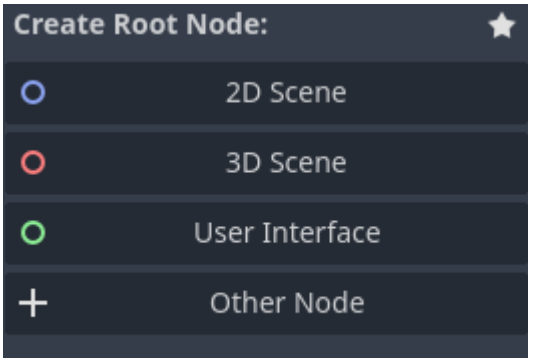
Présentation Godot
Cell 2025-2026

Sommaire :

Les scènes :

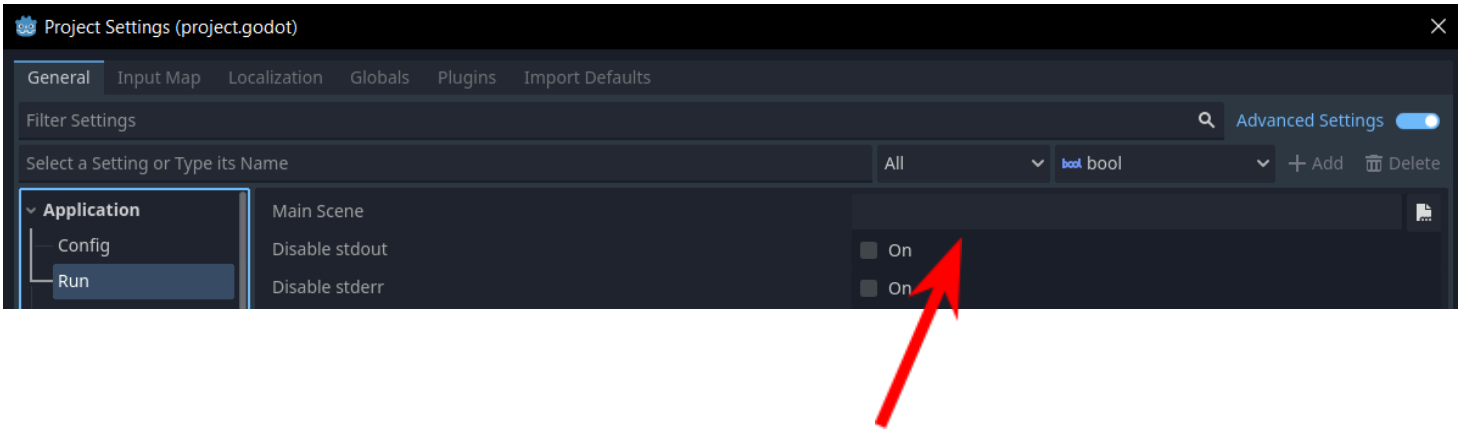
Les scènes sont la base de votre jeu.
Pour lancer le logiciel, Godot va prendre la scène principale et lancer tous les nœuds affiliés, ainsi que leur script.

Une scène est constituée d'un nœud central que l'on choisit quand on l'a créée (avec, de base, les choix Node2D, Node3D et Interface).

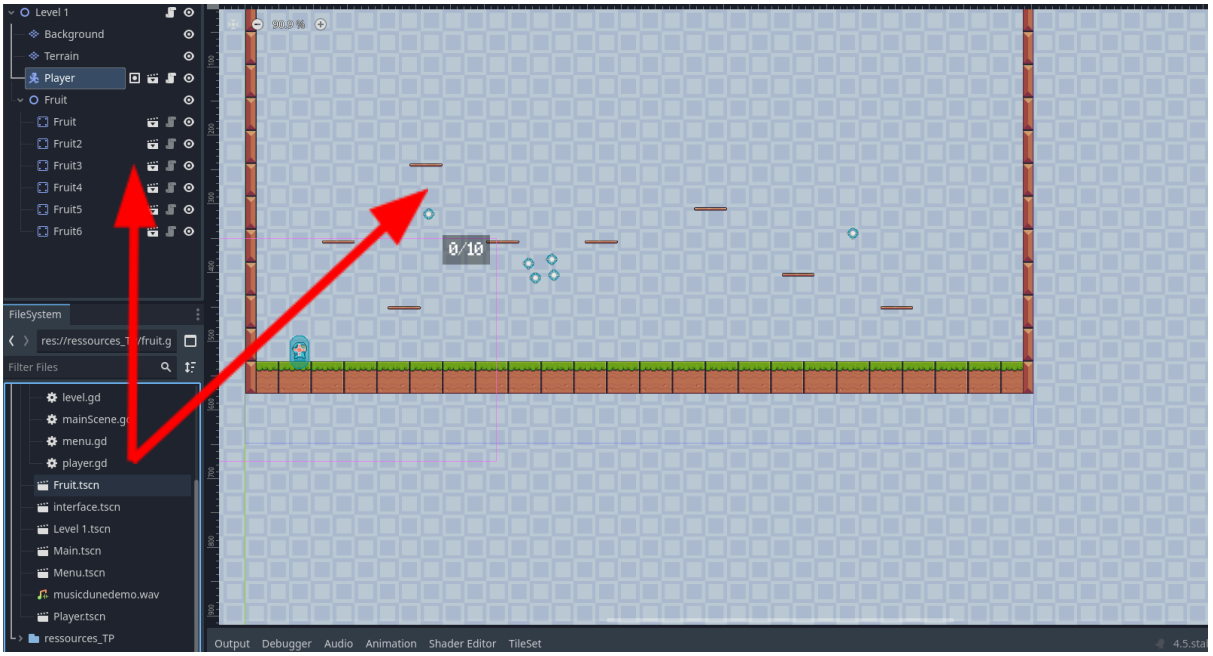


On aura tendance à choisir ces options pour les nœuds où l'action va se dérouler (concrètement : les niveaux, les menus ou encore le nœud central).
Et pour les scènes qui représentent des éléments précis de votre jeu (le jour, un PNJ, un objet...), on prendra la scène correspondant à ce que l'on veut (détail des nœuds plus bas, sinon consulter la doc ou des tutos).

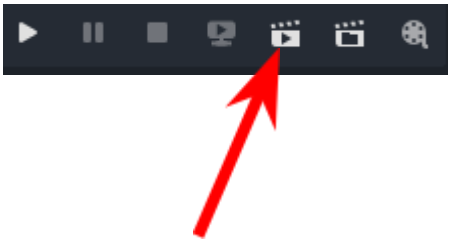
Si jamais vous voulez changer la scène principale, il faut aller dans les paramètres du projet et modifier l'option :
General → Run → Main Scene



Une scène peut être intégrée à une autre scène. Pour cela, il suffit de faire glisser la scène depuis vos fichiers en bas à gauche vers soit l'écran où vous manipulez les différents éléments, soit la liste des nœuds à gauche.



Si vous voulez lancer une scène précise, il faut cliquer sur le clap avec un triangle dessus, en haut à droite, ou appuyer sur **F6**.



Les nœuds:

L'unité de base dans Godot, ce sont les nœuds : ils représentent à peu près tout ce que vous pouvez imaginer.

Les nœuds fonctionnent selon un principe d'arbre. C'est-à-dire qu'il y a le nœud principal de votre scène, et ce nœud aura des enfants. Ces enfants peuvent aussi avoir des enfants...
Un enfant peut également être le nœud principal d'une autre scène (ce qui s'intégrera entièrement dans la scène que vous fabriquez).

En plus de cela, il y a 4 symboles possibles à côté des nœuds : 

- **L'œil** : il sera toujours présent et indique si le nœud est actif dans la scène (ouvert → oui, fermé → non).

- **Le parchemin** : il indique qu'un script (autrement dit un code) est attaché au nœud.

- **Le clap** : il indique que le nœud est l'importation d'une autre scène.

- **Le carré avec un rond** : il indique que le nœud appartient à un groupe (je ne détaillerai pas ce point, mais sachez que ça existe).

Godot est constitué de BEAUCOUP de nœuds : des utiles, des vitaux, des occasionnels et des inutiles (ce n'est pas une blague).
Il y en a pour tous les goûts et toutes les utilités. Je vais présenter les plus utilisés en détail, et faire un listing d'un certain nombre utiles, que je vous laisserai regarder par vous-même.

Les 3 noeuds principaux :

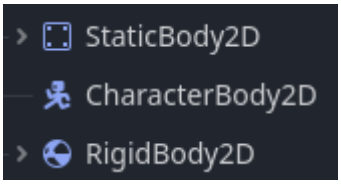
Les trois nœuds principaux et généraux sont :
- Node 2D
- Node 3D
- User Interface

Ces trois nœuds n'ont rien de particulier, ils servent juste à indiquer que la scène concerne cette catégorie, ou à regrouper un ensemble d'autres nœuds.

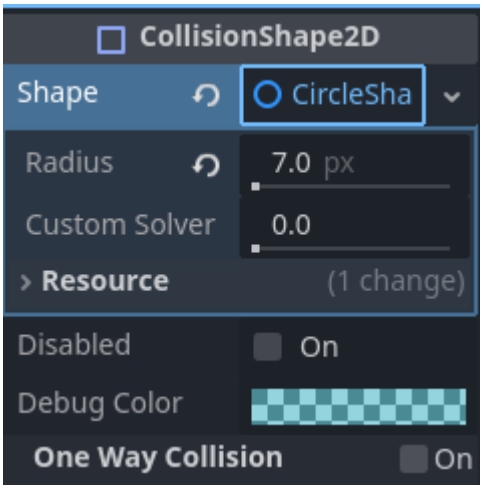
Les différents nœudsphysiques 2D :

Il y a trois nœuds pour représenter des objets physiques :

- **StaticBody2D** : va être utilisé pour les objets immobiles, par exemple un objectif qui ne doit pas bouger, un obstacle immobile ou encore un mur.
- **CharacterBody2D** : représente les éléments de votre jeu qui vont se déplacer. Typiquement : un ennemi, un PNJ ou encore le joueur.
- **RigidBody2D** : représente les objets qui vont avoir une physique : une caisse, un projectile...



Ces nœuds ont aussi besoin d'un **RigidBody2D**, qui correspond à la hitbox. Pour cela, il faut configurer la zone que vous souhaitez.



Les nœuds de sprites :

Il y a deux nœuds de sprite qui vont nous intéresser :

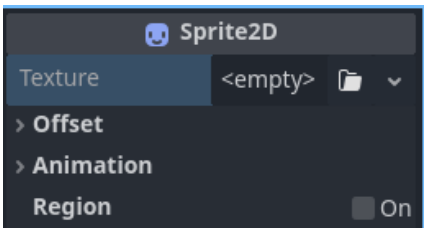
- **Sprite2D** : représente une image, donc utile pour les objets qui ne vont pas bouger, ou dont les changements ne correspondent pas à des animations..

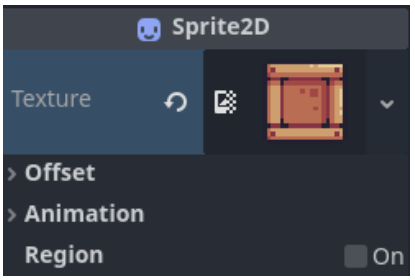
- **AnimatedSprite2D** : représente les animations de vos objets. Vous pouvez aussi l'utiliser pour des objets sans animation, mais c'est un peu inutile, surtout que c'est plus complexe à utiliser.

Utiliser Sprite 2D :

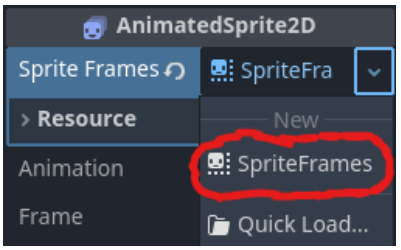


Il suffit de faire glisser votre image dans **Texture** et le tour est joué. Vous pouvez aussi sélectionner votre fichier avec l'icône de dossier.

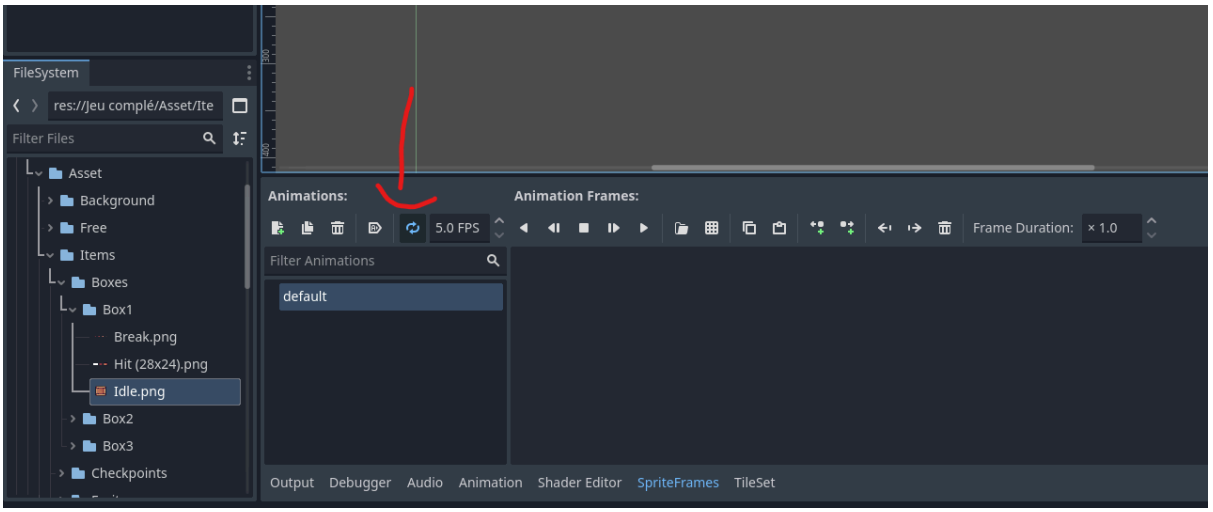


Utiliser Animated Sprite 2D :

Comme dit précédemment, c'est un peu plus complexe.
Tout d'abord, il faut sélectionner **SpriteFrames** dans **Sprite Frames**.

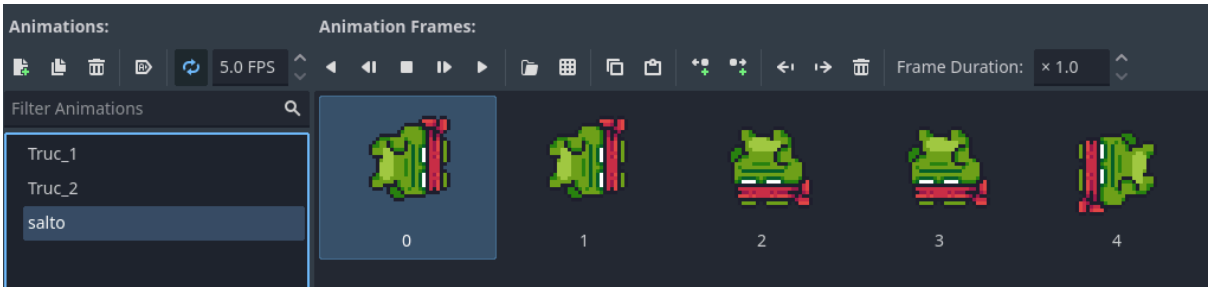


Recliquez dessus, ça devrait vous ouvrir une fenêtre en bas de votre écran :



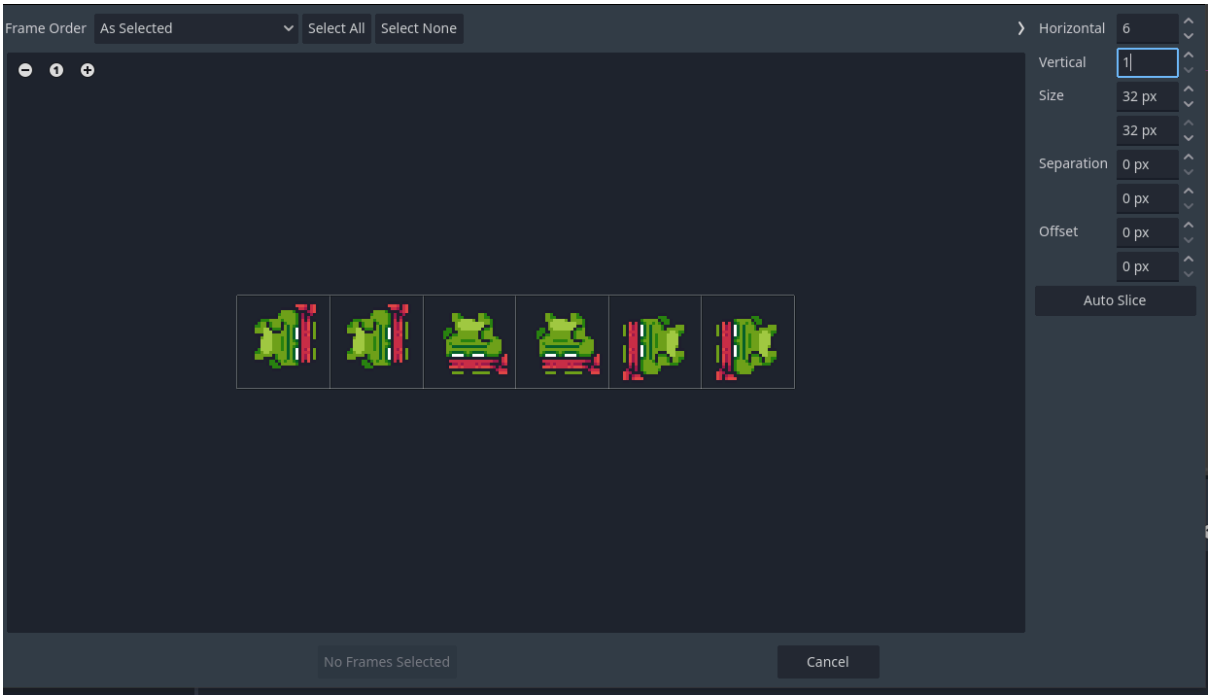
Avant de détailler les outils, je vais expliquer comment mettre l'animation :

Voilà à quoi ça ressemble avec quelques animations.



Avant de détailler tous les outils, je vais expliquer comment importer votre animation :
Pour cela, sélectionnez le quadrillage 3x3 entre le dossier et le symbole « copier ».
Cela va vous ouvrir une nouvelle fenêtre où vous pourrez sélectionner votre spritesheet d'animation.

Une fois que vous avez sélectionné votre image spritesheet, une nouvelle fenêtre va s'ouvrir :



Paramétrez la découpe de votre quadrillage avec les outils **Horizontal**, **Vertical**, ainsi que **Size**, **Séparation**...

Une fois tout cela fait, vous pouvez valider, et les différentes images de l'animation devraient apparaître dans votre menu (comme sur l'image plus haut).

Maintenant, les différents outils :

- La feuille avec le plus (le premier) permet d'ajouter une animation.

- Le deuxième et le troisième permettent respectivement de dupliquer et supprimer une animation.

- La flèche avec un A (quatrième) permet de lancer l'animation dès le lancement de la scène. Si aucune animation n'est là, l'animation sélectionnée en dernier va se jouer (pas l'idéal).

- Les deux flèches qui s'entremêlent permettent de répéter l'animation. Cette option est activée par défaut, et la désactiver va stopper l'animation sur la dernière frame après qu'elle s'est jouée.

- Enfin, **FPS** représente le nombre d'images par seconde.

Je vous laisse découvrir les autres outils, ce document est déjà suffisamment chiant à écrire.

Les nœuds d'interface :

Il y a différents nœuds qui permettent au joueur d'interagir.
Cela peut être pour les menus, les options ou juste pour les principes de votre jeu.

Je vous laisse découvrir les différents nœuds liés à l'interface. Cependant, il y a quand même une différence avec les autres nœuds, c'est ce menu :

Il y a différents nœuds qui permettent au joueur d'interagir.
Cela peut être pour les menus, les options ou juste pour les principes de votre jeu.

Je vous laisse découvrir les différents nœuds liés à l'interface. Cependant, il y a quand même une différence avec les autres nœuds, c'est ce menu :

Les nœuds généraux :

En plus de tous ces nœuds, il y en a d'autres qui ne sont pas toujours affiliés à des catégories, mais qui sont quand même utiles, voire essentiels :

Caméra2D : permet de repositionner la caméra par défaut de la scène. Il est pratique pour suivre un joueur.

Timer : permet d'envoyer des signaux au bout d'un certain temps après leur lancement. Il y a une option pour le lancer directement et pour qu'il se relance juste après qu'il est terminé.

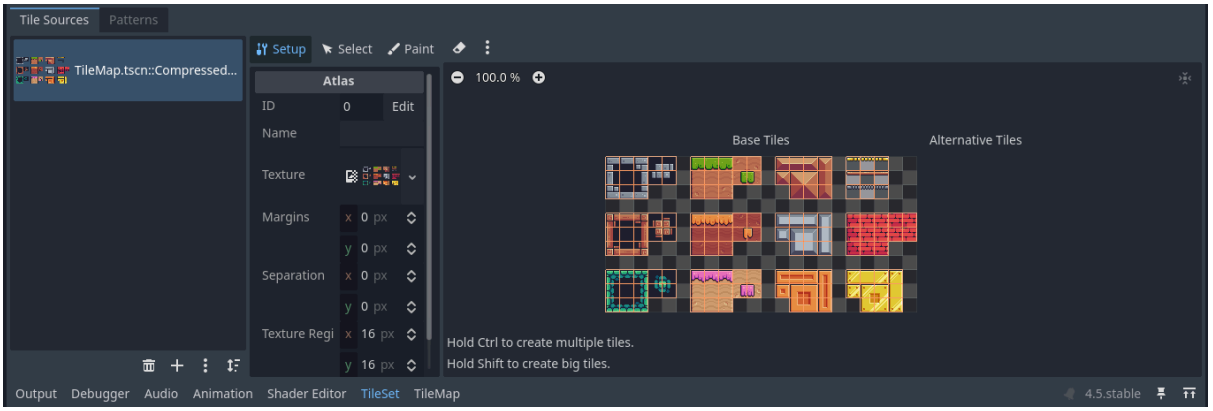
AudioStreamPlayer : la musique... Je n'ai rien à dire de plus.

Les Tiles Map :

Nous avons vu les objets et personnages, les bases, les graphismes... Mais il reste beaucoup d'aspects. Et sans tous les énumérer (ayez pitié de moi, s'il-vous-plaît), un aspect qui se fait très bien en Godot est la construction de niveaux.

Pour ce faire, on utilise un élément appelé les TileMaps, et pour être précis, le nœud **TileMapLayer2D**.

Comme pour les animations, avant de commencer à s'amuser avec ses tiles, il faut les importer.
Dans le menu à droite, choisissez **TileSet** dans l'option **Tileset** et recliquez pour qu'un menu apparaisse en bas.



Dans ce menu, il y a deux onglets qui nous intéressent (les onglets du bas) :

- **TileSet** : qui permet de configurer.
- **TileMap** : qui permet de poser.

TileSet :

À partir de là, vous pouvez sélectionner vos tiles pour indiquer quels blocs sont vos tiles. Vous pouvez les sélectionner individuellement, ou les regrouper avec **Shift + clic** enfoncé sur les tiles que vous voulez regrouper.

Chaque tile est reconnu par ses coordonnées sur la grille (utile si vous les modifiez/générez dans votre code).

Margins : permet d'avoir une marge, si jamais votre image est décalée.
Separation : met une séparation entre vos différents blocs.
Texture_region_side : indique la taille de vos tiles.

Une option très intéressante est '**paint**'.
Cette option vous permet d'ajouter un certain nombre de propriétés très utiles, comme par exemple des couches physiques (importantes pour faire du sol ou des murs) ou encore les connecter pour que Godot soit capable de faire des chemins tout seul (et c'est stylé). Je ne détaille pas plus, si ça vous intéresse, je vous laisse aller voir un tuto (j'ai déjà passé beaucoup trop de temps sur ce document).

TileMap :

C'est ce qui vous permet de poser les tiles. Sélectionnez le pinceau, le ou les tiles que vous voulez, et vous pouvez poser les tiles où vous voulez sur la scène.

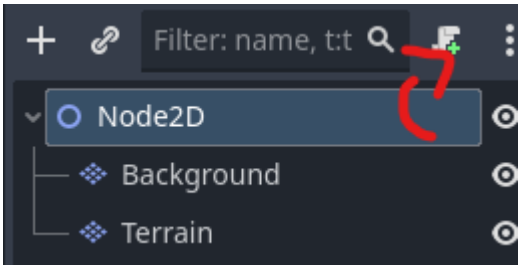
D'autres nœuds que vous pouvez aller regarder :

Voici quelques nœuds qui sont aussi intéressants (que je connais plus ou moins, ou pas du tout) :

- **Skeleton**
- **Path / Navigation**
- Certains nœuds de contrôle, comme **Label**, **LineEdit**, **Button**, **ColorRect** et les différents nœuds de **Range**
- Tous les nœuds 3D

Les scripts :

Les scripts correspondent au code.
Il n'y a pas besoin de coder tout de A à Z, mais à un moment donné, il faut coder.



Pour ajouter un script, appuyez sur le parchemin avec un plus. Un menu s'ouvrira pour que vous sélectionniez le nom, le dossier parent et aussi si vous voulez un modèle préconstruit (ce qui peut être utile dans ce cas-là).

Après, il s'agit de code tout ce qu'il y a de plus classique, je ne vais pas m'étendre dessus. Si jamais vous avez des questions, il suffit de regarder la doc bien fournie de Godot ou des vidéos sur YouTube.

Je vais quand même m'attarder sur quelques points :

Système de classe :

Comme beaucoup de langages de programmation, Godot offre la possibilité de créer ses propres classes.

Cela permet qu'un script n'ait pas besoin d'être attaché à un nœud pour exister, et qu'un script peut aussi être attaché à plusieurs nœuds en même temps.

Le système de classes fonctionne comme dans n'importe quel langage, donc je ne vais pas le développer.

Juste pour indiquer :
Pour créer une classe, qui représentera l'ensemble du fichier, il faut ajouter la ligne :
class_name Fruit [StaticBody2D]

et pour utiliser une classe, il faut utiliser la ligne : *extends [StaticBody2D]*

Signal :

Pour communiquer entre les différents nœuds, Godot utilise le principe de signal.
Ce principe est aussi très similaire à ce que peut faire un programme classique.

Certains nœuds disposent par défaut de signaux en lien avec le nœud. C'est le cas du bouton, qui peut envoyer un signal quand il est pressé, ou d'**Area**, qui peut détecter les entrées.

Il est aussi possible de créer son propre signal dans le code.
Pour cela, il faut d'abord créer une variable en début de code avec la ligne :
signal [nom_du_signal]

Puis, à l'endroit du code où vous voulez envoyer votre signal, entrer la ligne :
[nom_du_signal].emit([possible(s)variable(s)])

Intégration d'une scène à partir de code :

Il est possible que vous ayez besoin d'une scène à un moment précis dans votre jeu, puis que vous puissiez l'enlever.
Pour cela, il n'y a pas besoin d'ajouter la scène depuis l'interface et de l'activer quand vous en avez besoin.
En effet, Godot vous permet, avec le code, de charger une scène, puis de l'enlever. De plus, pour optimiser votre programme, vous pouvez aussi la précharger et l'ajouter uniquement au moment voulu.
Vous pouvez aussi changer la scène, mais je vous déconseille cela, et vous conseille plutôt d'utiliser une scène principale à laquelle vous ajoutez les plus petites scènes.

Commande sympa :

Précharger :
var [scène]_preload = preload("[chemin de la scène (vous pouvez la faire glisser depuis vos fichiers)]")

Charger :
[scène] = [scène]_preload.instantiate()

l'ajouter :
addchild([scène])

File du TP :

Le TP Godot est composé de plusieurs étapes, chacune explorant des éléments de Godot.

Pour récupérer le nécessaire au TP, allez sur ce lien et téléchargez les fichiers affiliés :
https://github.com/JasonJ13/Demo-Cell

Vous trouverez aussi un dossier avec le jeu complet tel que je l'ai créé (je n'estime pas que ce soit parfait, mais ça vous aidera si vous êtes bloqués).

Première étape :

Tout d'abord, il faut créer des scènes et des nœuds.
Pour commencer, créons une scène pour le niveau, et une scène pour le personnage du joueur :
Le niveau ne représente pas un élément concret ni un nœud précis à affilier.
Le personnage, par contre, possède un type de nœud recherché.

Pour chacune de ces scènes, nous voulons un ensemble de nœuds :

Pour le niveau :

- Les différentes tiles (disponibles dans les ressources, nous nous occuperons de créer le niveau un peu plus tard)
- Le nœud de scène du personnage du joueur

Pour le personnage :

- Une hitbox
- Une animation
- Une caméra (je recommande un zoom x2)

Deuxième étape :

Maintenant qu'on a notre personnage et notre niveau, nous pouvons lancer la scène. Seul problème : nous ne pouvons pas faire bouger notre personnage.

Pour ce faire, nous allons ajouter un script à notre personnage. Pas besoin de créer de zéro, le modèle offert par **CharacterBody2D** est largement suffisant pour un début.

Il va falloir rajouter une animation pour le déplacement, je vous laisse faire avec les indications données plus haut.

Parfait, maintenant, nous avons notre personnage qui peut se mouvoir dans son niveau.

Troisième étape :

On peut se déplacer, c'est bien, mais il faudrait un objectif dans le niveau.

L'objectif va être de récupérer des fruits, mais cela implique, en plus de créer une nouvelle scène, de modifier le personnage pour qu'il puisse les récupérer.

Pour la partie personnage, il vous faudra utiliser les signaux.

Je vous recommande d'essayer de faire le fruit par vous-même, mais sinon, la scène existe déjà.

Une fois tout cela fait, n'oubliez pas d'ajouter les fruits dans la scène.

Quatrième étape :

On va pouvoir créer nos propres niveaux maintenant.
Pour cela, créez une nouvelle scène et utilisez les tiles à votre guise pour créer (soit en copiant et dessinant avec celle de l'ancien, soit, comme je vous le recommande, de zéro).

N'oubliez pas d'ajouter les fruits et le joueur, puis vous pouvez lancer la scène.

Idée supplémentaire :

Vous connaissez toutes les bases de Godot et avez même une petite base pour un jeu.

Il reste cependant beaucoup de choses que vous pouvez faire, voilà quelques idées :

- La transition entre les niveaux une fois que tous les fruits sont récupérés
- Des menus pour votre jeu (pourquoi se contenter d'une scène avec un bouton « play » ?)
- Une interface qui indique combien de fruits il reste
- Un système de double saut pour le joueur
- L'ajout de sons et bruitages
- Un système de wall jump
- À peu près tout ce que vous pouvez imaginer

Annexe :

sprites utiliser pour le TP : Pixel adventure,
https://pixelfrog-assets.itch.io/pixel-adventure-1

Chaîne youtube pratique :
https://www.youtube.com/@dev-worm/videos