## EC para unity potente

Burst compiler y un tio de gdc no se que anton

## Recordatorio

Tenemos módulos (de física, de tal.. etc). Ahora vamos a hacer un modulo. Hay uno de entidades que se encarga de actualizar las entidades. Las entidades por si no tienen update (lo tienen los componentes).

Las entidades son tienen nada (hay que **añadirle** **transform** y **nombre** si los queremos).

Y cuidado con la **jerarquía**.

Hay un modulo boot para cargar las escenas del juego.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Prefab y escena es lo mismo. Quiero crear una entidad.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza baja

Ahora sale nuestra entidad. (tocamos el boot.json y creamos el my\_entity.json en )

Captura de pantalla de computadora

Descripción generada automáticamente

Creamos una clase .h y cpp

El .h para los headers, lo hacemos por si alguien quiere incluirlo. Si no necesitamos que yo existo me salto el .h

Texto

Descripción generada automáticamente

Tenemos un componente nuevo (hemos modificado el my\_entity.json y añadido un component).

Una pantalla de una computadora

Descripción generada automáticamente con confianza media

Hacemos update

Una captura de pantalla de una computadora

Descripción generada automáticamente

Ahora falta llamarlo

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

A las 19:03 ha hecho algo q no he visto de macros creo

Crear nuevas entidades. Si disparo. Queremos un template/prefab que es una entidad con sus componentes que si le digo que cree una entidad lo haga. Otra forma es intentar que uno hable con otras entidades y preguntarle si son de un u otro

Un dibujo de un pizarrón blanco

Descripción generada automáticamente con confianza baja

Le hemos dicho que apunte a la pos roja.

Delta = target – pos (dirección hacia el punto rojo).

Texto

Descripción generada automáticamente

Para pintar mas en el comp\_orient\_to\_cpp

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Para probar

**Quaternion**: 4 valores x,y,z,w q representan una rotación de tantos grados sobre uneje.

W es el cos(ang/2).

Q = AXIS \* sin(ang/2)

Q1 \* Q2 es distinto de Q2 \* Q1

La multiplicación es mas rápida. Lo q es mas costoso es un punto \* Quaternion.

Es muy fácil de construir. Se podría construir la matriz matemática para rotar pero no hace falta con los quaterniones.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Dado un fwd