

# UI Assignment

1.

## Unity UI vs UI Toolkit

Mientras el UGUI de unity se basa en componentes añadidos a GameObjects, permite que las UI se construyan de manera jerárquica como objetos dentro de las escenas, el UI Toolkit intenta emular más al desarrollo de UI en tecnologías web, más del estilo de HTML y XML. Las interfaces se generan principalmente con archivos de estilo y a través del código, permitiendo más personalización.

## UGUI

1. Ventajas: Es más simple de usar y aprender, sobretodo para los ya familiarizados con motores de videojuegos  
Está integrado al sistema de GameObjects así que es fácil de vincular entre ellos
2. Desventajas: En modelos complejos ese mismo sistema de GameObjects es una traba en el rendimiento  
No está diseñado para abordarse con flujos de trabajo Code Oriented

## UI Toolkit

1. Ventajas: Tiene mejor rendimiento en sistemas complejos  
Es ideal para usar herramientas personalizadas dentro del motor, y permite mayor variedad de creación
2. Desventajas: Efectivamente más difícil de aprender a usar, sobretodo para los desarrolladores que no tienen un trasfondo en tecnologías web  
Integrarlo al sistema de GameObjects de Unity para relacionarlo con los demás componentes es menos directos

2.

- a. El canvas es esencial porque actúa como contenedor y gestor de todos los elementos de UI. Es importante que los elementos UI que queramos ver visualizados en pantalla pertenezcan a la jerarquía del canvas.
  - i. Canvas Scaler: Ajusta el tamaño de la interfaz según la resolución target
  - ii. Canvas Renderer: Maneja el renderizado de los elementos, los convierte en gráficos visibles (a los elementos UI)
  - iii. Canvas Group: Gestiona la visibilidad e interactividad de todos los elementos dentro del Canvas. genera la opacidad en los elementos
- b. Image, Text, Button y Slider
  - i. Image: Sostiene archivos de imagen en la interfaz, se usa para íconos, fondos o botones.
  - ii. Text: Sostiene textos, se puede modificar la naturaleza de dicho texto y sus características, se utiliza para mostrar información como etiquetas o títulos o avisos.

- iii. Button: Actúa como elemento interactivo que responde a inputs del usuario, ejecuta acciones o cambia estados de otros elementos, se utiliza para generar interactividad responsiva entre el jugador y la interfaz
- iv. Slider: Es otro elemento interactivo que permite ajustar un valor dentro de un rango de manera visual, se suele usar para controlar volúmenes, brillo y otras configuraciones que varían entre un rango
- c. Layout groups
 

Son utilizados para ponerle un orden a los elementos que estén anidados en otro elemento de la UI, esto se realiza automáticamente y los organiza de manera ordenada según nuestras especificaciones que pueden ser ajustadas desde el editor. Es de alta utilidad para crear menús, barras de herramientas, inventarios, etc.
- d. Es el RectTransform, este elemento permite modificar los tamaños de los elementos, y los anclajes a sus parientes para determinar como se ajusta a estos. Esto es muy útil para que las interfaces se adapten correctamente a diferentes tipos de pantallas y resoluciones
- e. El Content Size Fitter, este elementos ajusta automáticamente el tamaño de los elementos y funciona en conjunto con los layout groups

### 3.

#### a.

1. VisualElement: Clase base para todos los elementos
2. Button: Permite recibir los clicks de los usuarios
3. Label: Son componentes de texto ligeros que enseñan información
4. ScrollView: Es un componente que permite deslizarse en ventanas de contenido que superen el contenedor donde están anidadas
5. TextField: Este es otro componente de texto, sin embargo este responde al usuario y recibe texto de su parte

#### b.

Visual Tree: Es la estructura jerárquica que existe y manejan los elementos del UI Toolkit, similar al DOM de HTML. Cada elemento es un nodo del visual tree, y la estructura define como se gestiona la interfaz. Su importancia radica en que claramente es el que sostiene y otorga toda la estructura del funcionamiento de la UI como componente, y de esta manera se encarga de definir las relaciones entre cada elemento, las transiciones e interacciones entre ellos, y procesa también los eventos donde entran inputs hacia los elementos. Es el encargado pues de mediar entre todos los sistemas.

#### c. UXML vs XML/HTML

La diferencia más directamente visible es que por supuesto, UXML es un archivo soportado dentro del motor de unity, gracias a sus propios elementos y estructura, por lo tanto sus referencias y las propiedades con las que cuenta, son nativas de Unity y por ende no aptas para desarrollo en tecnologías web más crudas.

#### d. USS vs CSS

De igual manera al punto anterior, es el archivo de estilos nativo del UI Toolkit, por lo tanto, una vez más, es algo que existe y funciona únicamente soportado dentro del motor de Unity, a raíz de sus propiedades y particularidades