



Clase 04



- C# Buenas Prácticas
- Unity General
- Performance (Básico)
- Presentación de un Proyecto
- Comentarios Generales
- AI básica



UI

- Todos los scripts que manejen UI que empiecen con prefijo "UI" (ejemplo: UIManager, UIPlayerHud)
- Todo script relacionado con UI va en una carpeta UI
- Hacer un script general para la UI, no tener un componente por cada Texto o Botón a menos que dicho objeto tenga cierta complejidad.
- Los scripts que manejan la UI van en el Canvas o alguno de sus hijos. Supongamos que queremos pasar la UI a otro juego, tengo que pasar los otros objetos también...



C# Buenas Prácticas

- Nombres descriptivos en scripts
- Comentarios en código.
- Programar para el otro. O el YO del futuro!
- No Overengineering, menos es más. Cantidad de líneas de código, cantidad de archivos.
- No usar variables estáticas excepto en casos muy particulares.
- Coherencia de código. Si ponen private todas las private con su signature private, si no ponen nada, todas sin nada.
No mezclar variables públicas con privadas.



C# Buenas Prácticas (cont)

- Variables que no se usan en el código, que no estén en el código (Unity tira un warning de variables no usadas).
- No tener métodos vacíos, por más que al generar el script Unity genere métodos como Start() y Update() si los métodos no tienen código, borrarlos.
- Que los componentes/scripts matcheen con el Gameobject asignado. Mal ejemplo: Un script de Loader en la Cámara. Si es necesario, crear un gameobject vacío con el nombre del script.



C# Buenas Prácticas (cont)

- `if (variableBooleana == true)` NO!

Usar `if (variableBooleana)` o `if (!variableBooleana)`

Recordar: menos es más.

- Nombres de clases/scripts/componentes. Si es un pino, que la clase se llame "Pino" y no "Pinos"
- Pdf: CSharpCodingStandards



General Unity

- Resoluciones de pantalla.
- Nombres descriptivos en GameObjects.
- Break Prefab Instance. (Menu->GameObject->Break Prefab Instance)
- Escenas selfplayable.



General Unity (cont)

Project Tab

- Find References in Scene.
- Usar estructura de carpetas similar a la mostrada anteriormente.

Hierarchy Tab

- Usar Empty gameobjects como carpetas.



Performance (Básico)

- No llamar funciones como `GameObject.FindGameObjectsWithTag()` o `GetComponent()` en el `Update()` (todos los frames). Son lentas, generan basura en memoria. Mejor cachearlas.
- Concatenar (+) strings es lento, no hacerlo en el update si lo que voy a mostrar no se modificó.
- Profiler.



Presentación de un Proyecto

Al compartir un proyecto (entregar un tp)

- Borrar las carpetas Library/Temp en el proyecto.
- Comentar todos los Debug.log() o directamente borrarlos.

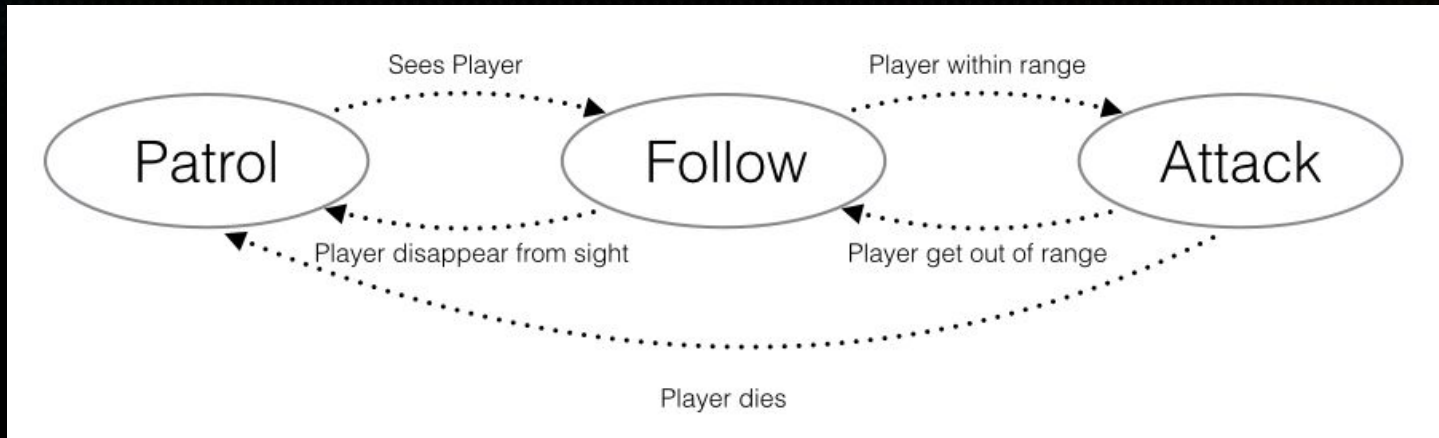
Arte de programador

- Proporciones, espacio en pantalla.
- Color
 - <https://coolors.co/>
 - <http://www.colourlovers.com/palettes>
 - <http://colorpalettes.net/>



AI Básica

- FSM (Finite State Machine/Máquina de estados Finitos)





Tarea

- Investigar Doxygen.

<http://www.stack.nl/~dimitri/doxygen/>

- Investigar AnimationCurve por código.

<https://docs.unity3d.com/ScriptReference/AnimationCurve.html>