

Instituto Técnico de Costa Rica

Manual Técnico

Proyecto

Campaña 1856-1857

Integrantes

- Luis Alfaro Enrique Alfaro Haro
- Mauricio Díaz Cernas
- José Gabriel Vindas Brenes
- Edgar Parra Barillas

Proyecto de Software

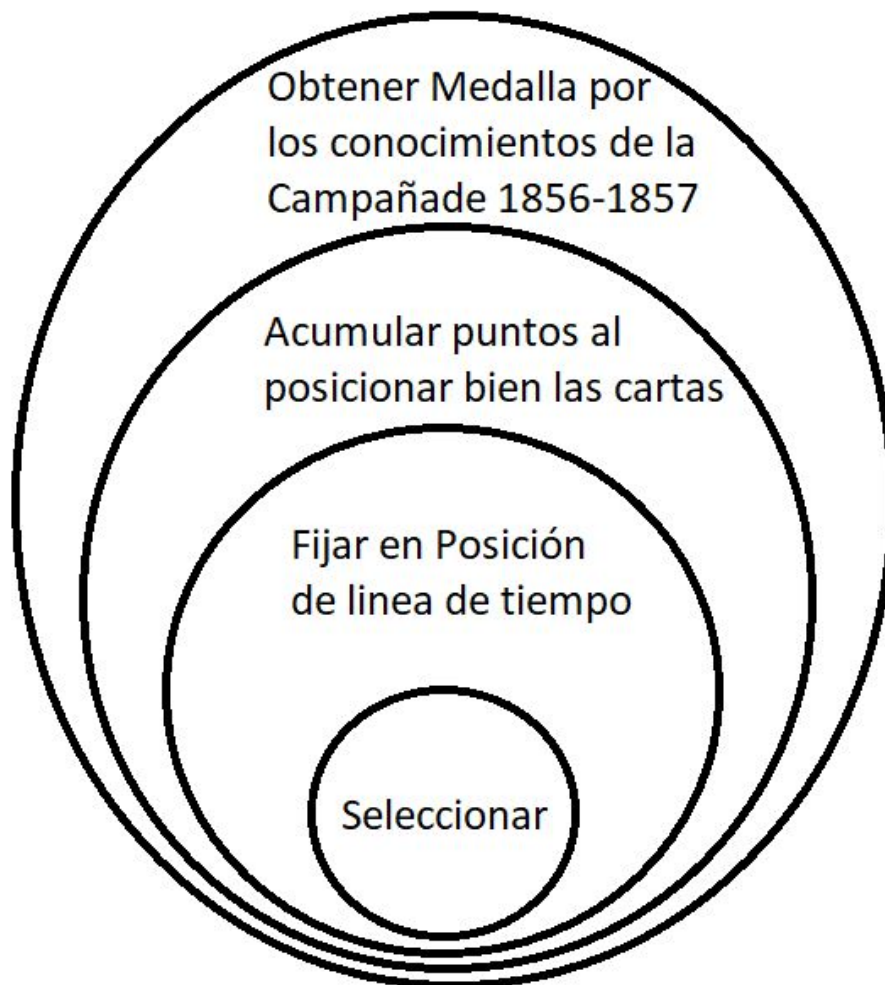
2020

Objetivo:

Elaborar un videojuego basado en un juego físico de línea de tiempo de cartas con contexto de la campaña nacional ante las fuerzas militares externas de los Estados Unidos de 1856 - 1857

Alcance:

El alcance del sistema es implementar las mecánicas principales del juego de mesa en sistema de puntos , validación de fechas , cambio de turnos.



En la imagen se puede observar la mecánica principal , secundaria , de progresión y narración del juego. El juego está destinado a un público meta de estudiantes de primer y segundo ciclo de colegios públicos. Desea narrar las principales y más relevantes acciones de la historia bélica de la campaña militar de 1856 y 1857.

Descripción de Guardado de Datos

El desarrollo se realizó en la plataforma de desarrollo Unity , en el cual utilizamos las funciones de serialización de los datos que esta tiene implementada para el guardado de los datos. Esto es el proceso automático de transformar estructuras de datos o objetos en un formato en el cual Unity puede guardar y reestructurar luego. Esto se aplica en las clases implementadas por el desarrollador para el guardado de los jugadores con la información:

Jugador
+nombre: string +apellido: string +usuario: string +personaje: string +puntaje: int
+AddCardToPlayerHand() +RemoveCardFromPlayerHand() +SetPlayerHand() +GetHand()

Se utiliza una clase serializable de Jugador para el guardado posterior de los jugadores que han ingresado al sistema.

En el caso de la información de las cartas se utilizó un sistema de Guardado de Unity llamada Scriptable Object que unity lo define como un contenedor de datos en el cual se puede guardar grandes cantidades de datos, independiente de la instancia de clases.

Card
+cardID: int +cardName: string +description: string +anno: int +mes: int +dia: int +posición x: int +posición y: int +artworkFront: sprite +artworkBack: sprite
+Print() +GetFecha() +GetNombreMes() +Operation1()

Esta forma de guardado hace que el juego tenga implementadas ya la construcción de sus cartas y solo sea necesario hacer estructuras de datos para guardar las cartas en modo de maso , mano de jugador o estructura de línea de tiempo.

Se utilizó el patrón de Diseño Singleton para mantener una instancia donde se pueda llamar a la utilización de las cartas como base de datos y además para el guardado y mantenimiento de la información de los jugadores , en los Scripts:

- Database - PersistentManager

Controlador de las Escenas de Unity

La animación , transición entre elementos de la Interfaz de Usuario y el uso del Air Console
Air Console es una interfaz de programación de aplicación o API por sus siglas en inglés.
Esté Api permite tener conexión con los celulares que usen la aplicación o su dirección web
(www.airconsole.com) de Air Console de forma que pueda usarse como control del juego.
La conexión se hace por medio de un código que se otorga al momento de iniciar el juego.

En cada escena se ejecuta un controlador , un controlador de botón y un controlador de animación.

El controlador: activa el sistema de movimientos entre los botones y mantiene un índice para conocer donde se encuentra el jugador en la interfaz de manera que se pueda ejecutar el proceso que el usuario desee ejecutar.

Scripts: MenuButtonController, MenuButtonControllerCP , PlayerRegisterController , ProfileController , ChangeCardController.

Controlador de botón: controla la animación y los efectos de sonido al ser activado y seleccionado el botón en la interfaz

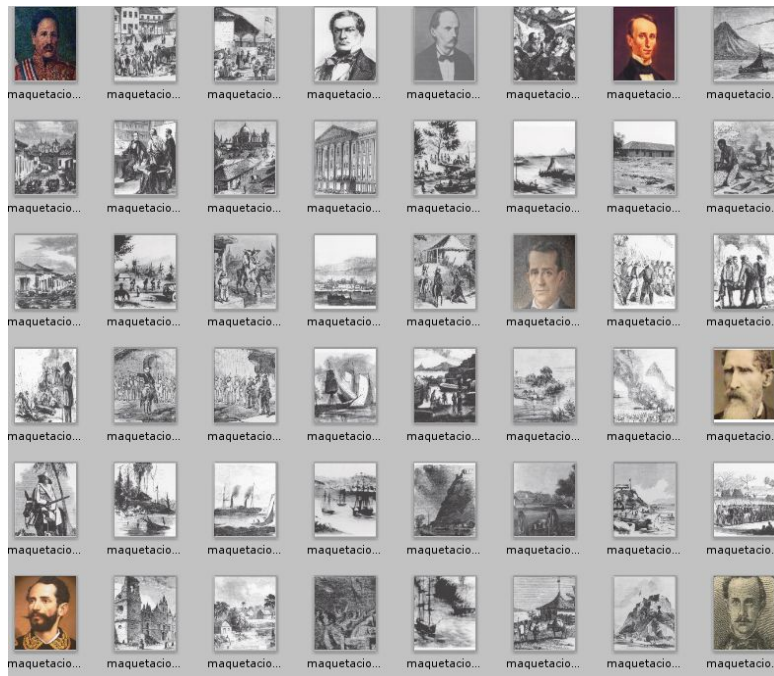
Scripts: MenuButton , MenuButtonCP, PlayerControllerBT, ProfileButtonController.

Controlador de animación: mantiene y ejecuta el efecto de sonido.

Scripts: AnimatorFunctions , AnimatorFunctionsCp ,PlayerListAnimation, ProfileAnimationController.

Diseño





Diseño de Grabados
Otorgados por el Cliente
Gustavo Godínez Vargas

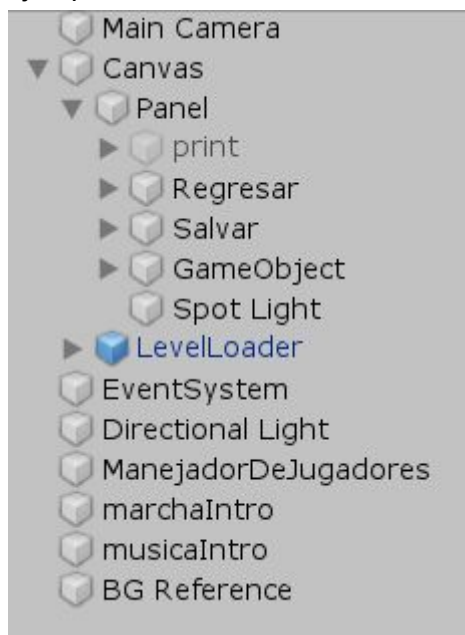
Para la creación de las Cartas
y de los Personajes.

El Diseño se basa en diferentes Escenas

Se Implementa el uso de un Canvas

- Dentro del Canvas se posicionan GameObjects (Estos pueden ser Botones, Imágenes, etiquetas)

Ejemplo:



Glosario:

Canvas: El área donde todos los elementos de UI deben estar. Es un GameObject con un componente Canvas en él.

GameObject: son los objetos fundamentales en Unity que representan personajes, elementos de la UI y el escenario.

ScriptableObject: contenedor de datos

Serializable: estructura de datos que pueden ser guardados por Unity.