz para la llamada del DotController para la logica al respecto del Hazard
- Player:
 - Separado en DotController, Pooling y MiniPlayers
 - El DotController es el que maneja el jugador en si, contiene toda la logica y el trackeo de MiniPlayers como la lista que los mantiene para que lo siga. Maneja la logica matematica en cuanto a los cambios del pool, y maneja la llamada del cambio de nivel
 - El Pooling instancia a todos los MiniPlayers para que spawnen cerca del Dot y le setea lo valores base
 - Los MiniPlayers manejan la logica de colisiones ya que cada uno puede colisionar singularmente con un objeto en particular y ahi llaman al Dot para que maneje la logica de la respuesta.
- Enemies:
 - Usan la PoolManager para instanciar los distintos enemigos
 - Tienen un Spawner para determinar la cantidad y posicion de spawn de los enemigos
 - Cuentan con un Controller el cual maneja la logica de las colisiones y los valores basicos como por ejemplo, la velocidad
- UI:
 - Es el script que maneja el flujo de los niveles y las funciones principales para las transiciones entre escenas, utilizando cargas asincronicas mediante los Addressables cuando es necesario.

Tecnicas

Adressable local

Static groping

- Basado en objetos estaticos de escena

Static batching

Static Batching, ya que hay varios elementos estaticos en la escena los cuales fueron marcados como estaticos debidamente

Frustum Culling

El frustum culling de la camara fue modificado para que el angulo de recorte sea apropiado

Reducción de materiales y mallas optimizadas

Contamos con materiales y shaders simples. ademas de poseer mallas optimizadas ya que los modelos que estamos usando son las formas geometricas por defecto de Unity

Uso de shaders livianos o simplificados

El juego cuenta con shaders personalizados los cuales no tienen calculos complejos, evitando los materiales por defecto del Unity.

Occlusion Culling

Aplicamos Occlusion a la camara para evitar el renderizado de objetos que no se esten viendo en el momento de ejecucion

Extra

Los elementos secundarios del mapa no castean sombras, y otros, como la lava, tampoco las reciben (esto bajo una barbaridad la cantidad de batches)

3 niveles

Pantalla de carga/transición entre escenas

Sistema de guardado y carga de partida

Canvas optimization

Canvas group

Reemplazamos la utilización de SetActive por el CanvasGroup el cual evita que Unity recalcule todo el Canvas

Multiples Canvas

Utilizamos multiples canvas para subdividir los elementos de la interfaz

Imagenes en vez de texto

Utilizamos imagenes para reemplazar el texto fijo del juego en elementos como el contador de niveles