

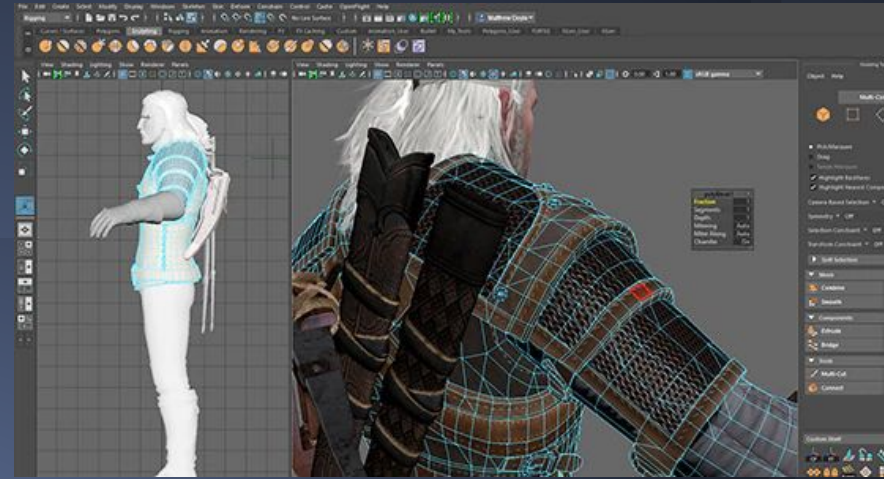


Desarrollo de Videojuegos

Herramientas para crear
jugabilidad avanzada
Interfaz de usuario

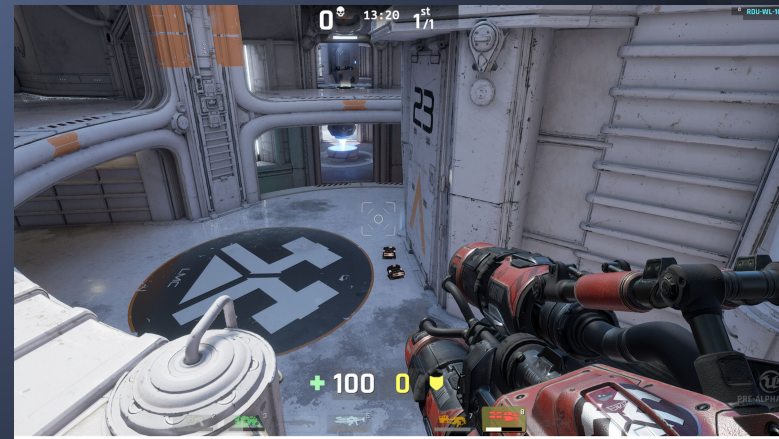
Motivación

- El diseño de un videojuego **NO termina cuando definimos su jugabilidad**
 - **Mecánica, dinámica y estética** detallada en un documento de diseño
- Es necesario **diseñar todo su contenido** (será *el grueso* de la producción)
 - Hay una parte **diegética** (**escenarios, objetos, personajes...** del mundo de ficción)
 - Otra parte es **extradiegética** (**música, voz en off, menús del juego...** que “sólo ve” el jugador)



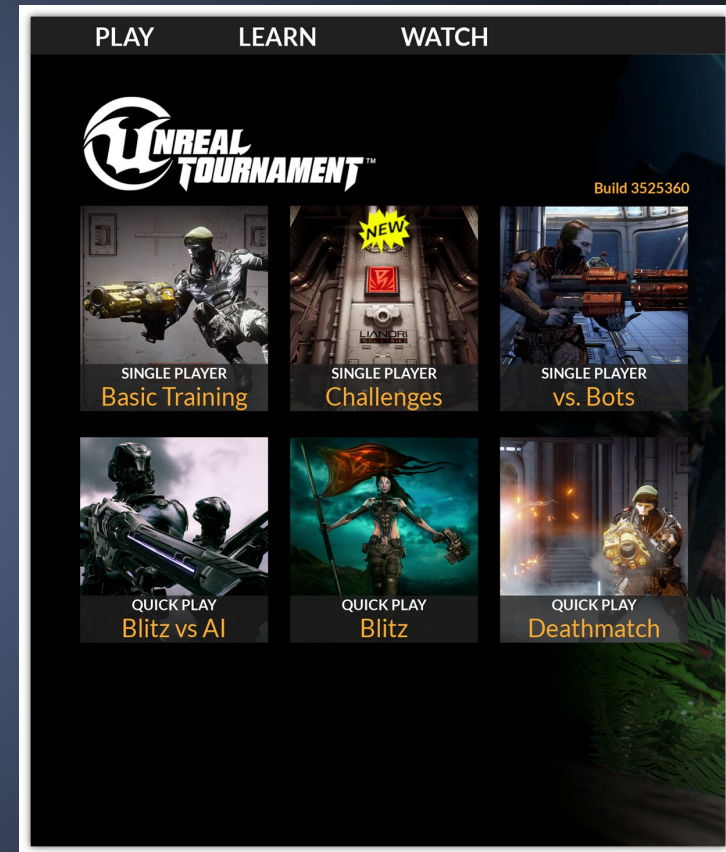
Interfaz de usuario

- Debemos definir previamente **qué información damos al jugador** y cuál no
 - Hazlo fácil y claro... y si puede ser, *con inmersión*
- En un juego “todo” es interfaz de usuario
 - La interfaz es *gráfica, y sonora, háptica...*
 - Puede tener elementos **diegéticos** (**cartelería, señales, diálogos...**), que son muy aconsejables
 - Suele tener elementos **extradiegéticos**
 - Sobre todo **HUDs y menús**, ¡nuestro foco ahora!



Tipos de interfaces

- **HUD**, para dar información importante
- **Menú**, para cambiar opciones de configuración o entre modos de juego
- **Información contextual**
 - Sobre los propios actores
- **Objeto interactivo** con modo de juego aparte (minijuego de puzles, diálogo emergente...)



Ejemplo de información contextual

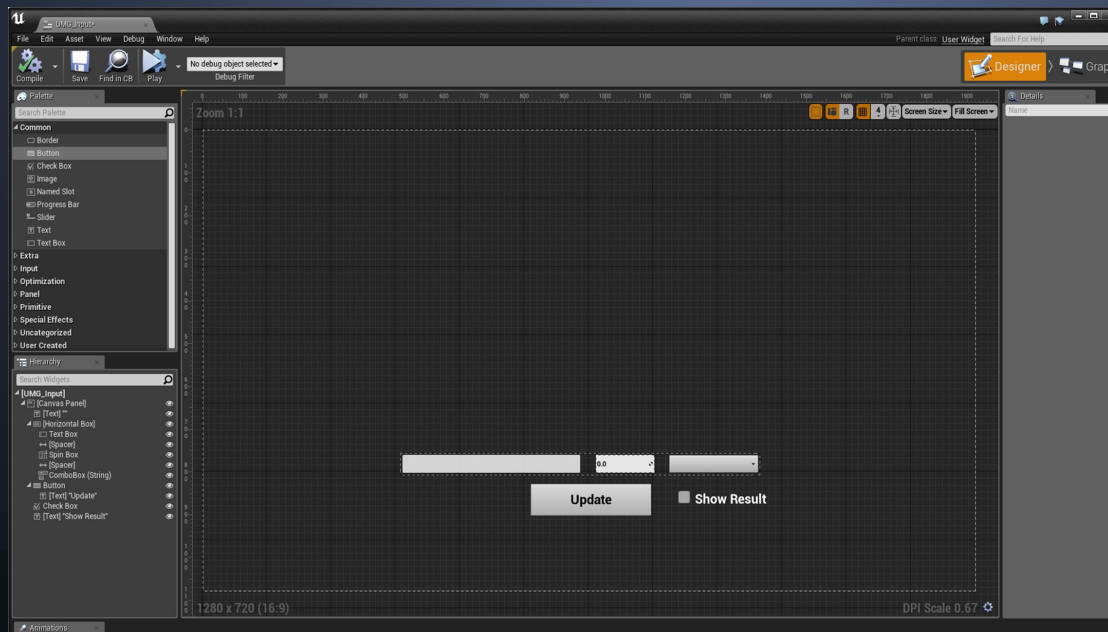
- Lo que va a la pantalla típicamente es como una *pegatina 2D* que se superpone a la vista
- Para algunos objetos del mundo conviene mostrar **información contextual**, tipo texto o imagen que “flota” sobre dichos objetos



UMG

- **Unreal Motion Graphics UI Designer**, la herramienta de autoría para interfaz visual
 - Podrán usarse sobre la pantalla como una pegatina 2D, o sobre una malla del mundo 3D

<https://docs.unrealengine.com/en-US/Engine/UMG/index.html>

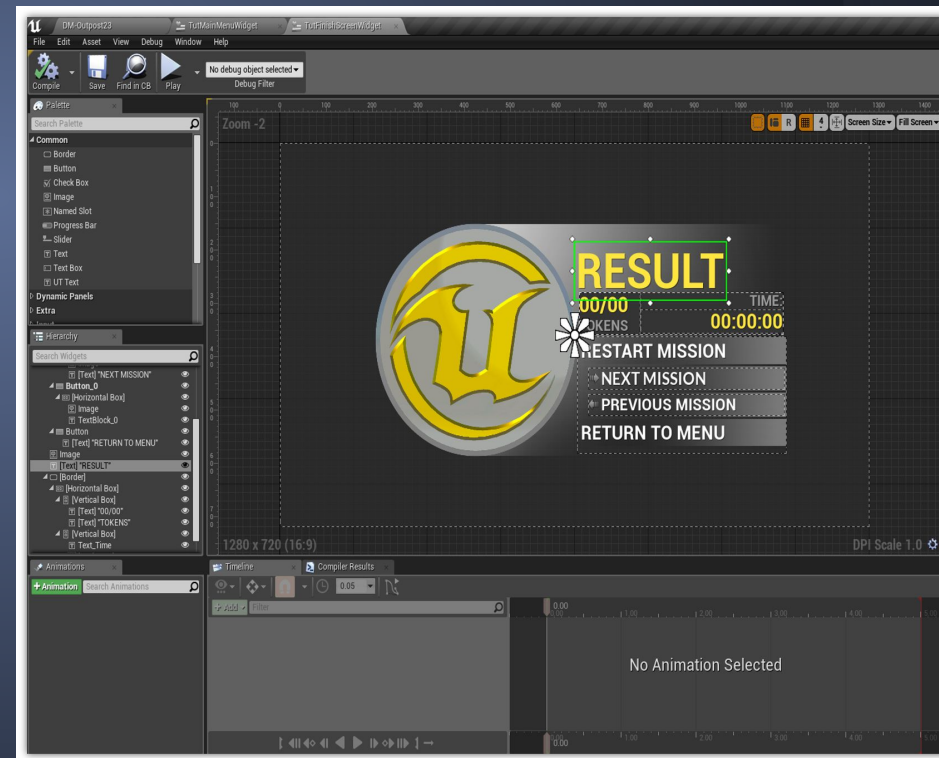


Widgets

- Estos artilugios son los **bloques** con los que se construyen interfaces en UMG
 - Algunos se pueden componer formando jerarquías (los que tienen ranuras/**slots**)
 - El padre controla el tamaño del hijo, donde se renderiza y si es o no visible... aunque el hijo también tiene propiedades que el padre puede considerar

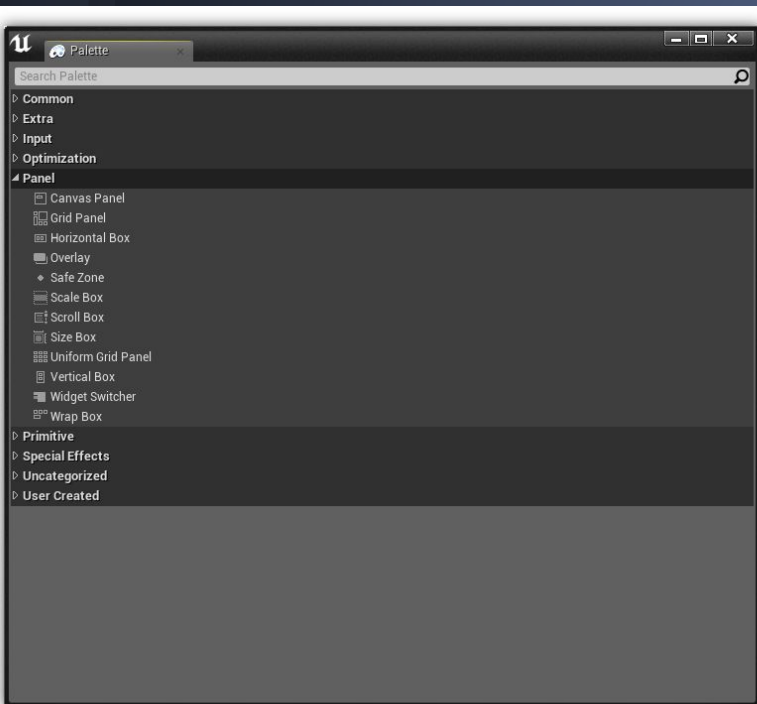
Tipos de Widgets

- Existen muchos pre-construidos, entre los más importantes:
 - **Paneles** (controlan posición y distribución de sus hijos)
 - **Widgets de muestrario de información**
 - **Widgets de entrada**
 - **Widgets de usuario**

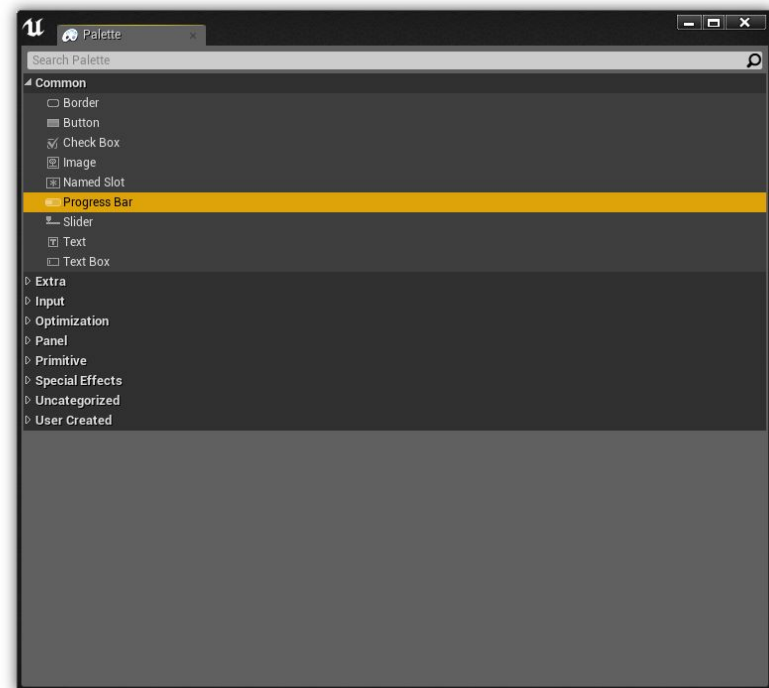


Paneles y artefactos de muestra

- Los de muestra son texto, imágenes, etc.
 - Los hay más complejos, como la barra de progreso



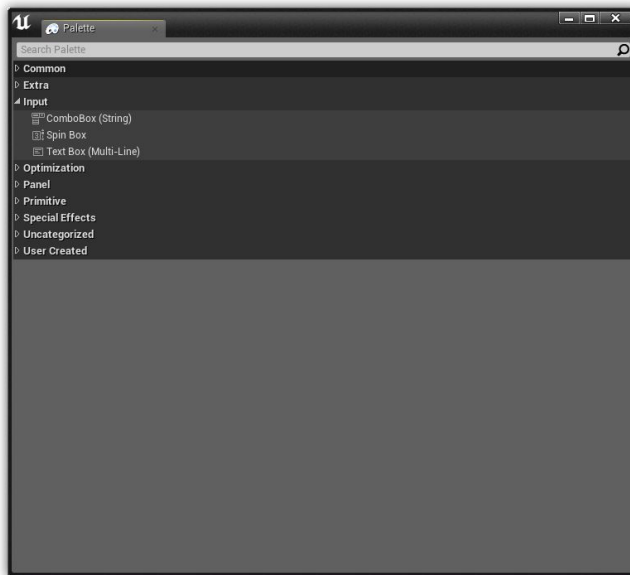
Panel Types



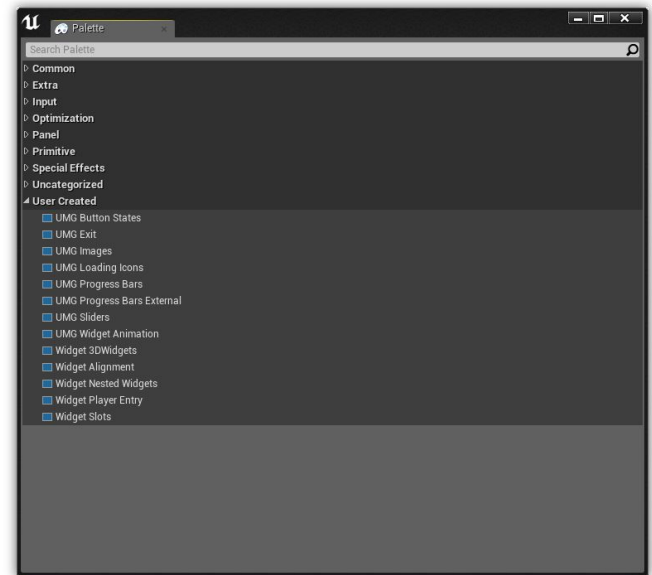
Common Display Widgets

Paneles y artefactos de muestra

- De entrada tenemos botones, cajas de texto.. y de usuario cualquier que creamos en la propia UMG



Extended Input Widgets

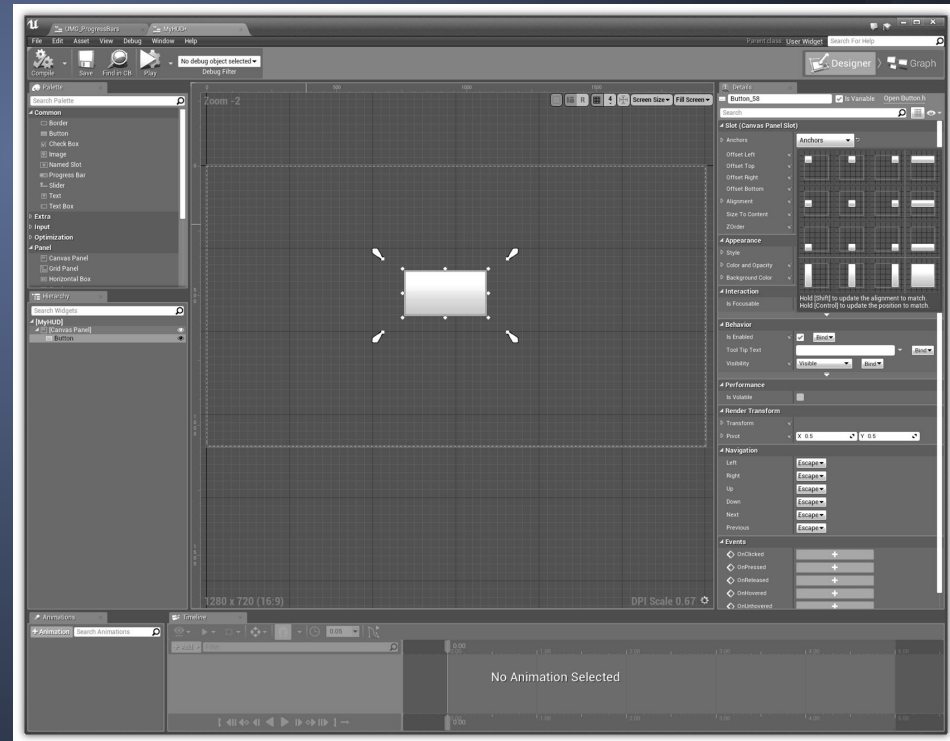


User Created Widgets

Anclas

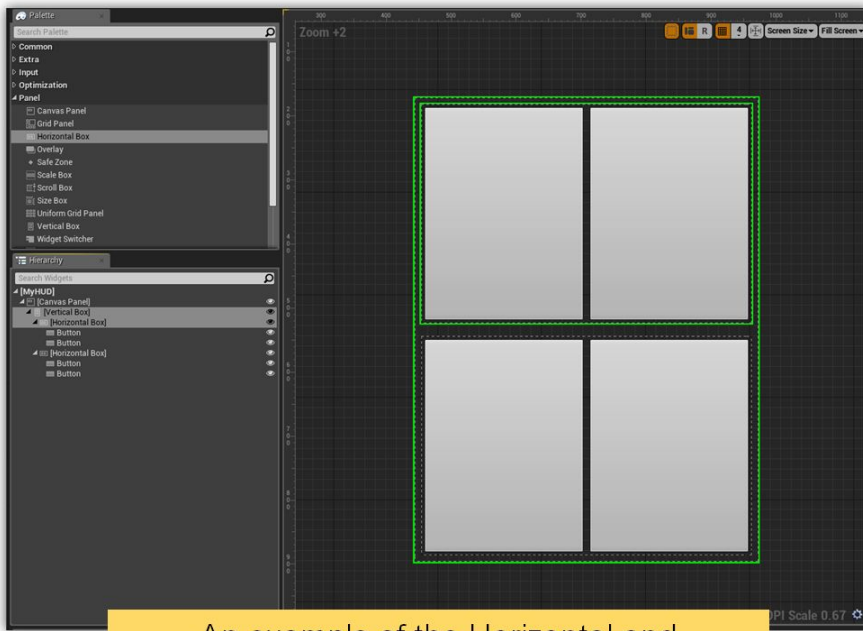
<https://docs.unrealengine.com/en-US/Engine/UMG/UserGuide/Anchors/index.html>

- Posición deseada en el panel principal del lienzo (Canvas)
 - (0,0) arriba-izquierda, (1,1) abajo-derecha
 - Importante evitar posicionamiento absoluto si es posible (mejor *relativo* y con *tantos por uno*)
 - Todo se puede asignar manualmente, incluyendo márgenes

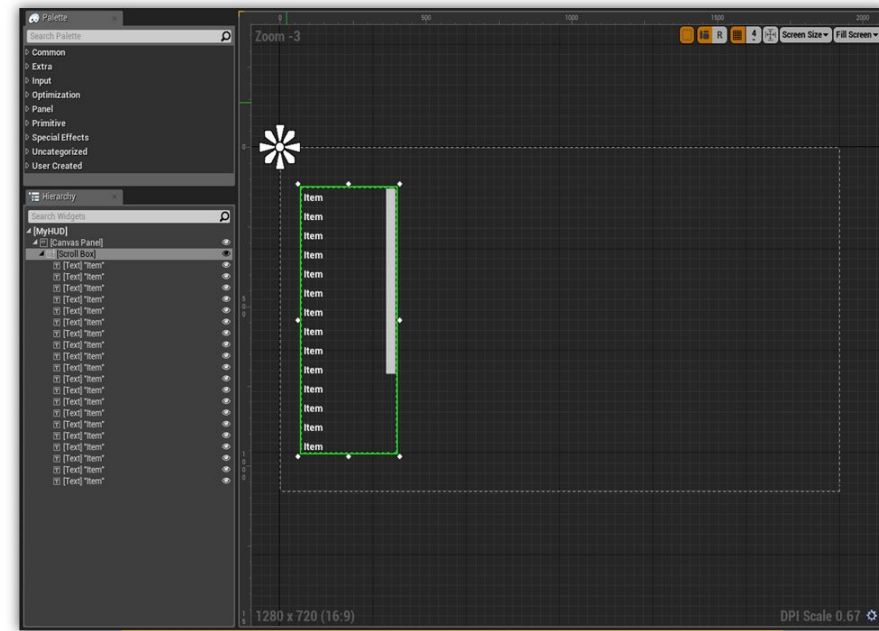


Caja horizontal/vertical o con scroll

- Distribuyen a sus hijos en un orden espacial
 - Horizontal/vertical... o mostrando scroll si los requisitos del hijo exceden los límites del padre



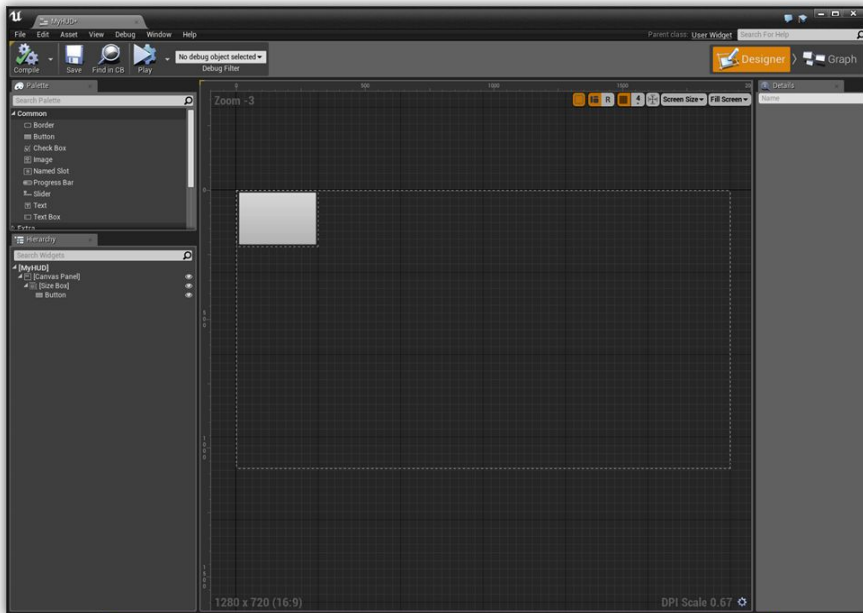
An example of the Horizontal and Vertical Box Widgets



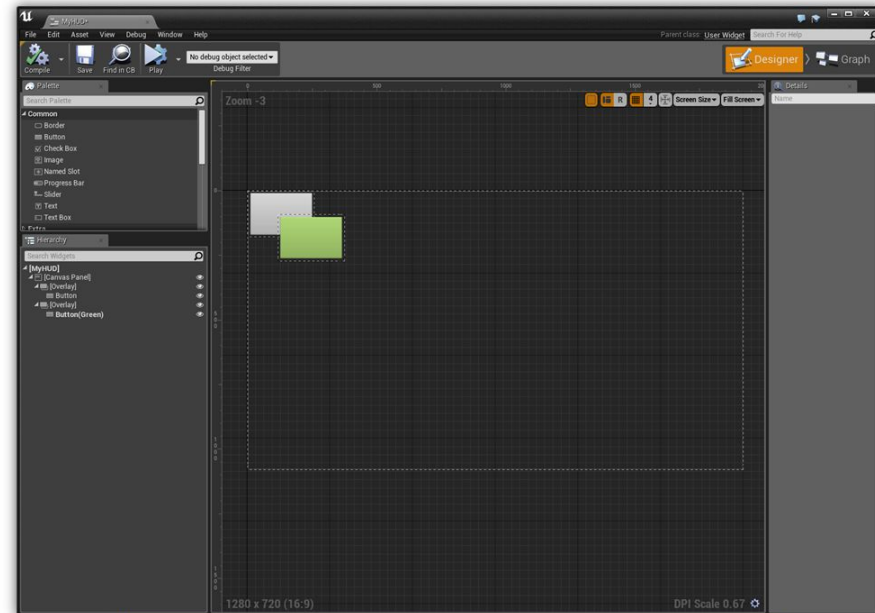
An example of the Scroll Box Widget

Caja de tamaño y sobreposición

- La caja de tamaño sólo tiene un hijo y le fuerza a un tamaño específico
- Overlay permite apilar hijos, uno sobre otro



An example of the Size Box Widget

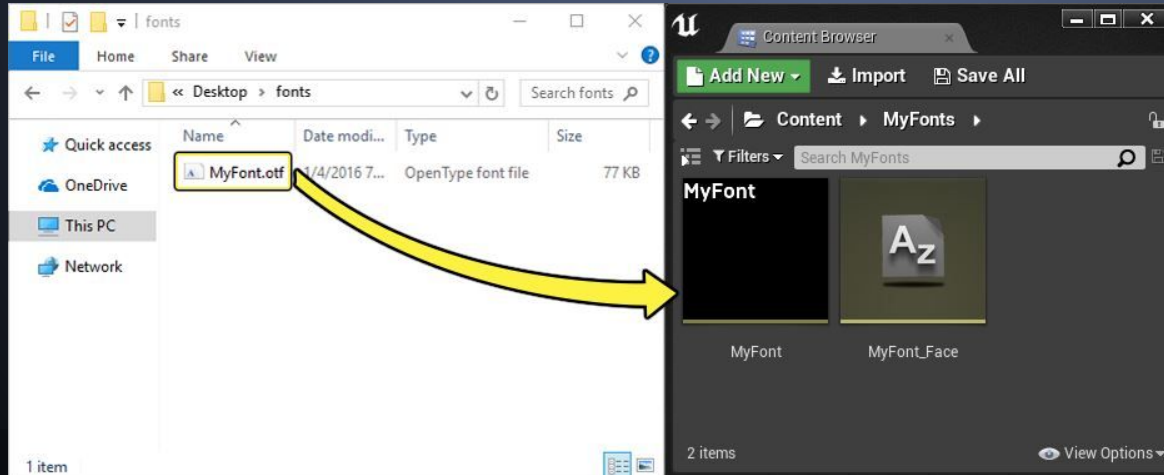
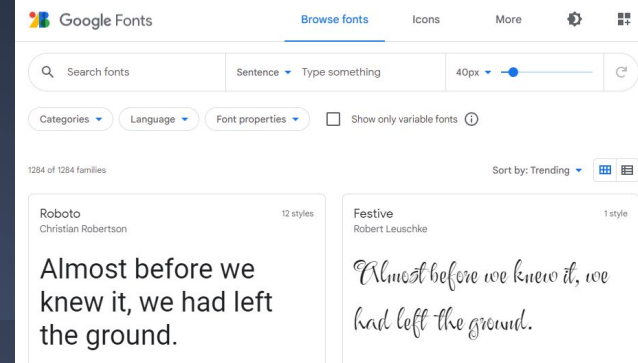


An example of the Overlay Widget

Fuentes

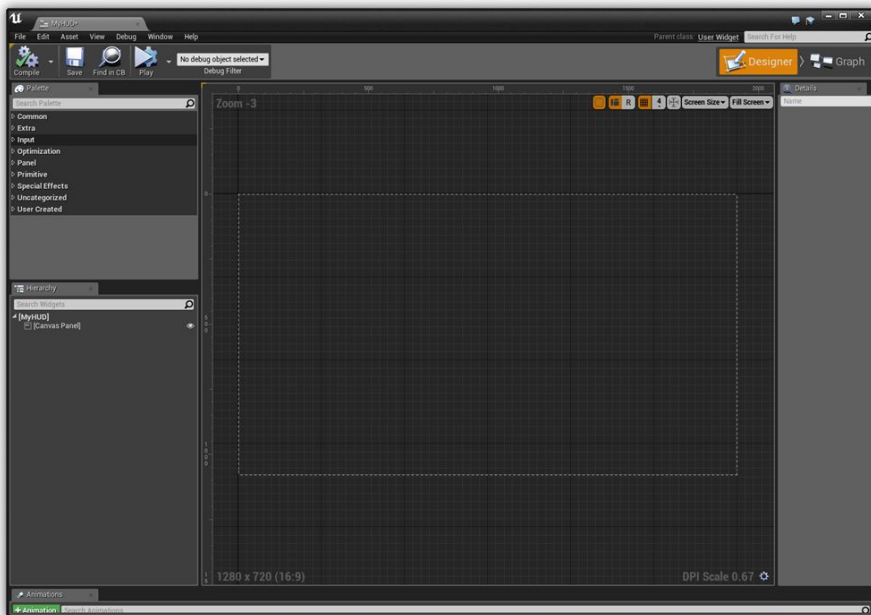
- A menudo son un recurso olvidado, pero se usan mucho y son *clave* para dar el **aspecto adecuado al texto** que aparezca en el juego
 - Aquí podemos **importar**, **asignar** y usar **fuentes**

<https://docs.unrealengine.com/en-US/Engine/UMG/UserGuide/Fonts/HowTo/index.html>

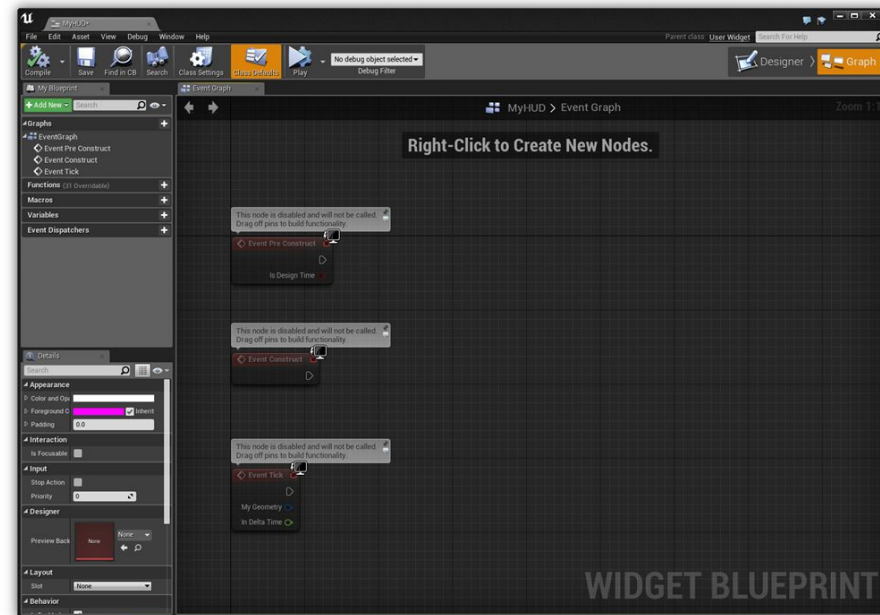


Dos pestañas en UMG

- En la de diseñador, editor WYSIWYG
- En la del grafo de eventos, se programa



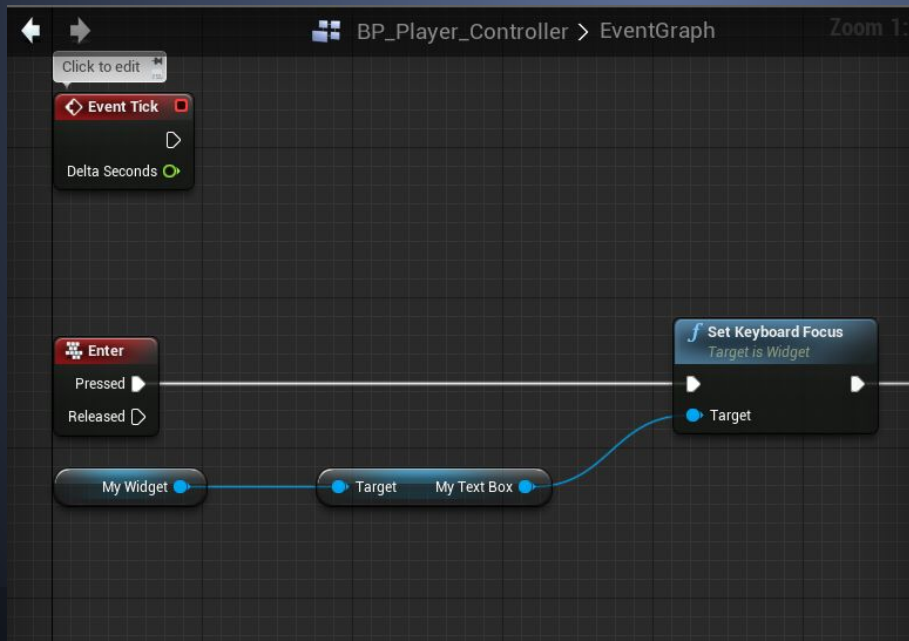
The interface for the UMG Designer



The interface for the UMG Event Graph

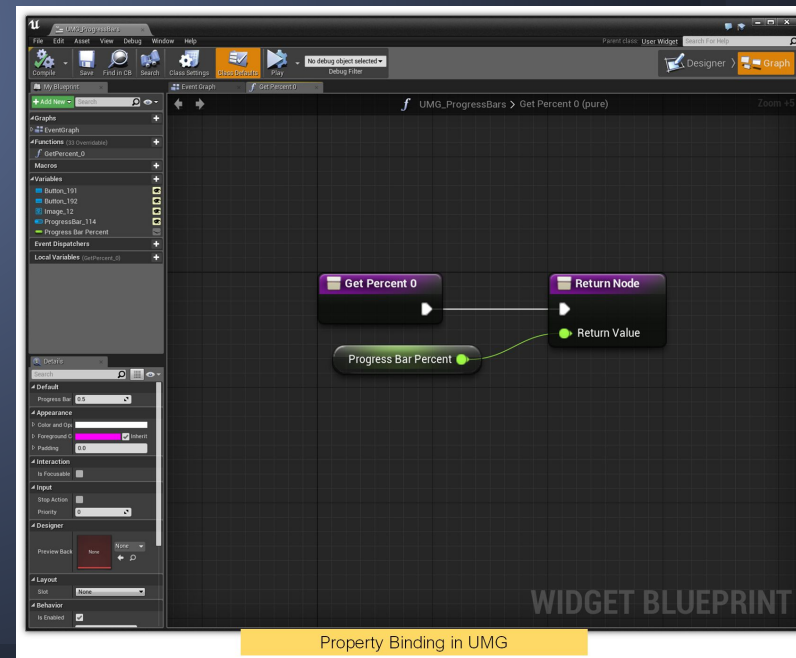
Conexión del interfaz con el juego

- El **foco de entrada** delimita qué componente (widget) exacto es el que recibirá la entrada del usuario, si lo hay
 - El ratón se usa para puntería, cursor en el menú...



Vinculación de propiedades

- Uno de los aspectos más útiles de UMG: vincular los widgets a funciones o variables de blueprints, de manera que si llaman a estas funciones o actualizan estas variables, se reflejarán estos cambios en pantalla



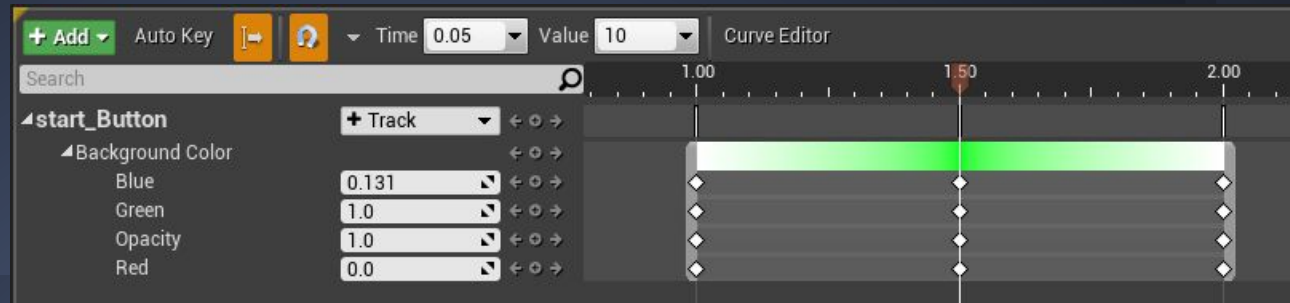
Property Binding in UMG

Participación

- ¿Qué *tipos de interfaces* pueden realizarse con UMG?
 - A. Menú, HUD, info contextual y objeto interactivo
 - B. HUD, Menú y objetos contextuales
 - C. HUD, Menú, objeto contextual e interactivo
 - D. Diegéticos, extradiegéticos o de UMG

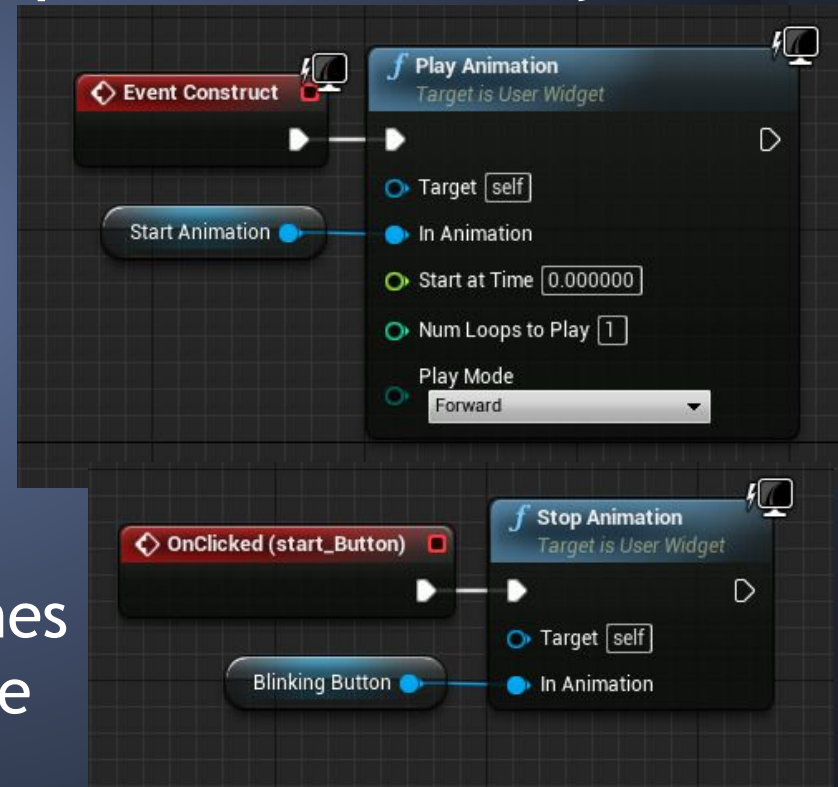


Animación



- Abajo en el Widget Blueprint Editor, hay posibilidad de **crear y controlar animaciones**

- Con pistas de animación
- Con una línea temporal donde se pueden marcar fotogramas clave, para que el sistema interpole
- Las imágenes y animaciones son **2D** pero pueden usarse materiales, renderizar escenas 3D tomadas del propio juego, etc.



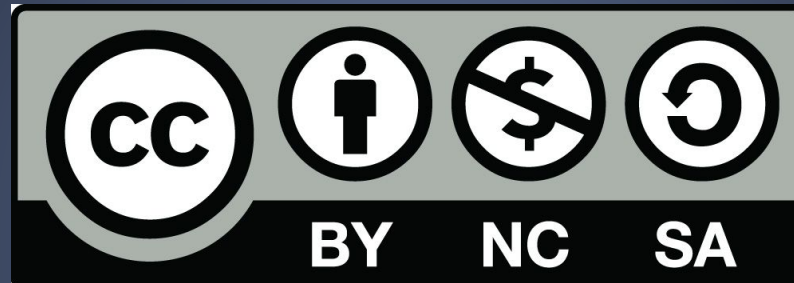
Buenas prácticas de diseño

- Aunque se diseña con una determinada **escala y resolución** ideal, conviene probar la interfaz en varias configuraciones
- Piensa en que tus recursos gráficos podrán sufrir **reescalado**
- Diseña *widgets básicos* **para reutilizarlos** y componer otro más sofisticados



<https://docs.unrealengine.com/en-US/Engine/UMG/UserGuide/BestPractices/index.html>

Críticas, dudas, sugerencias...



** Licencia sólo aplicable al texto original de estas diapositivas*

Federico Peinado (2019-2026)

www.federicopeinado.es