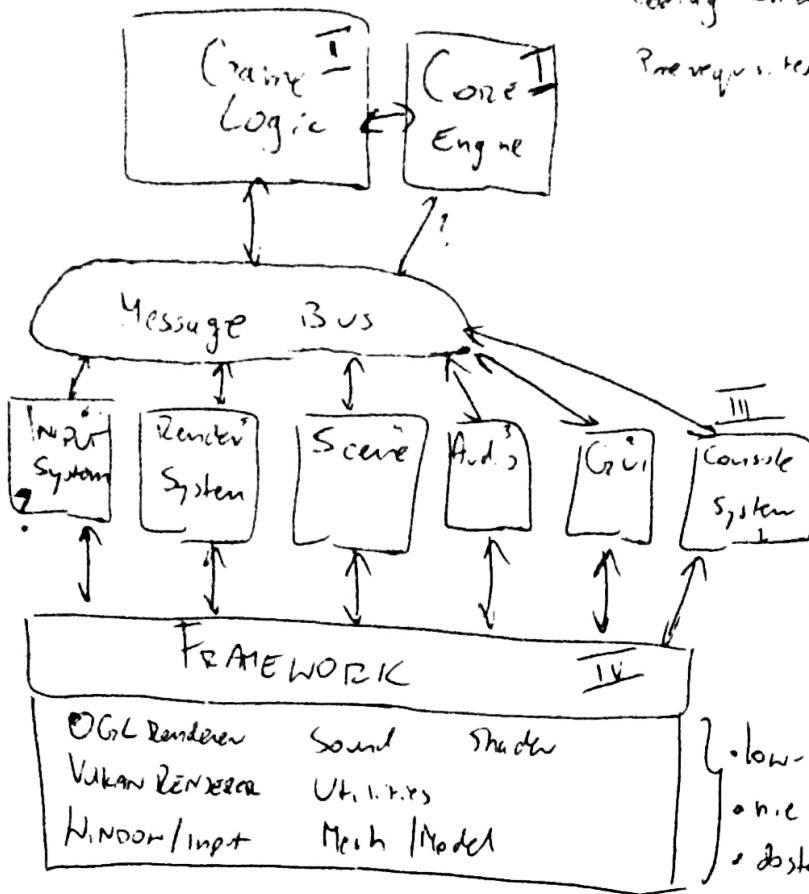


Coding order: I, II, etc

Prerequisites: entity x

Questions

- Scene Graph - kładzie GO ma domyślny comp. Transform.



Np wysłanie we we gdzie i po co.

Component Transform
Position
Rotation
Scale
Color parent.

low-level API
• nie obchodzi, czy jest, gdzie, i po co.
• dostarczając funkcje sub-logic coś np render shadow map etc.

Game Logic

- Base Game - funkcje update? na pewno input + GUI, vector + Game Objects.
- Game Object - vector + Components? parent ID? Parent Component?
- Core Engine - component Transform.
- run all systems
- get BaseGame
- message Bus (events based on entity x).
- allow to add new Systems. (extensibility feature).

Unikanie tworzenia obiektów przez new/malloc.

- Factory - do tworzenia GO.
- Memory Management System.

Camera - components
FreeCam etc.

Systemy nie muszą być napisane w opisie ECS.

Render System - współdzieli miejsce z frameworkiem: render w celu renderowania obiektów.

Scene: Prostownie hierarchia obiektów - Scene Graph.

interakcje tylko przez transformację, używaj matryc. GLOBALNA (MODEL/NOBLE)

Input: potrzebny? czy potrzebny? - Base Input System? - musimy ten system - oryginalnie do sprawdzania czy dany obiekt jest w stanie być interaktywny.

Audio: czeka na event by odegrać dźwięk?

GUI: renderuje GUI w odp kolejności. + wysyła eventy. - Base GUI System? przetwarza GUI Scene? - do celu potrzebny do Base Game.

Console System: czeka na eventy i wysyła na ekran (kursor).

• Animation System

↳ certain animat' - interesuje se tylko LOCAL ^{→ co precise pred} Transform. Model.
(~~Model~~ ~~Matrix~~) Anim Matrix?

• Memory Management System

• Factory class - to create objects via this class eg. new Game Object,
Camera Object etc.