Image Campus: UE4

Proyecto Práctico

Objetivo

Programar un pequeño juego, con una mecánica simple, utilizando Unreal Engine 4 y C++.

El Juego

- El juego podrá utilizar como base alguno de los template de gameplay de UE4 (First Person, Third Person, Top-down, Side scroller, etc)**, sin Starter Content**.
- El juego debe contar de un diseño propio por equipo, contando con ciertas complejidades base de gameplay que incluyan mínimamente algunos de los contenidos vistos en la cátedra.
- Deberá contar con un HUD que indique alguno de los elementos de gameplay.
- Todo el gameplay debe ser configurable en el editor

El mismo deberá contar con:

- Menu principal
- Condición de victoria
- Condición de derrota
- Flujo de rejugabilidad

Entrega

Para la entrega se deberán tener en cuenta lo siguiente:

- Utilizar UE4 4.24.3.
- Entregar sólo las carpetas que son necesarias para correr el juego desde el editor. No entregar ejecutables, ni archivos intermedios de compilación. Incluir Config, Content, Source y el archivo de proyecto .uproject.
- Toda la mecánica tiene que ser programada en C++. Minimizar el uso de blueprints para lógica de gameplay, usarlos más para configuración, UI o implementación de efectos.
- Enviar todo en un zip llamado ProyectoEquipo[número del equipo].zip.

Evaluación

Para la evaluación se tomarán en cuenta:

- La estructura y prolijidad del código.
- El correcto funcionamiento de los algoritmos programados.
- **Estándares de código mínimos**. https://docs.unrealengine.com/latest/INT/Programming/Development/CodingStandard/
- **Uso correcto del motor**.
- La sensación de jugabilidad.

Temas

- Gameplay Architecture: Actors, Objects, Delegates, Timers, Data Containers
- UFunctions, UProperties, UStructs, Meta Specifiers, Interfaces
- Clases Base de UE4: GameMode, GameState, PlayerController, PlayerState, GameInstance, PlayerState, GameInstance, Pawn
- Spawning Actors, Bindeo de Input
- Actor Components, Blueprint Classes, Asset References (TSubclassOf, TSoftObjectPtr), Carga Sync y Async
- UMG
- Data Driven Gameplay Assets (DataTables, DataAssets)
- Collision Presets, Channels, RayTracing