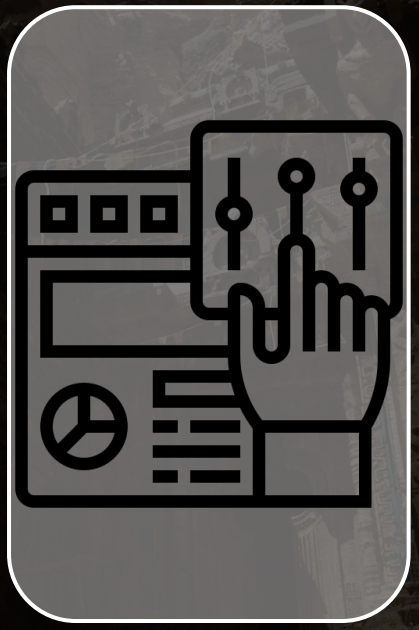
Lesson 6

UI



Nicolas Serf
serf.nicolas@gmail.com

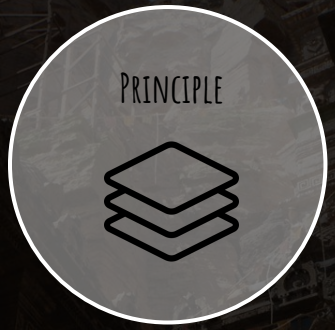
UI

UI

UI System

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used



Definition

An **UI system** is a piece of software which is **part** of every **game engine**. Basically, it offers the **environment**, the **tools**, the **premade widget**, etc... that are used to **create interfaces**

Critical

Even if from the outside, **interfaces** seems **secondary** on the **development cycle** of games, and even in some **production** it is the case. Interfaces are actually **critical** both for **performances** and **gameplay**

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used

Definition

UI Toolkit is the **practical implementation** in every game **engine** for **production**. Just like anything, UIs are in essence **OpenGL** or **DirectX calls** to draw in **2D**, it is **not convenient** to be used like so

CSS inspired

Being in **code** or from a **node** based / **visual solution**, **UIs toolkit** are mostly inspired by **CSS** / **HTML style**, with a **hierarchical system**.

Production

Production wise, you'll be using **UI toolkit** in every **fields** of **UIs**, going from creating **new tools** in the **editor**, to developing **HUD** or **UIs** for all the gameplay needs.

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used

Definition

Code is the first wrapper phase which abstract the complexity of OpenGL/DirectX. It will create a layer of code that will be usable by everything afterwards.

Low level

You can consider the code UI being fairly low level, but it will still be higher level than graphical interfaces. You could decide to create new code UI but it is likely never done when using engines

Editor

In most engine, you'll be using the code UI to create new windows, customized the editor etc... Simply because most engine are built using their own code UI wrapper.

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used

Definition

Around the **code UI**, most engines offers a **wrapper UI** implementation. Mostly **exposed** directly from the **editor**, it allows to **create interfaces without** having to **write** a **single** line of **code**.

Designers

Being most likely a **visual tool**, it will allows **designers / UI / UX artists** to be directly **integrating** interfaces on the engine. **UI devs** will be **needed** obviously, but on a **second pass** to link things

Implication

Performances wise, you'll **redirection** while developing interfaces like so, but it is **minimal**.
Functionalities wise, you'll most likely have to **less possibilities** than directly from **code**

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used

Definition

Names can differs from engine to engine, but the principle is the same. A widget is a visual element populating your interface.

Common features

Widget will have specific features, but offers a common interface for things like visibility, being able to be masked, interactions, etc...You have a class hierarchy on the UI system which allows that

UI

Layout

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used

Definition

A **layout** role is **simple** : **Ordering**. From the **hierarchy**, it will ensure that every element being a **child** will be **ordered one** after **another**. You can obviously **parametrize** this ordering.

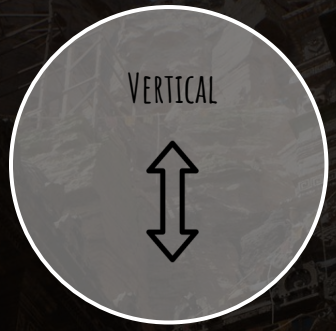
Auto ordering

By default, **elements** which are child of the **layout** will be **ordered automatically**. You don't have **controls**, except the **position** of the **child** in the **hierarchy**.

Controls

Even if you **don't have controls** over the **position**, you can have **control** over **multiple** other things like **width / height** based on the type of layout, the **ratio**, If you want to **ignore** the **layout**, etc...

UI



Definition

Vertical layout is the **most** commonly **used**. You must use it when you want to make elements being **below** each others.

Orientation

The **orientation** of the **vertical layout** makes every element being **one below** the previous one. It makes the **width** of every **element equals** the **width** of the **layout**. You still needs to defines the **height**.

UI

Definition

Horizontal layout has the exact same **functioning** as **vertical** except it makes **elements** being right after another on a **line**.

Orientation

Same **implication** as **vertical**, but this time, all **child elements** will have an **height equals** to the **layout height**. You still needs to define the **width**

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used

Definition

Grid is also a layout but differs from the other ones. It ignores the minimum, preferred, and flexible size properties of its contained layout elements and instead assigns a fixed size

Configuration

There is slightly more configuration to do on a grid if you want to have special cell management.

Simulating

Be aware that you actually can create a grid by having multiple vertical and horizontal layout all nested together. It will be a bit more cumbersome to create, but you'll have a bit more controls

UI

Scrollview

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used

Principle

Adaptative

Optimization

Definition

A **scrollview** is something available in all **engine** and serve a quite **basic idea**. If there is too many **element** inside a **container**, wrap them into a **scrollview** on which we can **navigate** with a **scrollbar**

Internal size

In order to compute the **scrollview**, the **scrollbar**, etc... the **system** needs to take into **consideration** all **child element size**. Afterwards, there is a **position** on the **scrollview** that allows to display **correctly**

UI

Adaptative

Principle

Optimization

Definition

A **scrollview** is **adaptative** by essence, it ensure to **maintains** a **logical state** of the elements to display what **needs** to **display**. That adaptativity is permitted by analyzing every **child** and the **scroll** itself

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used

Optimization

Principle

Adaptative

Definition

On **optimization side**, a **scrollview** can quickly become an **nightmare**, and that's why some **games** experiment some **freeze** or **FPS lost** when things like **complexe inventory** or crafting system are opened

Visibility

If you have **1000 element** in your **scrollview**, only **10** may be **visible** at a time, so on **graphical** side, thanks to **masks**, it will not be a **big issue**, but on **computation** side, it is another story.

Computation

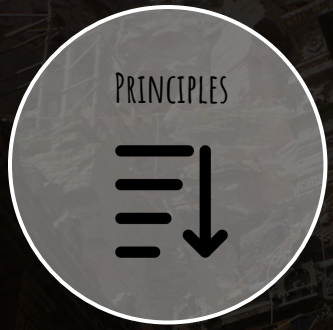
A **scrollview** manage itself the **position** of **every elements** inside the scroll, so it means that even if they are **not visible**, **position** of every elements are **computed** and **analyzed** every time the **scroll move**

UI

Anchors

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used



Definition

In essence, **anchors main functionality** is easy to get : **Maintaining** the **responsivity** of your game on all **resolutions**.

Process

They are used to **determine** the **position** and **scaling** of a UI element **relative** to its **parent**. **Anchors** can be at the **same point** (a single anchor point) or can define an **area** (spread across corners).

Pivots

Pivots on the other **hand** will **controls** the "**center of rotation**" on an **element**.There is **no logic** like anchors with **stretches** or **point**, a **pivot** is always a **simple point**.

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used

Values

Principle

Definition

An anchor are in **relative space**, they have a value between 0 (**left** or **bottom**) and 1 (**right** or **top**). This is the values that are **basically displayed** in every game engine with **draggable handles**.

Example

If you have a **split pivot** into [0.2 , 0.5] and [0.3 , 0.5], it means the **anchors** will always have 0.1 **times** the **screen width** between them. Which will be **directly impacting** the element having that setup.

UI

Most common

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used

Definition

A **button** is an **UI element** you can **interact** with, in order to receive a callbacks. They are **omnipresent** in most UIs which are **interactable**.

Events

There is **various events** on a **button**, like **OnHover**, **OnHoverExit**, **OnClick**, **OnPressed**, **OnReleased**, etc... You must choose carefully which **type** of event you are looking for.

Implementation

That's a **detail** but it is important, you want **buttons** to be **consistent** in a **game**, having the same **reaction** on **hover**, the same **logic** when you **click**, etc... You'll most likely create a **wrapper** about it.

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used

Definition

An **Image** is a **non-interactable** element which is used to **display textures**, **png**, etc...

Ratio

Just like **regular image**, you must be careful when it comes to **ratio** of an image. You can **preserve** an **aspect ratio** in most engine, but if you **scale** it by hand, you'll have a **stretched image**.

Scaling

There is **multiply solution** when it comes to **filling** an **image** element with a **png**. It depends from **engine** to **engine**, but you can have **tiling**, **filling**, **preserving**, etc...

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used

Definition

A **text** is a **non-interactable** element which is used to **display text**, just like the name suggest. You must be **careful** about **wrapping** the text, **overflow**, and other things.

Font

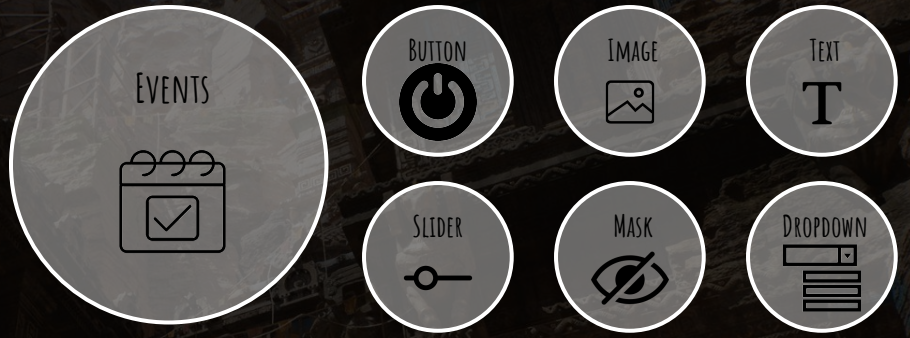
When it comes to **text**, you obviously have **font** that comes into the equation. A font choose is **important** because it has **impact** on the **localization**, if the font **doesn't support stress** for example.

Localization

Obviously, when you want to display **text**, you'll face **localization**, which means being able to **display** a text to various **cultures** and **languages** and **adapt** to it and it comes **numerous challenges**.

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used



Definition

Events are general but it is **worth** mentioning them here. An **event** is something that **appears** in the **editor details** and allows to **connect** whatever you want when they get **triggered**.

Callbacks

Events can be used **everywhere**, but **widgets** use them **intensively** because that's the way they **communicate** when there is an **interaction** on the widget.

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used

Definition

A slider is an **interactable element** that offers the **possibilities** to choose an **numeric value** by having a slider, so a **drag&drop** widget that have a value **associated** with the **position** of the handle.

Min-max

Every **slider** comes with a **min** and **max** value, which are the very **left position** of the **handle**, and the very **right position** of the **handle**.

Wrapper

Even if the **slider** can looks like just a **numeric modification**, you can also transform it in order to have it **non interactable** and **display gauge**.

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used

Definition

A **mask** is an **invisible** and **non-interactable** that allows to ensure that a **widget** will **not** be **visible** outside the **range** of the **mask**. It works with the **hierarchy** once again.

Visibility

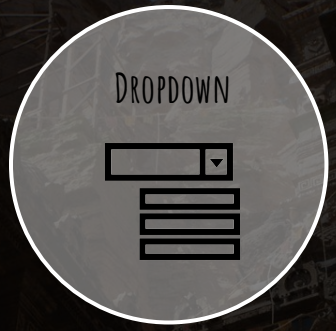
When we **refers** to **visibility**, it's either is the **widget visible** or not, in a global way. Mask ensure that a **widget** can be **partially invisible** if **outside** the **mask area**.

Hierarchy

In most **engine**, **hierarchy** is **important** and **mandatory** for **UIs**, but it is even more with **mask** because they affect **children**.

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used



Definition

A **dropdown** is a **widget** that display one **option** at a time, but can be **interacted** with in order to have a **scrollable list** of **options** from which the **user** can choose **from**.

Event

Just like **button**, **dropdown** has various **callbacks** for the events like : **OnOpen**, **OnSelect** option, etc...**Dropdowns** are often used for **options menu** for example.

Options

Dropdown has a **list** of **option** you can either feed in the **details**, or **build** at **runtime**.

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used

Definition

A **toggle** is a fairly **simple widget**, which act like a power switch. You either **activate** or **deactivate** it.

On/Off

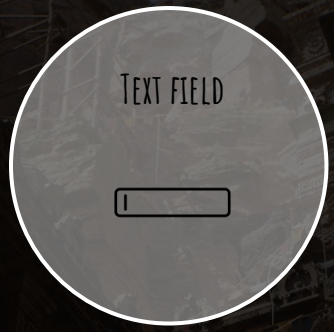
It is **straightforward**, a **toggle** represents a **boolean**, and you'll mostly use them in order to know if a **player** want or not to **activate** something.

Groups

In some engine, you have the **possibility** to have **toggle group**, which allows to have **multiple options** selectable as the **toggle value**, but **only 1 at a time**.

UI

1. UI system
2. Layout
3. Scroll
4. Anchors
5. Most used



Definition

A text field combine a **text** and an **interactable** widget on which you can **write** anything you want. Some engine **allows** to **configure** the **type** of character that can be **inserted** into the text field.

Size limits

You must be **careful** about **forcing** a **size limit** that makes the **interface** looks **coherent**, but also for future use of the text inserted. Think about **SQL injection**, **name displaying**, etc...


Live demonstration


? Questions ?