

Documentacion

Elementos usados:

Plugins:

Sin plugins por el momento.

Fonts:

→ Konorke

(Fuente libre para cualquier uso).

→ Sofija

(Fuente libre para cualquier uso).

Prefabs:

→ Cube Character

(Prefab del personaje principal)

Scenes:

→ Lobby

(Escena de la tienda e inventario)

→ Nivel

(Escena de gameplay - nivel)

Imágenes:

Sprites de totalmente originales de figure wars diseñados por Juan Diego Mozo.

Backgrounds:

→ Background_Lvl1

(Fondo para el nivel 1)

→ Background_Lvl2

(Fondo para el nivel 2)

→ Background_Lvl3
(Fondo para el nivel 3)

→ Background_Lvl4
(Fondo para el nivel 4)

Sprites:

- Cards:
 - CardGeneric(Sprite de la carta de la UI mostrada para seleccionar un personaje genérico)
 - CardCircle(Sprite de la carta de la UI mostrada para seleccionar el personaje círculo)
 - CardTriangle(Sprite de la carta de la UI mostrada para seleccionar el personaje triángulo)
 - CardSquad(Sprite de la carta de la UI mostrada para seleccionar el personaje cuadrado)

- Castle:
 - Castle1 (Sprite del castillo del lado del jugador)
 - Castle2 (Sprite del castillo enemigo)

- Characters:

→ Circle
(Sprite del personaje en forma de círculo que manejamos)

→ Cube
(Sprite del personaje en forma de cuadro que manejamos)

→ Triangle
(Sprite del personaje en forma de triángulo que manejamos)

- **Currency:**
(Sprite del currency)
- **Enemies:**

(Sprite del personaje enemigo)
- **InvocationPoint**
 - (Sprite del punto de spawn numero uno)
 - (Sprite del segundo punto de spawn)
- **Road:**
- **Roaddown**
(Fila de la parte baja del mapa)
- **RoadMidle**
(Fila de la parte del medio del mapa)
- **RoadTop**
(Fila de la parte alta del mapa)
- **RoadUnder**
(Tierra de la parte mas baja del mapa)

UI:

Buttons:

→ **UI_PrincipalBackButtonConsumir**
(Botón para consumir item del inventario)

→ **UI_PrincipalBackButtonDiscard**
(Botón para descartar item del inventario)

→ UI_PrincipalButton

(Botón de opciones -Tienda-Campaña-Inventario)

→ UI_PrincipalButtonPRESS

(Botón de opciones -Tienda-Campaña-Inventario presionado)

→ UI_PrincipalButtonBackButton

(Boton sin implementar)

Currency:

→ UI_Currency

(Diseño del espacio que muestra la cantidad de currency)

→ UI_Currency+

(Aumenta la cantidad a descartar de ítems y currency)

→ UI_Currency-

(Disminuye la cantidad a descartar de ítems y currency)

→ UI_CurrencyEmpty

(Espacio donde se muestra la cantidad a descartar de ítems y el currency)

HUD:

- HUDBar

(Barra en donde van las cartas de los personajes)

- HUDPauseButto

(Boton de pausa para dicha interfaz)

Inventory:

→ UI_InventoryConsumable

(Espacio para mostrar los items en el inventario de tipo consumibles)

→ UI_InventoryNonConsumable

(Espacio para mostrar los items en el inventario de tipo No consumibles-
Con un breve tamaño y color)

→ UI_Character

(Imagen con el diseño de los 3 personajes principales)

Items:

Consumibles:

→ ItemConsumable1

(Diseño del primer ítem de tipo consumible)

→ ItemConsumable2

(Diseño del segundo ítem de tipo consumible)

→ ItemConsumable3

(Diseño del tercer ítem de tipo consumible)

No Consumibles:

→ ItemNonConsumable1

(Diseño del primer ítem de tipo no consumible)

→ ItemNonConsumable2

(Diseño del segundo ítem de tipo no consumible)

→ ItemNonConsumable3

(Diseño del tercer ítem de tipo no consumible)

→ ItemNonConsumable4

(Diseño del cuarto ítem de tipo no consumible)

→ ItemNonConsumable5

(Diseño del quinto ítem de tipo no consumible)

→ ItemNonConsumable6

(Diseño del sexto ítem de tipo no consumible)

→ ItemNonConsumable7

(Diseño del séptimo ítem de tipo no consumible)

→ ItemNonConsumable8

(Diseño del octavo ítem de tipo no consumible)

→ ItemNonConsumable9

(Diseño del noveno ítem de tipo no consumible)

→ ItemNonConsumable10

(Diseño del décimo ítem de tipo no consumible)

→ ItemNonConsumable11

(Diseño del onceavo ítem de tipo no consumible)

→ ItemNonConsumable12

(Diseño del doceavo ítem de tipo no consumible)

Levels:

→ ButtonLevel

(Sprite de un botón vacío para representar un nivel del juego)

→ ButtonLevelClosed

(Sprite de un botón con un candado para representar un nivel bloqueado)

Logos:

→ UILogo_Campaign

(Logo correspondiente al botón de jugar campaña)

→ UILogo_Chest

(Logo correspondiente al botón para usar el inventario)

→ UILogo_Currency

(Logo correspondiente al botón para usar la tienda)

→ UILogo_Settings

(Boton todavia no implementado)

Message:

→ Message_Compra

(Imagen png sin fondo del texto que se muestra cuando una compra es exitosa)

→ **Message_Consumido**

(Imagen png sin fondo del texto que se muestra cuando un ítem es consumido)

→ **Message_Descarte**

(Imagen png sin fondo del texto que se muestra cuando un ítem es descartado y además se ajusta para mostrar el currency obtenido)

→ **Message_LoTienes**

(Imagen png sin fondo del texto que se muestra cuando se intenta volver a comprar un ítem que ya se tiene)

→ **Message_NoTienesConsumibles**

(Imagen png sin fondo del texto que se muestra cuando se agota la cantidad de ítems consumibles)

→ **Message_SinDinero**

(Imagen png sin fondo del texto que se muestra cuando se intenta comprar algo de la tienda y no hay dinero disponible)

PauseMode:

→ **PauseModeB1**

(Barra de pausa)

→ **PauseModeB1**

(Barra de pausa)

→ **PauseModeB1**

(Barra de pausa)

→ **PauseModeBar**

(Barra de pausa con espacio)

Store:

→ UIStore_ButtonBuy

(Botón verde que al presionar realiza la acción de comprar)

→ UIStoreConsumable

(Columna de 3 casillas en donde estan los items de tipo consumible)

→ UIStoreNonConsumable

(Plantilla compuesta de 3 filas con el fin de que cada fila almacena hasta 4 botones)

→ UIStoreConsumable_Basic

(Botón para los ítems de tipo consumible que se ubican en la plantilla de consumibles)

→ UIStoreNonConsumable_Basic

(Botón para los ítems de tipo no consumible que se ubican en la plantilla)

→ UIStoreNonConsumable_Blocked

(Tile de la tienda aún no implementado)

→ UIStore_Nonconsumable_Bought

(Tile de la tienda aún no implementado)

→ UIStoreNonConsumable_Empty

(Tile de la tienda aún no implementado)

→ UIStoreNonConsumable_Equiped

(Tile de la tienda aún no implementado)

Sonidos:

Por el momento no se tienen sonidos

Efectos de partículas:

Por el momento no se tienen partículas

Scripts de figure wars:

Gameplay:

→ Character

(Script que controla el movimiento del personaje , en la función update se actualiza la posición frame por frame para que avance a la derecha con una cantidad dada)

→ DragCharacter

(En el script se declaran varios campos serializados para la posición , el índice , generador y posición inicial , además de una propiedad de tipo booleana activo , en el update se maneja el input para arrastrar el personaje con condicional , la función drop se encarga de de con condicionales cuadrar la posición donde se soltara el personaje y las funciones de ontrigger enter y ontrigger stay permiten validar en cual línea estará el personaje)

→ GeneratorManager

(Script encargado de controlar el generador de personajes , con propiedades serializables de personajes , líneas , arrastrables , manager de recursos , enemigos y puntos de spawn, una función spawnear personajes que tiene dos parámetros enteros de index(índice) y line(línea) , instancia los personajes , en su correspondiente posición y orientación , además se tiene otra función para validar los personajes a arrastrar , además se usa el método de spawn de enemigos)

→ MouseOver

(Script encargado del mouse)

→ Pause

(Script encargado de pausar el juego con dos propiedades de tipo GameObject y booleanas llamadas panel y pausa , en el método start a

pausa se le asigna un valor falso y se cuenta con un método OnPause que se encarga de detener el juego y otros dos métodos Restart y Exit , el primero para reiniciar el nivel uno y el segundo para ir al lobby)

→ Castle

(Script para el castillo , tiene dos propiedades privadas de tipo flotante para la vida y el tiempo de reconstrucción y trabajan con dos funciones vacías de tomar daño , la cual tiene un parámetro de tipo flotante de cantidad y reconstrucción que tiene el mismo parámetro)

→ CameraSwipe

(Script con dos propiedades , la primera de privada de tipo flotante llamada velocidad y una privada de tipo generatorManager llamada generador)

→ Actor

(Clase abstracta con propiedades flotante protegidas llamadas life (vida) , attack (ataque) , def (defensa) y speed (velocidad y un método vacío para el movimiento de la entidad , además de un método OnCollisionEnter2D vacío para eventos que se desencadenan con una colisión);

→ Character

(Script con método de activar power up)

→ Enemy

(Script de enemigo generico)

→ Generic

(Script generico)

→ Boss

(Jefe del nivel)

→ Square

(Son un tipo de personajes que se pueden elegir a la medida del juego)

→ Circle

(Son un tipo de personajes que se pueden elegir a la medida del juego)

→ Triangle

(Son un tipo de personajes que se pueden elegir a la medida del juego)

→ Generic

(Script de un enemigo genérico)

→ Boss

(Script de un boss generico)

→ ResourceManager

(Script con oro de tipo entero y con nivel de protección público , también otra propiedad de tipo entero privada de cantidad por tiempo , las otras propiedades privadas son un texto de tipo Text y un flotante añadir oro . además se tienen tres métodos de tipo vacío que se encargan de añadir oro , añadir oro por tiempo y remover oro , añadir oro y remover oro tienen un parámetro de tipo entero cada una , la primera de añadir y la segunda de cantidad)

→ Castle

(Script de un boss generico)

→ Store

(Script de la tienda se declaran las propiedades , lista items , inventario Player inventory y UI manager ui ; En el awake se instancian todos los items no consumibles con sus propiedades específicas , se añaden los ítems de tipo no consumible a la lista de items , en el método sell se recibe un índice que especificara cuál ítem se vende ,se asigna el índice a la lista de items para saber cuál ítem del tipo no consumible se venderá , se valida si la cantidad de oro es menor o igual al valor del ítem y si esto se cumple se verifica si el ítem se compró , si esto ocurre se añade el ítem al inventario del jugador , se resta el valor del ítem al oro del jugador , el estado de compra del ítem cambia a verdadero y se muestra el mensaje de que la compra fue exitosa , seguido se llaman game objects de la ui para mostrar los mensajes de que se tiene el ítem y de que no se dispone de dinero)

→ Inventory

(Script encargado del inventario , se declaran las propiedades , se crean los power ups y se añaden a la lista de power ups , se equipa , se consume los power ups , , se equipan los ítems con otra función.)

→ Manager

(Se declara la tienda y se instancia el power up attack con ciertos valores)

→ Ulmanager

(Script encargado de manejar todo lo referente con la ui)

→ Item

(Script para declarar la propiedad pública de tipo entero item)

→ Consumable

(Script que hereda de Item y define tres propiedades para estos ítems , llamadas porcentaje , duración que son de tipo protected flotantes y cantidad con un nivel de protección pública de tipo entero , además de tener una función de consumir)

→ NonConsumable

(Script que hereda de Ítem , tiene tres propiedades con un nivel de protección protegido de tipo flotantes las cuales son atk(ataque) , def(defensa) y vel(velocidad), además posee una propiedad index(índice) con nivel de protección publico de tipo entero , se definen otras dos propiedades con nivel de protección público y de tipo booleano las cuales son bought(Comprado) y equiped(Equipado), además contiene dos métodos de equipar y desequipar)

→ Currency

(Enum de tres tipos , 1 ,2 y 3)

→ Armor

(Script que hereda del tipo no consumible , en el constructor tiene los parámetros de tipo flotante attack(Ataque) , defensa(def) y speed(velocidad) , también se tienen dos parámetros de tipo entero llamados precio y ind y un parámetro de tipo booleano , todos estos parámetros serán definidos en el constructor)

→ Hair

(Script que hereda del tipo no consumible , en el constructor tiene los parámetros de tipo flotante attack(Ataque) , defensa(def) y speed(velocidad) , también se tienen dos parámetros de tipo entero llamados precio y ind , todos estos parámetros serán definidos en el constructor)

→ Weapon

(Script que hereda del tipo no consumible , en el constructor tiene los parámetros de tipo flotante attack(Ataque) , defensa(def) y speed(velocidad) , también se tienen dos parámetros de tipo entero llamados precio y ind y un parámetro de tipo booleano , todos estos parámetros serán definidos en el constructor)

→ PowerAtk

(Hereda de consumible ,se declara el manager de ui , en el constructor de la función se reciben los parámetros precio , porcentaje y duración y se asignan estos valores entero , flotante y flotante a unas nuevas variables , también se le asigna el valor cero a la variable cantidad y se usa el método consumido de la ui)

→ PowerDef

(Hereda de consumible ,en el constructor de la función se reciben los parámetros precio , porcentaje y duración y se asignan estos valores entero , flotante y flotante a unas nuevas variables , también se le asigna el valor cero a la variable cantidad y se tiene un método consume)

→ PowerSpeed

(Hereda de consumible ,en el constructor de la función se reciben los parámetros precio , porcentaje y duración y se asignan estos valores entero , flotante y flotante a unas nuevas variables , también se le asigna el valor cero a la variable cantidad y se tiene un método consume)

Bugs:

→ Resolución incorrecta.

→ Demás bugs corregidos.

Sorteo ocurrencia:

El juego solo trabaja con una ocurrencia y trabajara con un tipo de currency hasta el final.