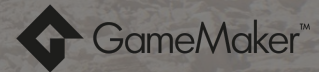


LESSON 1

GAME ENGINES



NICOLAS SERF

SERF.NICOLAS@GMAIL.COM



WHAT IS A GAME ENGINE ?



WHAT IS A GAME ENGINE ?

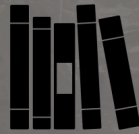


TOOLBOX

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

LIBRARIES



REUSABILITY



ACCESSIBILITY



OPEN / CLOSE



PLUGINS



CORE



DEFINITION

LIBRARIES CONCEPT IS NOT ONLY TIED TO GAME DEVELOPMENT AND GAME ENGINE, BUT A GENERAL CONCEPT OF DOING REUSABLE PIECE OF CODE ENCAPSULATED INTO AN EXTERNAL LIBRARY NOT TIED TO A SPECIFIC PROJECT

EXAMPLES

PHYSICS, MATHEMATICS, RENDERING, ETC...

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

REUSABILITY



LIBRARIES



ACCESSIBILITY



OPEN / CLOSE



PLUGINS



CORE



DEFINITION

REUSABILITY IS KIND OF THE KEY DEFINITION OF A GAME ENGINE, EVERY FEATURES MADE BY GAME ENGINE DEVELOPERS ARE FOCUSED ON REUSABILITY WITHOUT ANY PROJECT DEPENDENCY

EXAMPLES

PATHFINDING, RENDERING PIPELINE, OCCLUSION CULLING, ETC...

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

ACCESSIBILITY



LIBRARIES



REUSABILITY



OPEN / CLOSE



PLUGINS



CORE



DEFINITION

AN IMPORTANT PART OF A GAME ENGINE IS THE **ACCESSIBILITY** OF THE ENGINE. IT DIFFERS FROM ENGINE TO ENGINE, BUT IT IS IMPORTANT TO BE USABLE BY A DEVELOPER BUT ALSO NON-DEVELOPERS FOR SOME PARTS.

LANGUAGES

THIS IS NOT THE ONLY FACTOR DETERMINING **ACCESSIBILITY**, BUT CODING LANGUAGE USED IN ENGINE ARE IMPORTANT IN THAT SUBJECT. EACH GAME ENGINE IS ASSOCIATED WITH 1+ CODING LANGUAGES.

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

OPEN / CLOSE



LIBRARIES



REUSABILITY



ACCESSIBILITY



PLUGINS



CORE



DEFINITION

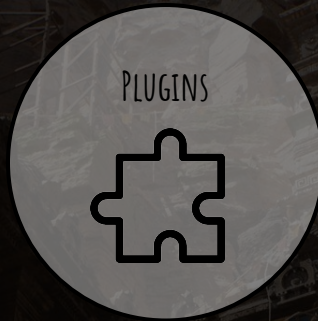
THIS IS DETERMINING FOR A LOT OF COMPANIES. SOME GAME ENGINES HAVE SOURCE CODE OPEN, SOME OTHERS DON'T. HAVING ACCESS TO SOURCE CODE, AND POSSIBILITY TO BUILD AND MODIFY THE ENGINE CAN BE ESSENTIAL

EXAMPLES

UNREAL ENGINE IS NOT OPEN SOURCE, BUT SOURCE CODE IS ACCESSIBLE, UNITY IS CLOSED, GODOT IS OPEN SOURCE, ETC...

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

PLUGINS CAN HAVE DIFFERENT NAME BASED ON THE ENGINE, BUT THE CONCEPT IS SIMILAR, IT IS A BIT LIKE AN EVOLVED LIBRARY, WHICH CAN CONTAINS CODE, ASSETS, EXTERNAL DEPENDENCIES, ETC... AND ARE ALWAYS OPTIONAL

PROVENANCE

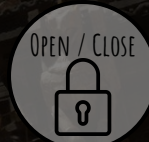
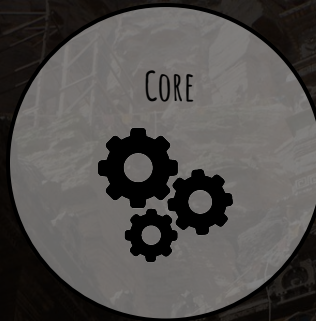
PLUGINS COMES EITHER FROM ENGINE DEVELOPERS, WHICH SHOULD ENSURE TO BE FULLY INTEGRATED AND POLISHED INTO THE ENGINE, OR FROM MARKETPLACE OR PRIVATE GITHUB, MADE BY THE COMMUNITY, FREE OR PROFITABLE

EXAMPLES

UNREAL ENGINE HAVE PLUGINS, UNITY HAVE PACKAGES, ETC...

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

EVERY ENGINE HAS A CORE FOUNDATIONS BASE CLASSES WHICH GIVE THE BASE HIERARCHY TO ACCESS THE MAIN FRAMEWORK OF THE ENGINE, LIKE REPLICATION, REFLECTION, ETC...

SET IN STONE

THE CORE FOUNDATION OF EVERY ENGINE IS KIND OF SET IN STONE. BY THAT, I MEAN THAT IF YOU DON'T LIKE AND DON'T WANT TO WORKS WITH IT, CHANGE ENGINE INSTEAD OF REWRITING EVERYTHING OR TRYING TO AVOID IT.

EXAMPLES

UNITY OFFERS A REALLY LIGHT BASE HIERARCHY, WHILE UNREAL'S GAME FRAMEWORK IS MASSIVE, EXPLAINING THE DIFFICULTY TO LEARN THE ENGINE



WHAT IS A GAME ENGINE ?

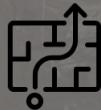


AI

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

PATHFINDING



BEHAVIORS



PERCEPTION



ENVIRONMENT



A LOT TO DO



DEFINITION

PATHFINDING IS A KEY ELEMENT FOR EVERY GAMES WITH AIs. IT ALLOWS AGENTS TO NAVIGATE IN THE LEVEL BY LOOKING FOR A POSSIBLE AND OPTIMAL PATH

IMPLEMENTATION

THE IMPLEMENTATION MAY DIFFER FROM ENGINE TO ENGINE, BUT GENERALLY SPEAKING, THEY ARE USING CUSTOM A* WITH A TRIANGULATION OF THE LEVEL BASED ON COLLISIONS

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

BEHAVIOR



PATHFINDING



PERCEPTION



ENVIRONMENT



A LOT TO DO



DEFINITION

EVERY AGENT HAS A **BEHAVIOR** IN A GAME, FROM SIMPLE IDLE **BEHAVIOR**, NOT INTERACTING AT ALL WITH THE WORLD, TO COMPLEX BEHAVIOR WITH MULTIPLE INTERACTION LAYER. BEHAVIORS DRIVES THE AGENT ACTIONS

TECHNIQUES

THERE IS MANY TECHNIQUES KNOWN TO CREATE BEHAVIORS LIKE STATE TREE, BEHAVIOR TREE, FSM, ETC... BUT SOME ENGINE ARE LIGHTER ON THE SUBJECT, AND DO NOT OFFER THAN MUCH

EXAMPLE

UNREAL OFFERS VARIOUS TOOLS TO CREATE BEHAVIORS, FROM **BEHAVIOR TREE** TO **BLACKBOARD**, WHILE UNITY NOT THAT MUCH, EXCEPT **ANIMATION** AND **ANIMATION BEHAVIOR** WHICH CAN BE HIJACK TO CREATE FSM OR BT.

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

WHEN YOU WANT TO HAVE AGENT **INTERACT** WITH THE WORLD, AND MORE SPECIFICALLY WITH OTHER AGENT OR PLAYERS, THEY NEED TO BE AWARE OF THE WORLD, THROUGH SENSES AND **PERCEPTION**

EXAMPLES

YOU COULD THINK OF VARIOUS TECHNIQUES TO PERCEIVE THE WORLD, THE 2 MOST OBVIOUS BEING **SIGHT** AND **HEARING**. THEY CAN BE SIMULATED WITH **RAYCAST** / **SHAPECAST** OR A BIT OF LOGIC

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

ENVIRONMENT



PATHFINDING



BEHAVIORS



PERCEPTION



A LOT TO DO



DEFINITION

ENVIRONMENT IS ANOTHER IMPORTANT FACTOR FOR AN AGENT. IF YOU WANT YOUR AIs TO BE AWARE OF THE ENVIRONMENT LIKE A PLACE TO HIDE, A BETTER SPOT TO FIRE, ETC... THIS IS TIED TO ENVIRONMENT AWARENESS

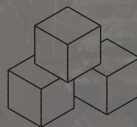
EXAMPLES

ONE OF THE MOST KNOWN TECHNIQUE IS EQS WHICH STAND FOR "ENVIRONMENT QUERY SYSTEM". SOME ENGINES LIKE UNREAL OFFERS TO SUPPORT OUT OF THE BOX FROM THE ENGINE.

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

A LOT TO DO



PATHFINDING



BEHAVIORS



PERCEPTION



ENVIRONMENT



DEFINITION

GLOBALLY SPEAKING, AIS SUBJECT ARE KIND OF TIED TO **GAMEPLAY MECHANICS** AND **PROJECT SPECIFICITIES**, SO THERE IS ALWAYS A LOT TO DO ON A PROJECT REGARDING AIS IF YOU WANT SMART INTERACTION

EXAMPLES

FACTION SYSTEM, WORLD BEHAVIOR, DEVELOPING COMPLEXE LOGIC OF TRANSITION AND INTERACTION FROM BEHAVIOR STATE, ETC...



WHAT IS A GAME ENGINE ?



INPUT / OUTPUT

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

AUDIO



CONTROLLERS



M&K



FILES



HARDWARE



DEFINITION

AUDIO IS BY ESSENCE, EVEN MORE NOW, A CRITICAL PART OF ANY GAMES WHICH BRING AS MUCH AS ARTISTS AND GAMEPLAY. IT IS RESPONSIBLE FOR BRINGING TO LIFE VISUAL ELEMENTS, FEEDBACKS, AND MORE

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

CONTROLLERS



AUDIO



M&K



FILES



HARDWARE



DEFINITION

NOWADAYS, A VAST MAJORITY OF GAMES SUPPORT **CONTROLLERS**, BECAUSE THEY'LL EITHER BE **MULTI-PLATFORM** GAMES, OR BECAUSE A LOT OF PLAYERS WANTS TO PLAY WITH **CONTROLLERS**. SUPPORT THEM IS NEARLY NOT AN OPTION.

IMPLICATION

SUPPORTING CONTROLLERS HAS A LOT OF **IMPLICATION** WHICH CAN BE SEEN IN GAMES. THERE IS WAY LESS KEYS AND WAY TO **INTERACT** WITH THE WORLD, KIND OF **SIMPLIFYING** GAMEPLAY

SUPPORT

EVEN IF IT SIMPLIFY THE GAMEPLAY, **SUPPORTING** CONTROLLERS MAY BE **CHALLENGING**. BUT NOWADAYS, ENGINES TEND TO UNDERSTAND THE **CHALLENGE**, AND OFFERS **GREAT TOOLS** AROUND THAT

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

A BIT LIKE CONTROLLER, IF YOU TARGET A GAME FOR COMPUTER, YOU'LL NEED TO SUPPORT **MOUSE & KEYBOARD**.

IMPLICATION

MOUSE & KEYBOARD IS ACTUALLY EVEN MORE COMPLICATED TO MANAGE THAN CONTROLLER BECAUSE YOU NEED TO TAKE INTO ACCOUNT **KEYBOARD LOCALIZATION** WHICH WILL BE DIFFERENT IF YOU TARGET INTERNATIONAL MARKET

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

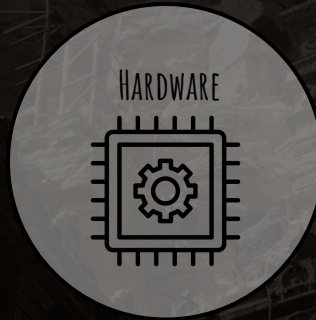
THIS IS OBVIOUSLY PART OF THE **INPUT / OUTPUT** IMPLICATION, YOU'LL NEED TO READ / WRITE FILES FOR A GAME. THEY ARE PREDOMINANT FOR VARIOUS THINGS LIKE **SAVE, CONFIGURATION FILES**, ETC...

IMPLICATION

THERE IS A LOT OF IMPLICATION REGARDING FILES MANAGEMENT, BEING **DEPENDANT** FROM THE **HARDWARE**, THE **ENCODING**, ETC... YOU'LL EITHER HAVE TO MANAGE IT, OR USE TOOLS PROVIDED BY LIBRARIES IN THE ENGINES

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

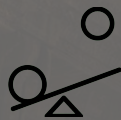
IT WILL BE HIDDEN BY MOST GAME ENGINE, AND YOU'LL NOT NEED TO WORRY ABOUT IT, BUT GAME ENGINE AND EXECUTABLE PRODUCED WILL BE DEPEND OF THE **HARDWARE** OF THE MACHINE OBVIOUSLY.

INPUTS

MOUSE, SPEAKERS, HEADSET, ETC... ARE ALL **HARDWARE COMPONENTS** WHICH ARE CONNECTED TO THE CONSOLE AND SENDING INPUT OR RECEIVING OUTPUT. ONE OF THE PURPOSE OF A GAME ENGINE IS TO PROPERLY CONNECT TO THAT



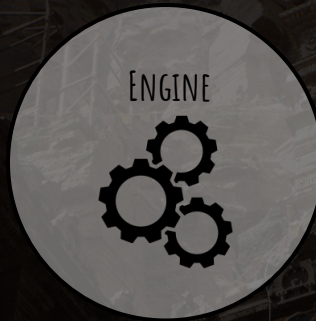
WHAT IS A GAME ENGINE ?



PHYSICS

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

THERE IS A DISTINCTION BETWEEN A **GAME ENGINE** AND A **PHYSIC ENGINE**. THEY ARE 2 THINGS COMPLETELY DIFFERENT. A GAME ENGINE WILL BE **CONTAINING** MOST OF THE TIME A CUSTOM OR COMMERCIAL PHYSIC ENGINE.

ROLE

THE ROLE OF THE PHYSIC ENGINE IS TO SIMULATE A WORLD, AND PROVIDE THE PHYSICAL LOGIC OF EVERY ELEMENT, COMPUTING FORCES, MANAGING COLLISIONS, ETC...

EXAMPLES

SOME COMMERCIAL GAME ENGINE NOW **EMBEDDED** A CUSTOM MADE PHYSIC ENGINE LIKE **CHAOS** FOR **UNREAL**, BUT SOME OTHER, LIKE **UNITY**, USES COMMERCIAL SOLUTION LIKE **PHYSX** MADE BY **NVIDIA**.

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

COLLISIONS



ENGINE



RAYCASTS



FORCES



PRIMITIVES



DEFINITION

COLLISIONS IS ONE OF THE MOST IMPORTANT SUBJECT WHEN IT COMES TO PHYSICS. IT IS THE **MECHANISM** THAT MADE SURE 2 OBJECTS WILL NOT PASS THROUGH EACH OTHERS IF THEY ARE SUPPOSED TO **COLLIDE**

GRANULARITY

WE'LL NOT GO INTO DETAILS, BUT **COLLISIONS** CAN BE **EXPENSIVE**, AND FOR THE MAJORITY OF OBJECT, COLLISIONS ARE **NOT PERFECT**, BUT USES **PRIMITIVES** ON WHICH PHYSIC ENGINE CAN COMPUTE **MUCH FASTER**

MATRIX

OFTEN, ENGINE OFFERS A **MATRIX COLLISION** WHICH ALLOWS TO DEFINE WHICH TYPE OF **OBJECT** SHOULD COLLIDE WITH OTHERS

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

RAYCAST IS PROBABLY THE MOST BASIC OPERATION FOR A PHYSIC ENGINE. IT CONSIST OF FIRING A RAY FROM A POINT, TO A DIRECTION, TO DETERMINE IF THAT RAY HIT SOMETHING

MULTI VS SINGLE

ALL ENGINES OFFERS THE POSSIBILITY FOR A RAY TO NOT BE STOPPED BY AN OBJECT, SO YOU CAN KNOW EVERY OBJECT THAT GET PASS THROUGH.

EXAMPLES

RAYCAST ARE USED IN A LOT OF SITUATION, LIKE CHECKING IF AN OBJECT IS IN FRONT, TESTING INTERACTION BETWEEN MOUSE AND AN OBJECT, BY DOING A RAYCAST, FOR LIGHTING, ETC...

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

FORCES



ENGINE



COLLISIONS



RAYCASTS



PRIMITIVES



DEFINITION

FORCES IS ANOTHER IMPORTANT ELEMENT WHEN IT COMES TO PHYSICS. THERE IS AT ANY TIME OF SIMULATION FORCES THAT GET APPLIED TO OBJECT LIKE GRAVITY FOR EXAMPLE

ADD FORCES

FORCES GENERATED BY ENVIRONMENT (LIKE GRAVITY) ARE **NOT THE ONLY FORCES** THAT CAN AFFECT AN OBJECT OBVIOUSLY. YOU CAN MANUALLY **ADD A FORCE** TO AN OBJECT, TO MAKE IT **MOVE**, OR WHEN IT **BOUNCES** FOR EXAMPLE

MASS

PHYSIC ENGINE AND FORCES ARE COMPLEXE, AND TRIES TO **SIMULATE** AS CLOSE AS POSSIBLE REAL PHYSICS, SO YOU'LL BE GIVEN A LOT OF **VARIABLES** WHICH ARE IMPLIED IN FORCE COMPUTATION LIKE **MASS**, **DRAW**, ETC...

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

PRIMITIVES



ENGINE



COLLISIONS



RAYCASTS



FORCES



DEFINITION

LIKE EXPLAINED A BIT EARLIER, IT IS BETTER TO USE **PRIMITIVE SHAPES** WHICH ARE WAY EASIER TO COMPUTER FOR PHYSIC ENGINE.

COMPLEX

IF YOU REALLY WANT **PRECISE COLLISION**, YOU CAN BUILD AN OBJECT WITH **MULTIPLE COLLIDER SHAPES**, LIKE A SPHERE FOR THE HEAD, CUBE FOR THE TORSO, ETC... OR USE A **COMPLEX COLLISION** WHICH WILL MATCH THE MESH

EXAMPLES

THERE IS 4 SHAPES THAT ARE COMMONLY USED : CUBE, SPHERE, CAPSULE AND PLANE



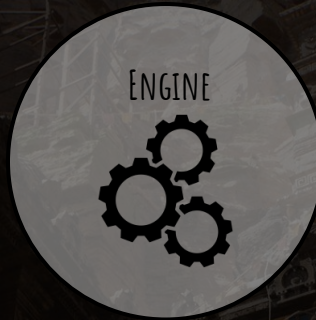
WHAT IS A GAME ENGINE ?



RENDERING

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

RENDERING IS ALSO AN **ENGINE** BY ITSELF, WAY MORE TIED TO ENGINE MOST LIKELY BUT STILL. RENDERING ENGINE ARE WRAPPER AROUND RENDERING LIBRARIES LIKE **DIRECTX** OR **OPENGL**

ROLE

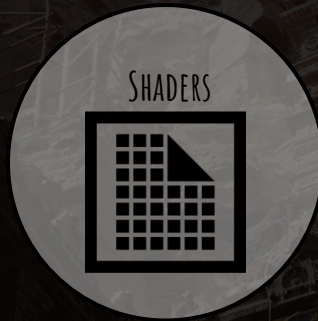
THE ROLE OF RENDERING ENGINE IS TO **COMPUTE** EVERY OBJECTS IN THE SCENE TO BE **FITTED** INTO THE **CAMERA**. IT INCLUDES A LOT OF **PROCESSING** AND **COMPUTATION** WHICH ARE INVISIBLE FOR DEVELOPERS

2D / 3D

SOME ENGINE ARE DOING A RENDERING ENGINE WHICH SUPPORT DIRECTLY **2D** AND **3D**, BY **WRAPPING** 2D WITH 3D. SOME OTHERS TAKE THE DECISION TO DO 2 DIFFERENT RENDERING ENGINE LIKE **UNITY**.

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

SHADERS ARE TINY PIECE OF CODE WHICH ARE **NOT LIKE STANDARD C++ CODE** WHICH DIRECTLY MANIPULATE PIXEL OR FRAGMENT OF 3D MODELS. A **FRAGMENT** IS A CHUNK OF PIXELS
SHADERS HAVE A VARIETY OF USABLE, LIKE ALTERING POSITION, COLOR, SATURATION OF ANY VERTEX

EXECUTION

SHADERS ARE IMPORTANT BECAUSE THEY ARE EXECUTED IN **PARALLEL ON GPU**, ALLOWING TO DO THE HEAVY STUFF FOR RENDERING THAT COULDN'T BE MADE ON CPU. SHADERS ARE RESPONSIBLE FOR MAKING MATERIALS WORKS BASICALLY

LANGUAGES

A SHADER IS ASSOCIATED WITH A SHADING LANGUAGE LIKE GLSL, HLSL, ETC...

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

CULLING IS A RENDERING TECHNIQUE MORE THAN IMPORTANT IN ORDER TO HAVE A PROPER FRAMERATE. IT CONSIST OF A SIMPLE THING : DO NOT RENDER A PIXEL WHICH WILL NOT BE VISIBLE

REASON FOR CULLING

THERE IS A LOT OF REASON FOR CULLING : PART OF THE MESH IS **HIDDEN** BY ANOTHER MESH, MESH IS **OUTSIDE** OF THE CAMERA VIEW, ETC...

DISTINCTION

WHILE **CULLING** IS THE PROCESS OF NOT VISUALLY COMPUTING SOME MESHES / VERTICES, IT SHOULD NOT BE CONFOUNDED WITH **DELETION**, A **CULLED MESH** WILL STILL BE LOGICALLY **PRESENT** REGARDING COLLISION, ETC...

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

MATERIALS ARE AS IMPORTANT AS MESHES. THEY ARE BASICALLY AN INSTANCE OF A SHADER, THE APPLICATION OF THE CODE INTO A MESH IN ORDER TO GIVE A COLOR, TO DEFINE HOW THE MESH REFLECT LIGHT, ETC...

TEXTURES

TEXTURES ARE DIFFERENT FROM MATERIALS, BUT WORKS IN DUO WITH THEM. TEXTURES ARE WRAPPED AROUND A MESH USING ITS UVs. THE LOGICAL COMPUTATION OF A TEXTURE IS MADE WITH SHADERS THROUGH MATERIALS

OPTIMIZATION

MATERIALS BEING OMNIPRESENT IN GAME, THERE IS A BUNCH OF TECHNIQUES IN ORDER TO OPTIMIZE THEM LIKE DOING INSTANCES, BATCHING THEM INTO 1 DRAW CALL, REDUCE SHADER COMPLEXITY, ETC...

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

A GAME WITHOUT **LIGHTING** WILL BE COMPLETELY **BLACK**. FROM THIS STATEMENT, IT IS EASY TO SEE HOW IMPORTANT LIGHTING IS. IT WILL BRING **ATMOSPHERE**, **VISIBILITY**, AND **AMBIANCE** INTO THE GAME.

CRITICAL

AS IMPORTANT AS THEY ARE, LIGHTING IS ALSO **CRITICAL PERFORMANCE** WISE. **SHADOWS** COMES FROM LIGHTING, **AMBIENT OCCLUSION** COMES FROM LIGHTING, ETC... AND ALL THIS CAN COST A LOT AND QUICKLY ESCALATE

TYPE

THERE IS VARIOUS **TYPE** OF **LIGHTING**, FROM SIMPLE **POINT LIGHT** TO **DIRECTIONAL LIGHT**, YOU'LL BE BRINGING LIFE TO A SCENE BY SIMULATING THE **SUN**, THE **MOON**, **LANTERNS**, **LAMP POST**, ETC... THANKS TO **LIGHTS**

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

VFX ARE IMPORTANT TO ANY GAME THAT WANT TO BRING **IMMERSION**, **GAMEPLAY FEEDBACKS** AND **VISUAL FIDELITY**. VFX ARE GRAPHICAL ELEMENT, USING **SHADERS** AND **MATERIALS** THAT CREATE VISUAL EFFECTS

OPTIMIZATION

VFX ARE A BIT BLURRY ON THE **OPTIMIZATION** SIDE. THEY TYPICALLY RUNS ON **CPU** WHEN LOGICALLY INTERACTION LIKE COLLISION IS NEEDED. OTHERWISE THEY CAN USE **GPU** AND BE REALLY EFFECTIVE AND DISPLAY MILLIONS OF VERTICES.

EXAMPLES

AN EXPLOSION FOLLOWED BY A SMOKE CLOUD, BLOOD SPRAYING FROM A WOUND, ON FLOOR INDICATOR SHOWING A SPELL WILL BE CASTED, ETC...



WHAT IS A GAME ENGINE ?



UI

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

HUD STANDS FOR **HEAD-UP-DISPLAY**. IT CAN BE SEEN AS AN OVERLAY WHICH STAYS PERMANENTLY (OR NEARLY PERMANENTLY) WHICH YOU ARE PLAYING A GAME.

INTERFACE

HUD SHOULD NOT BE MISTAKEN WITH **INTERFACES** WHICH ARE DIFFERENT. INTERFACE CAN BE INTERACTABLE, LIKE AN OPTION MENU, AN INVENTORY, ETC... HUD IS REALLY JUST A DISPLAYER, LIKE THE NAME SUGGEST.

EXAMPLE

HEALTH BAR, AMMO FEEDBACKS, SHAKING SCREEN, ETC...

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

WE'LL NOT DIVE RIGHT NOW INTO **ANCHORS** BUT IT IS AN IMPORTANT CONCEPT EXISTING EVERYWHERE. IN ORDER TO HAVE YOUR GAME RESPONSIBLE BASED ON **SCREEN RESOLUTION** ETC... DEVELOPERS USES **ANCHORS**.

RESPONSIVENESS

RESPONSIVENESS IS IMPORTANT IN GAME WHERE THERE IS MULTIPLE POTENTIAL **RESOLUTIONS**. **ANCHORS** ENSURE THAT WHATEVER **WIDTH** THE SCREEN IS, A ELEMENT WILL ALWAYS TAKE **FULL WIDTH** FOR EXAMPLE

UPDATE

IN ADDITION TO **RESPONSIVENESS**, USING **ANCHORS** IS PRIMORDIAL EVEN IF YOU TARGET ONLY 1 **RESOLUTION**. IF AN ARTIST DECIDES TO **UPDATE** THE WIDTH OF AN UI, **ANCHORS** WILL ENSURE THAT EVERYTHING STILL **WORKS** PROPERLY

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

UI IS THE MAIN ELEMENT WHEN IT COMES TO GIVE FEEDBACK TO A PLAYER WHEN HE DOES SOME ACTIONS. UI WILL BE RESPONSIBLE FOR DEFINING ALL THE FEEDBACKS, AND DEVELOPER WILL ENSURE THAT EVERYTHING IS IN PLACE

EXAMPLES

A NOTIFICATION BOX WHEN A MESSAGE IS SENT, SCREEN SHAKING WHEN TAKING DAMAGE, AMMO INDICATOR FLASHING WHEN RELOAD IS NEEDED, ETC...

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

LOCALIZATION



HUD



ANCHORS



FEEDBACKS



LAYOUT &
HIERARCHY



INTERACTIONS



DEFINITION

NOWADAYS, NEARLY EVERY GAME IS SELL ON INTERNATIONAL MARKET, IN ORDER TO OPTIMIZE SELLS. GIVEN THAT, LOCALIZATION OF UIs, WHICH DISPLAY TEXT IS A KEY ELEMENT WHICH IS WELL KNOWN BY ENGINES.

EXAMPLES

A LOT OF ENGINE NOW PROVIDES **PLUGINS** OR **NATIVE** SOLUTIONS IN ORDER TO SIMPLIFY **LOCALIZATION** OF TEXTS

IMPLICATION

TRADUCTION IS OFTEN MADE BY **OFFSHORE COMPANY**, WHICH ARE NOT WORKING IN ENGINE BUT EXCEL OR JSON FOR EXAMPLE. IT FORCES DEVELOPERS TO HAVE A SOLUTION TO EASILY **IMPORT** THAT FILE.

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

LAYOUT &
HIERARCHY



HUD



ANCHORS



FEEDBACKS



LOCALIZATION



INTERACTIONS



DEFINITION

HIERARCHY IS NOT A CONCEPT TIED TO UI BUT IT IS CENTRAL IN UI DEVELOPMENT. EVERY ELEMENT IS CHILD OR PARENT TO ANOTHER ONE, GIVING THE POSSIBILITY TO DICTATE POSITION, VISIBILITY, ETC... AT DIFFERENT LEVEL

LAYOUT

LAYOUT USES THE LOGIC OF HIERARCHY BY ENSURING PLACEMENT OF CHILDREN BASED ON A SET OF PROPERTY DEFINED IN THE LAYOUT. THERE IS VARIOUS LAYOUT LIKE VERTICAL, HORIZONTAL, GRID, ETC...

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

INTERACTIONS



HUD



ANCHORS



FEEDBACKS



LOCALIZATION



LAYOUT & HIERARCHY



DEFINITION

INTERACTIONS IS OBVIOUSLY AN IMPORTANT ASPECT OF UIs. IT IS THE WAY A PLAYER CAN INTERACT WITH VARIOUS SYSTEM LIKE OPTIONS, INVENTORY, ETC...

DIFFERENTIATING

THERE IS ALWAYS A MORE OR LESS STRICT **DISTINCTION** BETWEEN UI AND GAME INPUTS. IN ORDER TO AVOID TO RECEIVE INPUT ACTION AT GAMEPLAY TAG, BUT INTEGRATION OF THAT DIFFERS FROM ENGINE TO ENGINE



WHAT IS A GAME ENGINE ?



NETWORK

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

ABSTRACTION



REPLICATION



PACKETS



AUTHORITY



EXTERNAL
TOOLS



DEFINITION

NETWORKING IS A COMPLEX SUBJECT, EVEN MORE IN VIDEO GAME WHERE IT CREATES A LOT OF COMPLICATION ON THE DEVELOPMENT SIDE. ENGINE TENDS TO **ABSTRACT** AND **HIDE** AS MUCH AS POSSIBLE WHEN IT COMES TO NETWORK

OPEN CODE

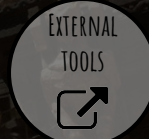
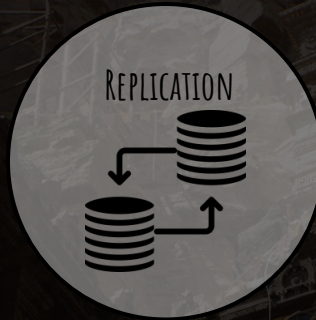
HAVING ACCESS TO SOURCE CODE WHEN IT COMES TO NETWORKING IS REALLY IMPORTANT TO UNDERSTAND WHAT IS GOING ON, HOW CALLS ARE MADE, ETC...

ENGINE

SOME ENGINE, LIKE **UNITY**, ARE BAD WHEN IT COMES TO **MULTIPLAYER**, EVEN IF THEY PROMISED A NEW NETWORKING ENVIRONMENT. SOME OTHERS, LIKE **UNREAL**, DO BETTER BUT ALL IN ALL, A LOT OF WORKS NEEDS TO BE DONE

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

REPLICATION CONCEPT IS QUITE STRAIGHTFORWARD IN A NETWORK ENVIRONMENT, IT CONSISTS OF ENSURING THAT EVERYONE IS UP TO DATE WITH EVERYONE ELSE

SERVER

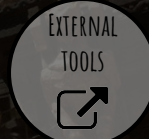
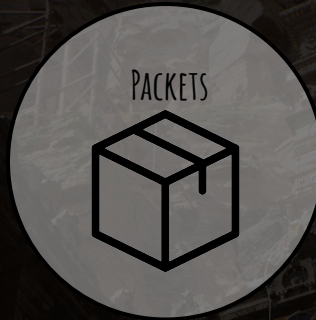
SERVER IS A KEY COMPONENT IN NETWORKING AND REPLICATION. BASICALLY, THE SERVER IS RESPONSIBLE FOR BEING THE PLACE WHERE REPLICATION WILL BE DISPATCHED INTO EVERY CLIENTS (OR NOT)

TECHNIQUES

RELIABLE, NON-RELIABLE, RPC, PROPERTY REPLICATION, PREDICTION, ETC... THERE IS A LOT OF CONCEPTS AND TECHNIQUES AROUND WHICH WILL NOT BE COVERED IN DETAILS

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

PACKETS IS THE NAME GIVEN TO MESSAGE THAT PASSES ON THE NETWORK AND CONTAINS THE INFORMATIONS. IT IS ALSO THIS CONCEPT OF PACKETS THAT IS PROPERLY ANALYZE WHEN THERE IS PACKETS LOSS, PERFORMANCE ISSUE, ETC...

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

AUTHORITY



ABSTRACTION



REPLICATION



PACKETS



EXTERNAL TOOLS



DEFINITION

THE **AUTHORITY** CONCEPT IS PRIMORDIAL IN A NETWORK ENVIRONMENT. IT WILL DICTATE IF THERE IS A **VALIDATION** CHECK BEFORE SETTING SOME VARIABLE, IF SETTING VARIABLE MUST BE **DONE ON SERVER**, ETC...

NETMODE

A MULTIPLAYER GAME WILL ALWAYS BE ASSOCIATED WITH A **NETMODE** WHICH INDICATE HOW **AUTHORITY** AND OTHER STUFF LIKE OWNERSHIP IS DRIVEN.

EXAMPLES

BASICALLY, GIVEN EVERY ACTOR IS **SPAWNED** BY **SERVER**, WHEN **SERVER** IS **AUTHORITATIVE**, EVERY ACTOR HAS **AUTHORITY** ON **SERVER SIDE**. BUT FOR EXAMPLE, A **VFX** **SPAWNED** ON A **CLIENT** HAS **CLIENT** FOR **AUTHORITY**.

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

WHEN IT COMES TO PROFESSIONAL DEVELOPMENT FOR A MULTIPLAYER GAME, IT WILL ALWAYS REQUIRES EXTERNALS TOOLS LIKE GAMESLIFT, OR SIMILAR, WHICH WILL BE RESPONSIBLE FOR STORING DATAS, CONNECTION CHECKING, ETC...



WHAT IS A GAME ENGINE ?



BUILDING & SHIPPING

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

COOKING IS THE ACTION OF CONVERTING ASSETS LIKE TEXTURES, AUDIO FILES, ETC... WHICH HAVE A SPECIFIC FORMAT LIKE JPEG, WAV, ETC... INTO A COMPREHENSIVE AND FORMATTED FORMAT FOR TARGETED PLATFORM

GOAL

COOKING IS PART OF THE BUILDING PROCESS, WHICH AIMS TO CREATE A FILE / BUNCH OF FILE THAT WILL BE EXECUTABLE ON THE TARGETED PLATFORM

EXCLUSION & PROCESS

GENERALLY SPEAKING, COOKING PROCESS WILL ONLY INCLUDE ASSETS THAT ARE IMPLICITLY REFERENCES FROM SCENES. EVERYTHING ELSE WILL NOT BE ADDED, LEADING TO ERRORS / PINK TEXTURE FOR EXAMPLE.

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

TRANSPARENT



COOKING



EXECUTABLE



DEFINITION

BUILDING PROCESS SHOULD BE **TRANSPARENT** OR NEARLY TRANSPARENT. THE PROCESS IS DRIVEN BY THE PLATFORM YOU ARE BUILDING ON, AND ENGINE ARE PROCESSING EVERYTHING **UNDER THE HOOD**.

DLC & UPDATES

IT CAN BECOME A COMPLEX SUBJECT, BUT WHEN YOU WANT TO DEPLOY AN **UPDATE** OR A **DLC** FOR A GAME, THE IDEA IS TO ONLY COOK **MODIFIED** AND NEW ASSET IN ORDER TO HAVE A **LIGHT UPDATE**. ENGINE TRIES TO HANDLE THAT

WHAT IS A GAME ENGINE?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

THE FINAL RESULT OF A BUILD WILL BE AN **EXECUTABLE**. EITHER A .EXE, .PKG, ETC... BASED ON THE **PLATFORM** BUT THE FINAL ISSUE IS SIMILAR. THE EXECUTABLE WILL BE ABLE TO BE **LAUNCHED** ON THE PLATFORM IN ORDER TO **PLAY**.

TESTING

EXECUTABLES WILL BE GIVEN TO **QA**, **MANUFACTURER**, ETC... FOR THE GAME TO BE TESTED. IT IS THE ONLY PRODUCT THAT IS AWAITED.



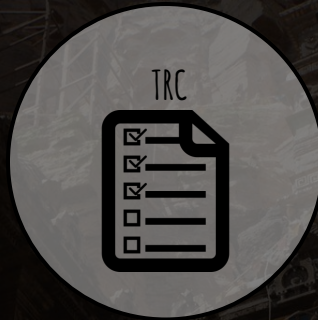
WHAT IS A GAME ENGINE ?



MULTI-PLATFORM

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

TRC STANDS FOR TECHNICAL REQUIREMENT CHECKLISTS. THIS IS A LIST OF FEATURES THAT NEEDS TO BE IMPLEMENTED FOR A PLATFORM.

VALIDATION

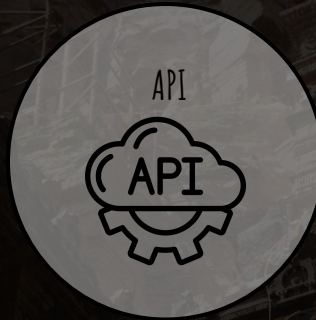
TRC ARE IMPORTANT TO HANDLE BECAUSE YOUR GAME WILL BE REFUSED BY **MANUFACTURER** IF YOU CAN'T PASS THE CHECKLIST. IT ALSO MEANS DURING **DEVELOPMENT** YOU NEED TO BE CAREFUL ABOUT **ABSTRACTION**

EXAMPLES

SAVE CORRUPTION, ACHIEVEMENTS UNLOCK, CONTROLLER SUPPORT, COMPATIBILITY, ETC...

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM



DEFINITION

OBTAINING TRC, IT MEANS YOU NEED TO WRITE CODE THAT IS SPECIFIC FOR THAT ENGINE. IN ORDER TO ACHIEVE THAT, EVERY CONSOLE COMES WITH AN API PROVIDED BY THE MANUFACTURER EXPOSING CONSOLE'S FEATURES

WRAPPER

APIs ARE NEARLY ALWAYS WRITTEN IN C++ (OR CONSOLE LANGUAGE), YOU'LL THEN HAVE WRAPPER THAT WILL ENCAPSULATE THAT IN ORDER TO BE USED IN YOUR ENGINE, FOR EXAMPLE C# ON UNITY

WHAT IS A GAME ENGINE ?

1. TOOLBOX
2. AI
3. INPUT / OUTPUT
4. PHYSICS
5. RENDERING
6. UI
7. NETWORK
8. BUILDING & SHIPPING
9. MULTI-PLATFORM

SPECIFICITIES



TRC



API



DEFINITION

EVERY ENGINE IS DIFFERENT WHEN IT COMES TO WHAT THEY OFFER. FOR EXAMPLE, PS4 OFFERS ACHIEVEMENTS WHILE SWITCH DOESN'T. THE WAY TO IMPLEMENT THINGS IS DIFFERENT, ETC...

ABSTRACTION

EACH ENGINE IS DIFFERENT, IT LEAD TO DEVELOP PROPERLY ABSTRACTION ABOUT FEATURES ON YOUR GAME. DEVELOP A WAY TO UNLOCK ACHIEVEMENTS REGARDLESS OF THE PLATFORM, AND SIMPLY HAVE A PROPER IMPLEMENTATION



PROS

ACCESSIBILITY



EVEN IF A GAME ENGINE CAN LOOK TERRIFYING / OVER COMPLICATED AT FIRST BECAUSE OF EVERYTHING THAT IS ALREADY DONE AND PRESENT, IT IS A FALSE IMPRESSION.

YOU'LL LOSE WAY MORE TIME TO DEVELOP YOUR OWN ENGINE INSTEAD OF LEARNING FUNDAMENTALS AND BASICS OF AN ENGINE.

SMALL TEAM



A LOT OF BIG COMPANIES RUNS AN IN-HOUSE ENGINE BECAUSE THEY HAVE THE WORKFORCE FOR IT.

MAKE NO MISTAKE, DEVELOPING AN ENGINE NEEDS A LOT OF DEVELOPERS FULLY FOCUS ON DEVELOPING AND IMPROVING THE ENGINE.

SMALL TEAMS CANNOT AFFORD THAT

CHEAP



IT GOES WITH INDEPENDENT, IN THE LONG RUN, EVEN IF THERE IS A FEE APPLY BY COMMERCIAL ENGINE, IT IS WAY CHEAPER THAN CREATING A NEW ENGINE.

THERE IS NO SECRET WHY EVEN BIG COMPANIES LIKE BLIZZARD, OR CD PROJEKT USE COMMERCIAL ENGINE FOR SOME GAMES, OR COMPLETELY ABANDONED IN-HOUSE ENGINE.

PROS

UP-TO-DATE



THAT'S ALSO ONE OF THE MOST **IMPORTANT** FACTOR ABOUT USING A **COMMERCIAL ENGINE**.

IT WILL ALWAYS BE **UP-TO-DATE** WITH **STANDARDS**. EVEN IF ENGINE ARE COMPLETELY DIFFERENTS, A LOT OF **COMPANIES** ARE USING THEM, SO THEY NEED TO STAY **COMPETITIVE** REGARDING OTHER **COMMERCIAL ENGINE**.

GAMEPLAY FOCUS



AS A **GAME COMPANY**, YOU WANT TO DELIVER A **GAME**, SOMETHING RELATED TO **GAMEPLAY**. A **GAME ENGINE** IS JUST A **TOOL** TO DELIVER A **GAME**, BUT IT IS **NOT** THE **DELIVERED PRODUCT**.

HAVING MORE TIME TO **FOCUS** ON **GAMEPLAY** AND **MECHANICS** OF THE **GAME** WILL **INCREASE** THE **FINAL QUALITY** OF THE **GAME**

TRANSFERABLE KNOWLEDGE



GENERAL KNOWLEDGE LIKE **ISSUES**, **LOGICALLY THINKING**, **TECHNIQUES** FOR **VIDEO GAMES**, **ETC...** WILL BE **OBVIOUSLY TRANSFERABLE** FROM ANY **ENGINE**.

BUT HAVING KNOWLEDGE IN A **COMMERCIAL ENGINE** WILL BE **FULLY TRANSFERABLE** IF YOU **LEAVE** FOR **ANOTHER COMPANY** THAT IS USING THE **SAME ONE**.



CONS

CONS

LEARNING CURVE



THAT'S A REALITY FOR EVERY GAME ENGINE YOU'LL BE USING. THERE IS ALWAYS A **LEARNING CURVE**, WHICH IS RELATED TO THE **COMPLEXITY** OF THE ENGINE.

THIS **LEARNING CURVE** CAN SEEMS A **BIG OBSTACLE** FOR SOME PROFILE / TEAM, WHICH COULD LEAD TO **BAD DECISIONS** OR A **FAIL** IN THE PROJECT

FOUNDATION



IT IS KIND OF LINKED TO THE **LEARNING CURVE**, BUT EVERY ENGINE COMES WITH A **BASE CLASS FOUNDATION**. IN UNREAL UNITY THIS ONE IS MASSIVE AND EXPLAIN FOR EXAMPLE THE HUGE **LEARNING CURVE**.

ANYWAY, THIS BASE WILL BE **INEVITABLE** AND IT CAN BE SEEN AS A **CONS** FOR A TEAM LIKING AN ENGINE EXCEPT THE **FOUNDATION**.

DEPENDANT



EVEN IF ENGINE ENSURE TO STAY UP TO DATE REGARDING FEATURES, THERE IS STILL **BUGS** AND **LACKS** IN AREAS.

IN THAT SITUATION, YOU'LL BE **DEPENDANT** FROM THE **ENGINE DEVELOPERS**, YOU'LL NEED TO WAIT FOR **PATCHES** AND DEVELOPMENT IF YOU ENCOUNTER AN **ISSUE** AND THE ENGINE IS **NOT OPEN SOURCE**.

CONS

EFFICIENCY



AN ENGINE WAS NOT **DESIGNED** SPECIFICALLY FOR YOUR GAME, SO IT MAY BE LESS **EFFICIENT** THAN CODE YOU WRITE SPECIFICALLY FOR YOUR GAME.


IT ALSO COMES WITH **IMPLICATION** ON NETWORK CODE FOR EXAMPLE WHICH CANNOT BE AS **OPTIMIZE** AND SPECIFIC AS YOU WOULD WANT. FOR EXAMPLE, **MMORPG** MORE THAN OFTEN COMPLETELY REWRITE NETWORK

DOCUMENTATION



IN **THEORY**, THERE IS DOCUMENTATION IN ANY GAME ENGINE WHICH DOCUMENT **CODE**, **FEATURES**, ETC..., AND GLOBALLY HOW TO USE ANY COMPONENT PROVIDED BY THE ENGINE

IN **REALITY**, DOCUMENTATION IS SOMETIMES COMPLETELY **ABSENT**, OFTEN **INCOMPLETE** OR **UNCLEAR**. THIS CAN LEADS TO A LOT OF **TIME LOST**, OR NEEDING TO READ **SOURCE CODE**.

The background image shows ancient stone ruins, possibly Mayan or Aztec, with intricate carvings and hieroglyphs. The scene is dark and atmospheric, with a central text overlay. The text consists of a large question mark followed by the word "QUESTIONS?".

? QUESTIONS ?