Lesson 2

Basis

Nicolas Serf
serf.nicolas@gmail.com

Basis game development

Basis game development

Assets

Basis Game Development

1. Asset
2. World
3. Object
4. Details
5. Finder
6. Console
7. Template

Definition

An **asset** is the **basis** of every game engine and every project. These are the **elements** you'll find in your explorer and **constitute** your project. A **script**, a **texture**, an **image**, etc... All of this are **assets**

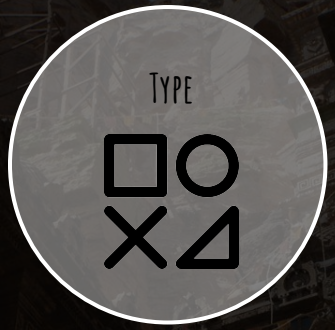
Content

Assets is the **content** of your game. Everything is **asset** and when it comes to **cooking**, engine will be using the **assets** in your project to be **packaged** into the **executable**.

Import

Some assets will be **created** from the engine, some will be **imported**. For most cases, you can simply **drag & drop** a file from **windows's explorer** into **engine explorer** to import it.

Basis Game Development



Definition

Type of assets depends on the engine you're using but the idea is that every asset has a type which goes from **image type** (jpg, png, etc...) to **sound type** (Wav, mp3, etc...)

Transformation

It is also important to notice that an asset is not **mandatory** of the **type** from which you **imported**. There is some format that are **proprietary**, or not suits to the engine, which convert them into an **internal** format

Editor

World

Basis game development

1. Asset
2. World
3. Object
4. Details
5. Finder
6. Console
7. Template

Definition

A world is composed of **multiples levels**. The name can vary but the **principle** is exactly the **same**. A level is where the **level designers** will be working to create a **place** where the player will **interact** and **play**

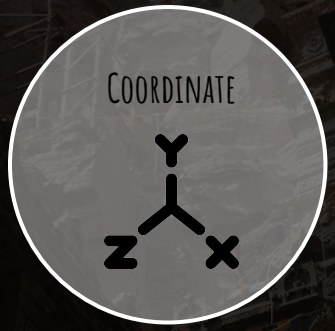
Asset

A level is an **asset**, like everything, and it will **contains** all information about every **object** you've placed in it.

Transition

Levels are the main elements when it comes to **loading a map**, having a **loading screen**, etc...

Basis game development



Definition

Coordinate is not a concept tied to world directly, but is **central** to it. A location will always be **related** to the **world 0, 0, 0**. The world offers the **coordinate system** and **reference point** for every objects

Left hand vs right hand

This is something you really need to **take care**. Some engine take **left hand** as **coordinate** when others take **right hand**. It leads to having a **X axis** inverted between **forward** and **backward**

Y vs Z up vector

Also important, we are used to **Y** being the **upper vector** because of 2D, but for most game engine, **Z** is the **upper** vector will Y is the left vector

Basis game development

1. Asset
2. **World**
3. Object
4. Details
5. Finder
6. Console
7. Template

Definition

World is what manage the levels underneath when you are **loading levels**, you are actually adding a **level** into the **world**. The design process has to take that into consideration because **multiple level** can **layer**.

Artistic task

Designing a **level** is an **artistic task**, and developers are usually **not involved** but you must be aware of world design feature like for example **world partition** on UE

Basis game development

1. Asset
2. World
3. Object
4. Details
5. Finder
6. Console
7. Template

Definition

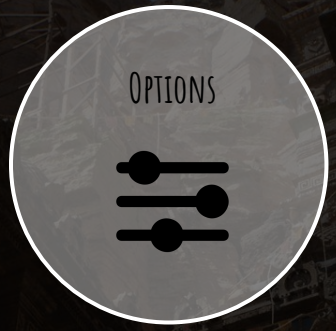
When you are designing levels within the world, you have access to a **variety** of **widgets** that offers options to modify the object like **moving**, **scaling**, **rotating** for the most common one.

Others

Transform widget is not the only one and this will be dependant on the engine, but you can for example find **grid size**, **rotation scale**, **snapping**, **gizmos**, etc...

Editor

1. Asset
2. World
3. Object
4. Details
5. Finder
6. Console
7. Template



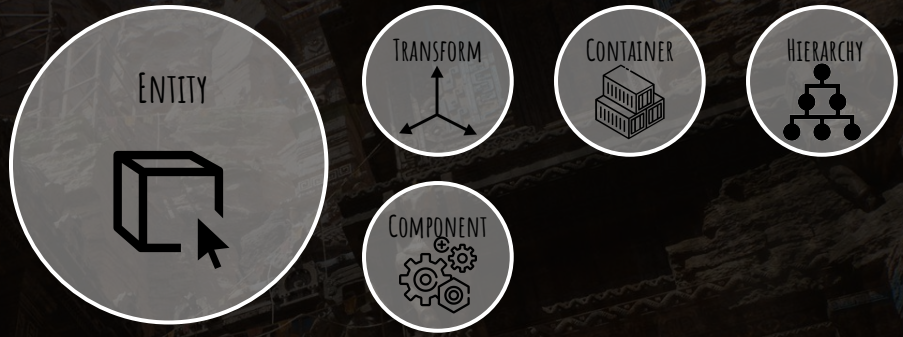
Definition

When you're working in the world, engines also offers **various options** to alter the **looks** of the scene. Like showing the **depth**, a **wireframe view**, **showing navmesh**, **changing resolution**, etc...

Editor

Object

What is a game engine ?



Definition

Object have different names based on engine : **Gameobject**, **actor**, etc... but the concept stays the same. They are **objects** which will be **placeable** into a level.

An empty box

You'll understand it in the next slides, but an object is **basically** an **empty box** with a **transform**, on which you'll be **connecting** a lot of things in order to give **functionalities**

What is a game engine ?

1. Asset
2. World
3. Object
4. Details
5. Finder
6. Console
7. Template

Definition

If objects are **placeable**, it means they have a **transform**. A transform is a set of 3 property : **Location**, **Rotation** and **Scale**.
Location is a **vector3** representing **x,y,z position** in the **world**
Rotation is a vector3 representing **Pitch**, **yaw**, **Roll** in the **world**
Scale is a vector3 representing x,y,z **scaling on axis** in the **world**

World vs local

Related to hierarchy, **local transform** shouldn't be **confused** with **world transform**. A local transform will always be **relative** to **parent**, while world transform will be relative to **world** 0, 0, 0.

What is a game engine ?

1. Asset
2. World
3. Object
4. Details
5. Finder
6. Console
7. Template

Definition

As explained a bit earlier, an object by **itself** is nothing but an **empty thing** in a level. It comes to life when you bring element to it.

Example

There is as many element that you can add, as your imagination comes with. Anything can be added to an object, **a mesh** to have a **representation**, **a sound player** to have **sound**, etc...

What is a game engine ?

Definition

Another really important point about objects is the **hierarchy system**. Every object is a **child** of **another**. This hierarchy explains the local / world difference and have also some implication

Root

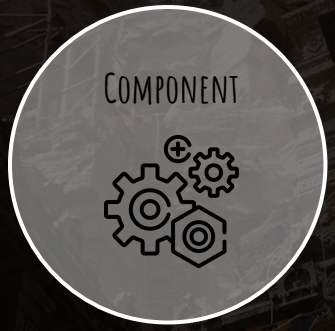
A **level** must be seen as a **graph**, and for any graph there is a **root**. This means **top level object** in a level are actually **child** of the **root** of the level, which is invisible in the editor.

Implication

When an object is **moving**, **rotating** or **moving**, it will directly **affect** the **children** because of **natural hierarchy inheritance** of **transform** property. Thankfully, there is a way to **avoid** that in every engine.

What is a game engine ?

1. Asset
2. World
3. Object
4. Details
5. Finder
6. Console
7. Template



Definition

A component is basically an **encapsulated logic** which can be connected to an object in order to gives it new functionalities. There is **graphical** component and **logical** component.

Engine's components

Every engine comes with a set of **component** already **defined** and **implemented**. Takes for example a **mesh**, a **skeletal mesh**, an **audio player**, etc... All of this are components provided by engine.

Custom components

There is **multiple** ways to develops new **features**, one is to bring a new feature through a **component**. Imagine an **inventory**, if you want our actor to have an inventory, simply add that **component**

Editor

Details

What is a game engine ?

1. Asset
2. World
3. Object
4. Details
5. Finder
6. Console
7. Template

Definition

Details allows to have **more information** about something as the name suggests. In engine, when you select an **entity** in the **world**, the details panel will update with the **informations** of that **object**

Multi-selection

Multi-selection is a thing in some engine When you are selecting **entities** of same **type**, or containing same **components**, you may be able to edit **multiples instance's properties** at the same time

What is a game engine ?

1. Asset
2. World
3. Object
4. Details
5. Finder
6. Console
7. Template

Definition

A **property** is anything contains within the **entity**. It will **differ** based on which engine you use, but the logic stays the same. Features you develop **exposes** some **properties** that designers will **tweak**.

Exposing

It is **developer's job** to **expose** some key variables of the feature to be tweakable. They will appears in the **detail panel** when you **select** the entity / component.

Runtime

Changing properties at **runtime** may be something you want to **maintains** for designers to easily tweaks properties to try things, but it is not always as **straightforward** as that if there is **side effects**

What is a game engine ?

1. Asset
2. World
3. Object
4. Details
5. Finder
6. Console
7. Template

Definition

You can **customize** display of **properties** in the details. It can be more or less difficult, based on the engine and the tools they offers but it is always possible.

The good, the bad, the ugly

Doing editor modification should always be considered with **care**, because you may completely **messed-up things**. But when used **properly**, it is a **powerful** way to increase **productivity**.

Tab creation

You can even create **complete new tab** window to be used in the editor, but it will **not be covered** in this lesson.

Editor

Finder

What is a game engine ?

Definition

Finder is like a **windows explorer**, but **specific** to your **project** location. It is the place where every assets is located.

Root

There is always a **root folder**, which is something **not shown**, but every assets needs to be **located** in it. The assets here will be **translated** into **comprehensive format** for engine, and able to be **cooked**

Moving stuff

When you **move** stuff in and out of **content folder**, be careful to do it **from engine**, and **never** from **windows explorer**. You may **messed-up** things, because engine will not **transform** the assets sometimes.

What is a game engine ?

Definition

Every finders comes with a **filter feature** that allows to show only some **type of assets**, or **filter by name**, etc...

Implication

This is specific to **project** and **companies**, but having a **filter options** can be impacting how you architecture the **file folder**.

What is a game engine ?

Definition

Obviously, being the **location** of **assets**, **finder** will offer the possibility to **create** assets from it. You'll most likely have to **right click** in it, and choose the **type of asset** you want to **create**.

Naming convention

Naming convention is really important to stay **organized** regarding file hierarchy. If you choose to start every material with **M_**, be sure to **stick** with that **rule** in the **entire project**.

Editor

Console

What is a game engine ?

Debug

Logging

Definition

Every engine comes with a **console** that allows to see **debug message**, **errors**, **warning**, etc... That's the **primary tool**, with **breakpoint**, that'll allows to **track issues**

Warning

Whatever engine you use, **do not ignore warning**, in the short term, they could be **nothing** but in the long run, they may be issue regarding **performance**, **unknown bug**, etc...

What is a game engine ?

1. Asset
2. World
3. Object
4. Details
5. Finder
6. Console
7. Template

Definition

Logging process is **primordial** for development. It is what **fills debug console** with **messages**. You'll most likely log with a **category**, and some log are **generated** by the **engine** when an **error occurred**.

Creation

It will depends of the engine used, but most of them allows to **create** new **logging categories**, either from **visual** or **code side**. It will allows to **develop features** independent with a proper **logging category**.

Filters

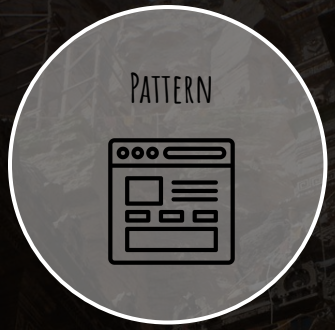
When it comes to **filtering**, **error**, **warning**, **info** are some categories that are common to every engine. But given you can create **new logging category**, you may be able to **filter** with the custom categories

Editor

Template

What is a game engine ?

1. Asset
2. World
3. Object
4. Details
5. Finder
6. Console
7. Template



Definition

A **pattern** is like the name suggest, something that will be **similar** and **followed** by **everything** that **starts** from it.

Cake example

Take a **cake** as an example. You'll be using a **mould** to make the **foundations**. Then, you can **decorate** the cake as you liked, they may all be **different** in the end, but with a **common base**, **immutable** for all.

What is a game engine ?

1. Asset
2. World
3. Object
4. Details
5. Finder
6. Console
7. Template

Definition

The operation made by a **template** is quite **simple**, making sure that **every instance** coming from that **template** is **up to date** with the template;

Inheritance logic

You can take in **comparison standard inheritance** of class system. When you do a **modification**, it is **implicitly affecting child class**. Template have the same logic.

Overridden properties

When you **modify** some **properties** (like **position**, a **variable** in a **component**, etc...) of a template, it **may not affect children** if they have modifies that **property**. **Children** have **priority** in that case.

What is a game engine ?

1. Asset
2. World
3. Object
4. Details
5. Finder
6. Console
7. Template

Instances

Pattern

Operation

Definition

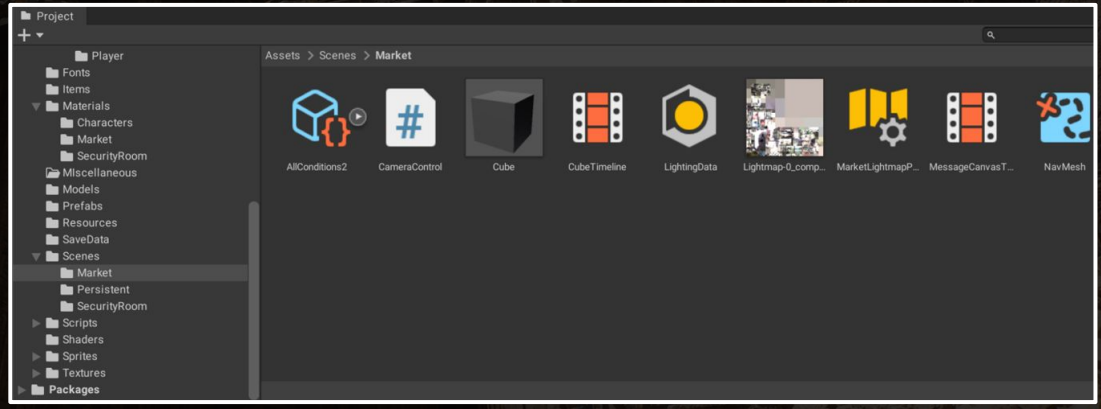
An **instance** is a template which is **placed** into the **world**, on which you'll be able to modify properties, and which will be **affected** by **changes**.

Instance vs child

A **child** of a **template** still remains a **template**, you can create a **hierarchy** of **template**, one being more **specific** from the higher one, but it is still not an **instance**. The difference is


UNITY

Unity Asset & Finder

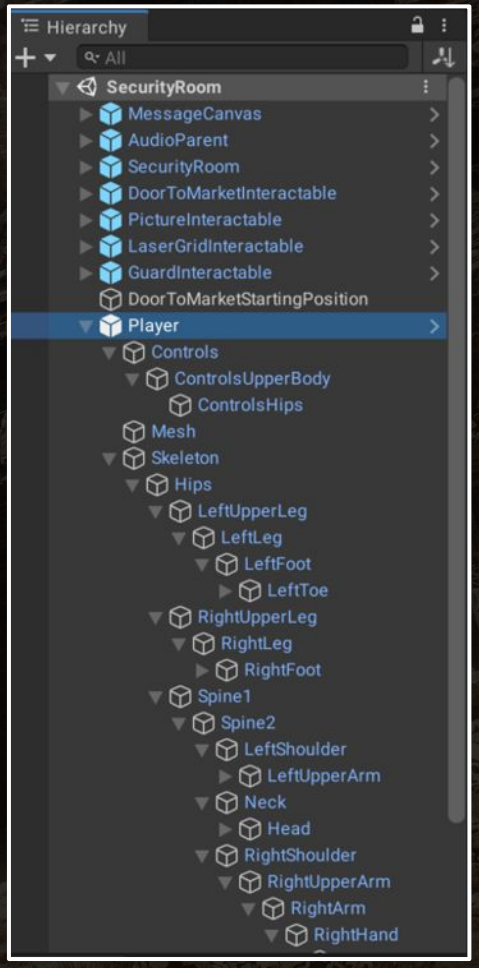


UNITY WORLD


Game
Scene
Asset Store
Sprite Editor
Shaded
2D
Gizmos
All
ATC
CORP
Persp

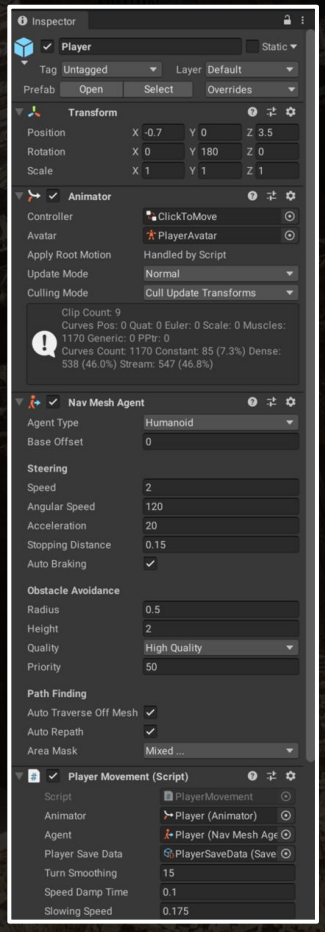
UNITY

OBJECT

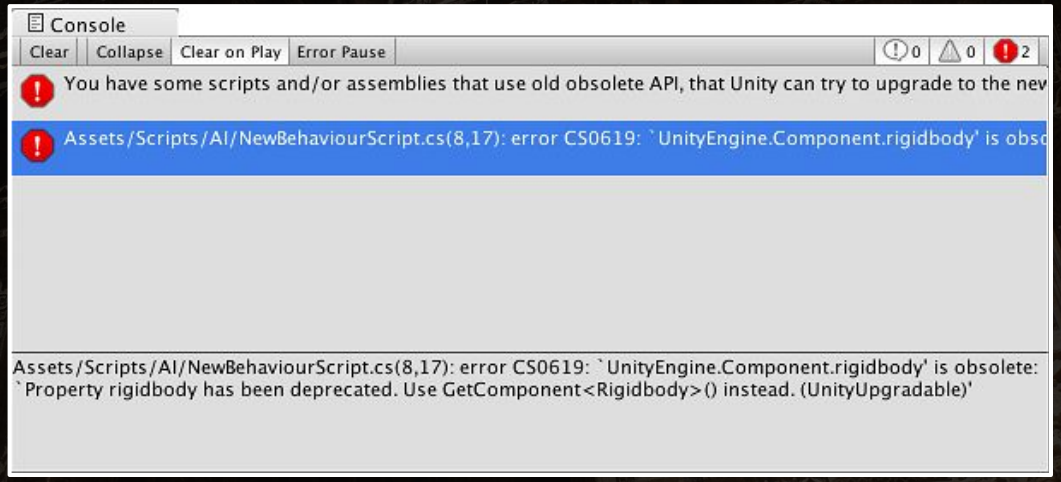


UNITY

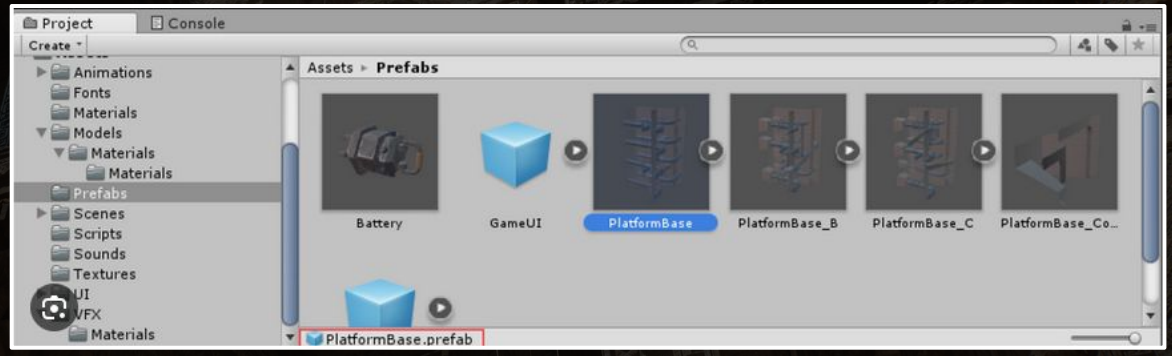
DETAILS

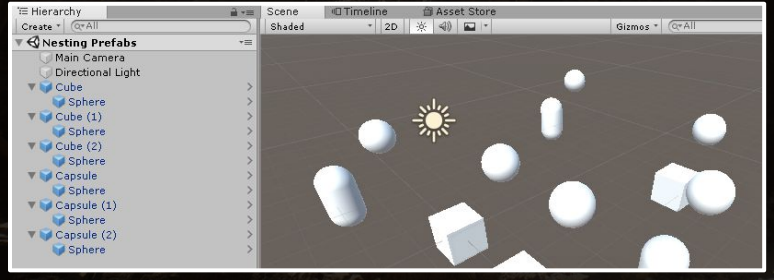


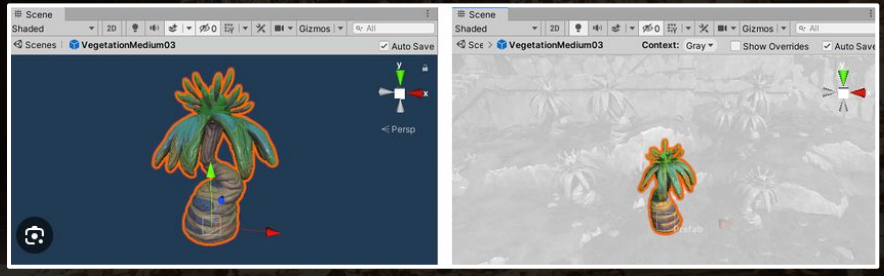
Unity Console



Unity Template








Live demonstration


? Questions ?