

Développer des jeux

avec Godot 4 et le langage C#

par
Anthony Cardinale
avec la contribution de Julian Murgia

Cet extrait provient du livre *Développer des jeux avec Godot 4 et le langage C#*
écrit par Anthony Cardinale – © 2023 Editions D-Booker

Développer des jeux • avec Godot 4 et le langage C#
par Anthony Cardinale, avec la contribution de Julian Murgia

ISBN (imprimé) : 978-2-8227-1109-8

Copyright © 2023 Éditions D-Booker
Tous droits réservés

Conformément au Code de la propriété intellectuelle, seules les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective ainsi que les analyses et les courtes citations dans un but d'exemple et d'illustration sont autorisées. Tout autre représentation ou reproduction, qu'elle soit intégrale ou partielle, requiert expressément le consentement de l'éditeur (art L 122-4, L 122-5 2 et 3a).

Publié par les Éditions D-Booker, 229 rue Solférino, 59000 Lille
www.d-booker.fr
contact@d-booker.fr

Les exemples (téléchargeables ou non), sauf indication contraire, sont propriété des auteurs.

Conception de la couverture : D'après une création de Marie Van Der Marlière
(www.marie-graphiste.com)
Logo de Godot Engine : © Andrea Calabró (Creative Commons Attribution License [CC-BY 3.0])
Mise en page : Générée sous Calenco avec des XSLT développées par la société NeoDoc (www.neodoc.biz)
Diffusion : AFPU-D (<https://www.afpu-diffusion.fr>)
Distribution : DILISCO (<https://dilisco-diffusion.centprod.com>)

Dépôt légal : Sept. 2023
Édition : 2
Version : 2.0

Table des matières

Avant-propos	vii
1. Qu'allons-nous voir dans ce livre ?	vii
2. Crédits photos	x
3. Sources des exemples	x
4. Terminologie en anglais et interface française	x
5. URL raccourcies	xi

Notions fondamentales 1

1. Découverte de Godot Engine	3
1.1. Téléchargement de Godot	5
1.2. L'interface de Godot	6
1.3. Modification de l'interface	16
1.4. Nœuds et arbres	17
2. Création d'une scène sous Godot	19
2.1. Organisation du projet	20
2.2. Import des textures	21
2.3. Création d'une balle physique	24
3. L'instanciation avec Godot	35
3.1. Création de la plateforme	35
3.2. Création de la scène principale	38
3.3. Instanciation de scènes dans Godot	39
3.4. Lancement du jeu	41
4. Initiation à la création de scripts avec Godot	43
4.1. Premier script	44
4.2. Indentation	48
4.3. Variables et constantes	48
4.4. Commentaires	49
4.5. Conditions	49
4.6. Afficher du texte dans la console	50
5. Plus loin avec C#	51
5.1. Création d'un bouton	51
5.2. Récupérer un texte saisi par l'utilisateur	55
6. Interview de Gilles Roudière, contributeur au projet Godot	57

Développement d'un jeu 2D	63
7. Mise en place du projet	65
7.1. Création du projet	65
7.2. Trouver des ressources pour son jeu	66
7.3. Importer les ressources	68
7.4. Paramètres du projet	71
7.5. Système de coordonnées 2D	72
8. Création du personnage joueur	73
8.1. Création du nœud joueur	73
8.2. Création du script du personnage	76
9. Animation du personnage	87
9.1. Création des animations	88
9.2. Programmation des animations	92
10. Création d'une plateforme 2D	95
10.1. Création d'une nouvelle scène	95
10.2. Création de la plateforme	96
10.3. Programmation de la gravité	98
10.4. Programmation du saut	101
11. Mise en place d'un TileSet	103
11.1. Création des tuiles	105
11.2. Solidité des tuiles	108
12. Conception d'un niveau du jeu	113
12.1. Création de la structure de notre niveau	113
12.2. Mise en place de la caméra	115
12.3. Création d'un arrière-plan en parallaxe	117
13. Interaction avec les objets	123
13.1. Création d'une pierre précieuse	123
13.2. Création du script de ramassage	126
13.3. Test de notre script	132
14. Création de l'interface utilisateur	135
14.1. Mise en place de l'interface	136
14.2. Programmation de l'interface	142
15. Ajout des ennemis	145

15.1. Création d'un ennemi statique	145
15.2. Création d'un ennemi mouvant	148
16. Musique et effets sonores	155
16.1. Musique d'ambiance	155
16.2. Effets sonores	158
17. Finalisation de notre jeu	163
17.1. Création du menu principal	163
17.2. Script du menu principal	167
17.3. Compilation du projet	169
Création d'un jeu 3D	175
18. Modélisation 3D : Initiation à Blender	177
18.1. Téléchargement de Blender	178
18.2. L'interface de Blender	179
18.3. Les raccourcis et outils indispensables	180
Création rapide	182
Changer de mode	183
Points, arêtes, faces	183
Les transformations	184
Mode Transparence	186
Couper le modèle 3D	187
Outils de sélection	188
Extrusion	190
Biseau	190
Récapitulatif	191
19. Modélisation 3D du niveau de notre jeu.....	193
19.1. Création des zones	193
19.2. Alignement des deux zones	199
19.3. Création des couloirs	201
19.4. Isoler des éléments	206
19.5. Exporter pour Godot	208
20. Mise en place du projet Godot	209
20.1. Importation de la modélisation	210
20.2. Coloration du niveau	211
20.3. Ajout de la physique	215
20.4. Création de la scène du premier niveau	218

21. Création de la balle	221
21.1. Création de l'objet	221
21.2. Préparation des collisions	227
21.3. Instanciation de la balle et redimensionnement	229
21.4. Ajout d'une caméra	232
22. Anticrénelage, éclairage et post-processing	235
22.1. Anticrénelage	235
22.2. Éclairage	237
22.3. Post-traitement	241
23. Déplacement de la balle	249
23.1. Script de la balle	249
23.2. Script de la caméra	256
24. Déclenchement de la fin du niveau	259
24.1. Détecter la fin du niveau	260
24.2. Gérer la chute dans le vide	264
25. Quelques objets à ramasser	267
25.1. Création d'un objet à ramasser	267
25.2. Script de collision	271
26. L'interface utilisateur	277
26.1. Affichage du compteur de cubes et du temps	277
26.2. Programmation du compteur	279
26.3. Programmation du chronomètre	281
27. Finalisation et publication du jeu	285
27.1. Émettre un son quand un objet est ramassé	285
27.2. Compilation du jeu	289
Aller plus loin avec Godot	293
28. Les pistes pour s'améliorer	295
Concepts et termes anglais	299
Index	303
À propos des auteurs	309

1

Découverte de Godot Engine

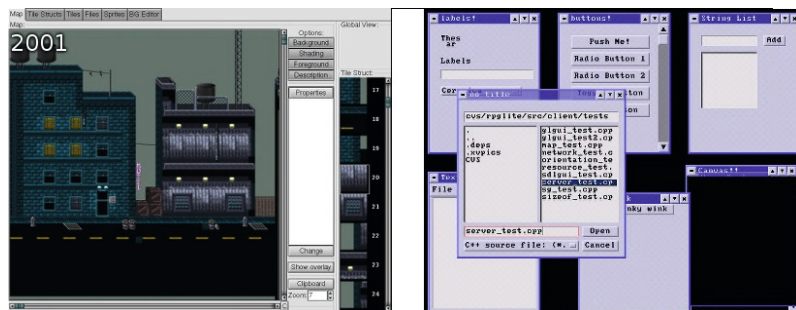
Au début des années 2000, les moteurs de jeux tels qu'on les connaît aujourd'hui n'existent pas, du moins pas sur le marché. Les seuls pseudo-moteurs existant ne sont que des outils très spécifiques qui, pour beaucoup, ne servent qu'à créer des jeux de tir à la première personne (FPS) tel que Doom.

Note > Doom est le premier FPS à avoir eu un succès planétaire. Suite à ce succès, de nombreux développeurs ont tenté de développer des jeux similaires, ce qui a conduit à créer l'appellation Doom-Like. De nombreux outils de création de FPS ont alors vu le jour.

Pour disposer d'un outil plus généraliste, Juan Linietsky et Ariel Manzur se sont lancés dès 2001 dans le développement d'un moteur de jeu qu'ils appelleront Godot en référence à la pièce de théâtre *En attendant Godot*. Ce nom a été choisi car un tel projet, s'il aboutit, demandera forcément beaucoup de temps de développement.

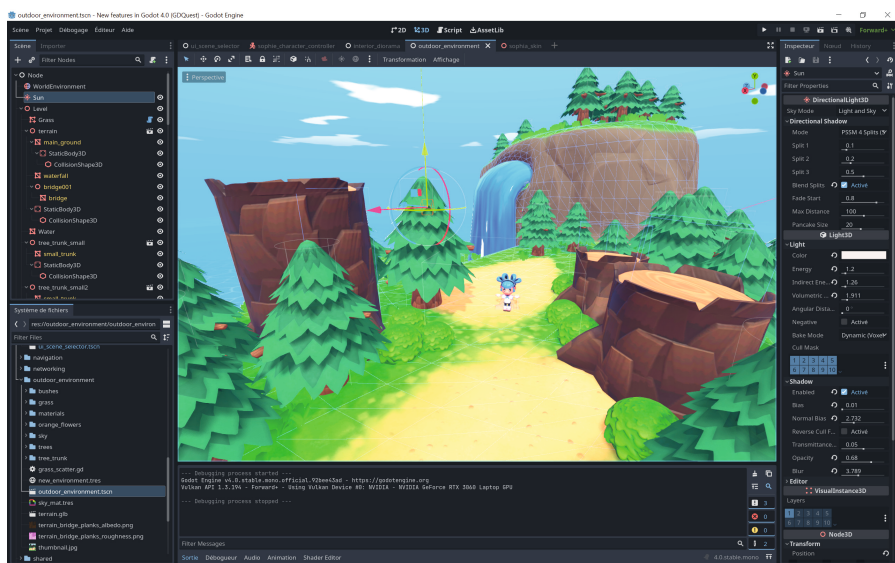
Juan Linietsky et Ariel Manzur avaient pour idée de rendre leur moteur de jeu libre dès le départ mais ils ont préféré le garder dans un premier temps propriétaire jusqu'à ce qu'il devienne mature. Après des années de travail acharné, le moteur gagne petit à petit en maturité comme on peut le voir sur les images partagées par les fondateurs sur leur blog [<https://d-booker.io.my/godot-hist>].

Figure 1.1 : Godot Engine en 2001



C'est en janvier 2014 que Godot devient libre et accessible gratuitement en téléchargement via le site officiel (godotengine.org) ou via la plateforme Steam depuis 2016.

Figure 1.2 : Godot Engine en 2023

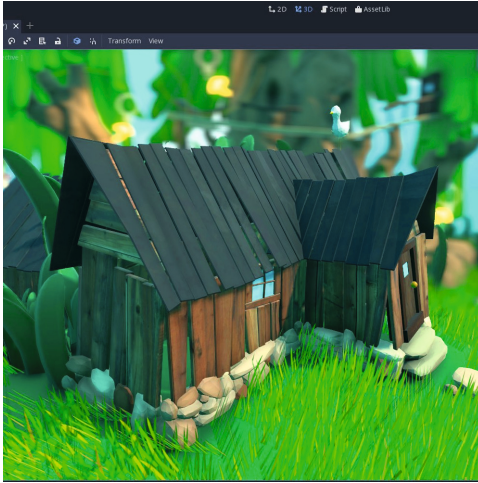


À la différence de la plupart de ses concurrents, Godot Engine a l'avantage d'être totalement gratuit et open-source. D'autres moteurs de jeux, propriétaires, vous imposent d'acheter une licence ou de reverser une partie des revenus générés par vos jeux. Godot, pour sa part, est distribué sous licence libre MIT. Cette licence vous permet de télécharger, utiliser et créer gratuitement avec Godot. Vous pouvez accéder au code source, le modifier et utiliser le logiciel à n'importe quelle finalité. Vous êtes propriétaire de vos créations et vous ne devez rien à personne. Vous pouvez distribuer ou vendre vos jeux librement.

Godot se démarque aussi par sa puissance. Il utilise les dernières techniques de rendu pour proposer des graphismes très poussés (voir Figure 1.3). Il permet de créer des jeux 2D et 3D à destination des principales plateformes (PC, mobiles et consoles).

Depuis 2014, Godot a beaucoup évolué pour devenir un environnement de développement intuitif et performant. L'interface très complète vous permet de paramétrer vos assets (modèles 3D, sons, textures...), de créer vos niveaux, de programmer vos scripts et de tester le tout à la volée. Dans sa version 4, Godot a frappé un grand coup en se rapprochant de ce qui se fait de mieux en matière de moteur de jeu et de simplicité d'utilisation même pour les débutants.

Figure 1.3 : Exemple de rendu 3D sous Godot



Aujourd'hui, Godot est de plus en plus utilisé et tend à devenir un incontournable dans le développement de jeux. Que vous soyez développeur de jeux, entrepreneur ou simplement passionné par le développement de jeux, Godot est un excellent choix pour concevoir des jeux de grande qualité de façon intuitive.

1.1. Téléchargement de Godot

Note > Ce livre est basé sur la version 4.0 du logiciel. Au moment où vous le lirez, de nouvelles versions seront peut-être sorties. Cela ne devrait pas poser de problèmes, les versions 4.1.x comme par exemple 4.1.2 n'impliquent que des corrections de bogues.

Pour commencer, nous allons télécharger Godot. La meilleure façon est de le faire directement via le site officiel [<https://godotengine.org>]. Cliquez ensuite sur le bouton DOWNLOAD et choisissez la version qui correspond à votre système d'exploitation.

Téléchargez la version .NET (compatible avec le langage C#) et non la version standard (uniquement compatible avec GDScript). Ce livre étant basé sur des exemples C#, il vous faudra obligatoirement utiliser cette version .NET. Le C# a été introduit à partir de la version 3 de Godot et est à présent pleinement utilisable. Il permet d'obtenir de meilleures performances.

Une fois le fichier ZIP téléchargé, il ne vous reste plus qu'à l'extraire pour pouvoir accéder à l'exécutable de Godot Engine.

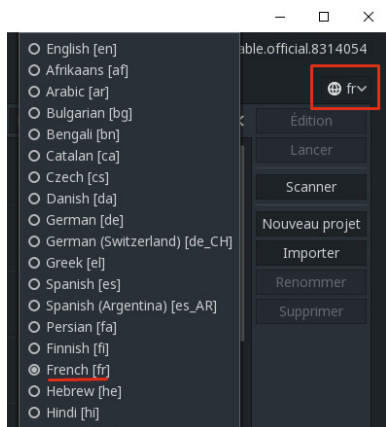
1.2. L'interface de Godot

Comme la plupart des moteurs de jeux évolués, Godot propose une interface de développement complète. Via cette interface, vous pourrez gérer vos ressources comme les modèles 3D, les scripts, les textures, les shaders, les musiques, etc. Vous pourrez également construire vos niveaux de façon visuelle via un éditeur WYSIWYG¹. L'interface vous permet d'écrire vos scripts et de tester votre jeu à la volée directement dans l'éditeur.

Le gestionnaire de projets

La première fenêtre qui s'ouvre lorsque vous lancez Godot est le gestionnaire de projets. Cette fenêtre permet de créer des projets ou d'ouvrir un projet existant. En haut à droite, vous pourrez choisir la langue de l'interface.

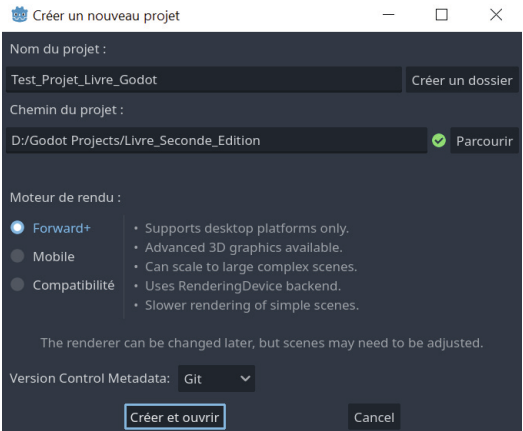
Figure 1.4 : Choix de la langue



Pour créer un nouveau projet, cliquez sur le bouton NOUVEAU PROJET. Ici, vous devrez spécifier un nom pour votre projet et un dossier de destination dans lequel tous les fichiers seront sauvegardés.

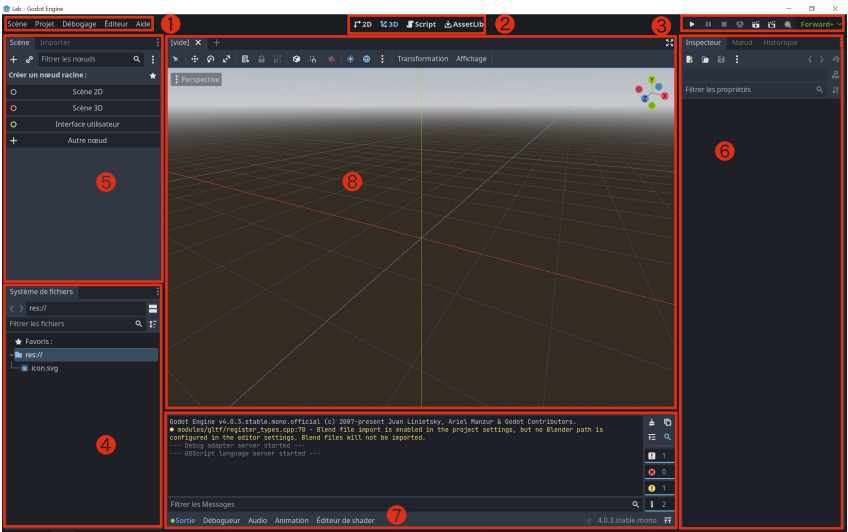
¹What You See Is What You Get, c'est-à-dire Ce que vous voyez est ce que vous obtenez. Il s'agit d'interfaces permettant de créer de façon visuelle.

Figure 1.5 : Création d'un nouveau projet



Une fois les informations saisies, cliquez sur CRÉER ET OUVRIR pour lancer l'éditeur de Godot. Voici à quoi ressemble son interface et comment elle est organisée.

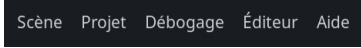
Figure 1.6 : L'interface de Godot



Comme vous pouvez le voir, plusieurs zones se distinguent. Examinons d'un peu plus près à quoi ces zones correspondent.

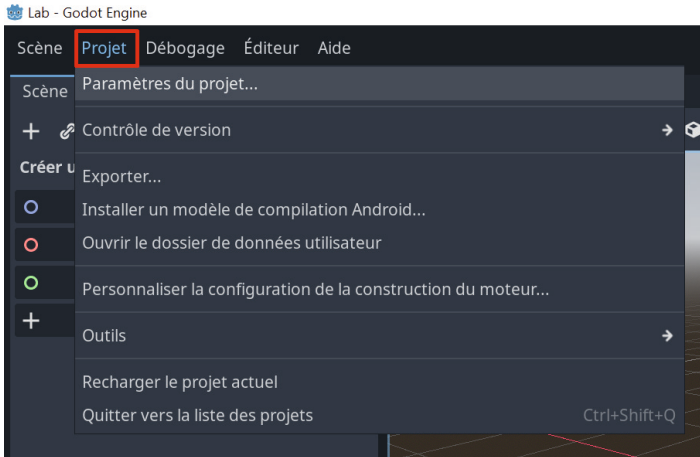
Le menu principal [0]

Figure 1.7 : Menu principal



Le menu principal est très classique : il vous permet de créer une nouvelle scène, de sauvegarder votre scène, d'importer des fichiers, d'accéder aux paramètres du projet, de modifier l'affichage ou encore d'accéder à l'aide du logiciel. Dans notre cas, nous utiliserons principalement ce menu pour accéder aux options du projet via PROJET/PARAMÈTRES DU PROJET. Nous y reviendrons plus tard dans ce livre lorsque nous créerons notre premier jeu.

Figure 1.8 : Utilisation du menu



Les espaces de travail [②]

Figure 1.9 : Workspaces

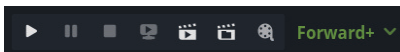


En haut au centre de l'éditeur, vous trouverez les workspaces. Ce menu vous permet d'accéder à un espace de travail. Par défaut nous sommes en 3D, cela permet de travailler dans l'éditeur de niveaux 3D. Vous pouvez passer en 2D pour créer des jeux 2D ou des menus ou encore passer en mode Script pour la programmation. Le dernier bouton permet d'accéder au magasin de ressources où vous pourrez trouver des éléments préconfigurés. Dans un premier temps, nous travaillerons en 2D.

Le menu de playtest [③]

Playtester un jeu consiste à tester une partie de son jeu pour vérifier que le gameplay est fun. Comme vous pouvez le voir, j'utilise beaucoup de mots anglais. Dans le milieu du développement de jeux, certains mots ne se traduisent pas et s'utilisent directement en anglais. Vous trouverez les boutons de playtest en haut à droite de l'interface. Ils permettent de lancer le jeu, de le mettre en pause ou de l'arrêter.

Figure 1.10 : Playtest

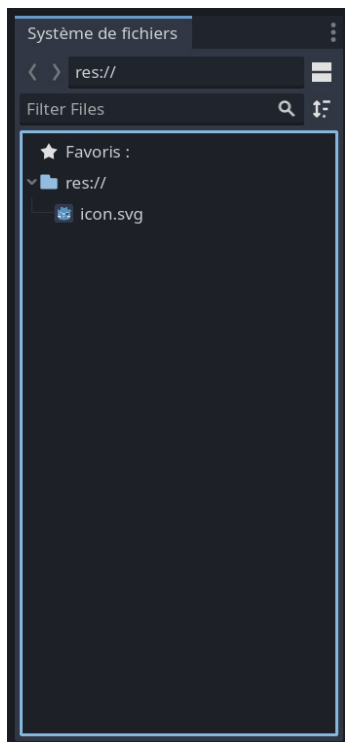


Vous pouvez lancer le jeu à l'aide du bouton PLAY ou via la touche F5. Pour tester votre jeu, vous devez spécifier une scène de démarrage, par exemple son menu principal.

Vous pouvez également tester la scène sur laquelle vous êtes en train de travailler. Pour cela, utilisez la cinquième icône  ou le raccourci F6.

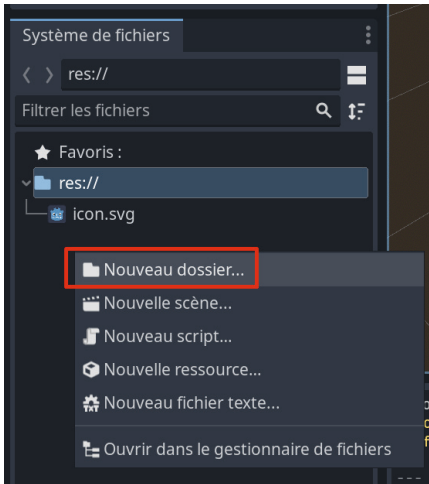
Le système de fichiers [4]

Figure 1.11 : Système de fichiers

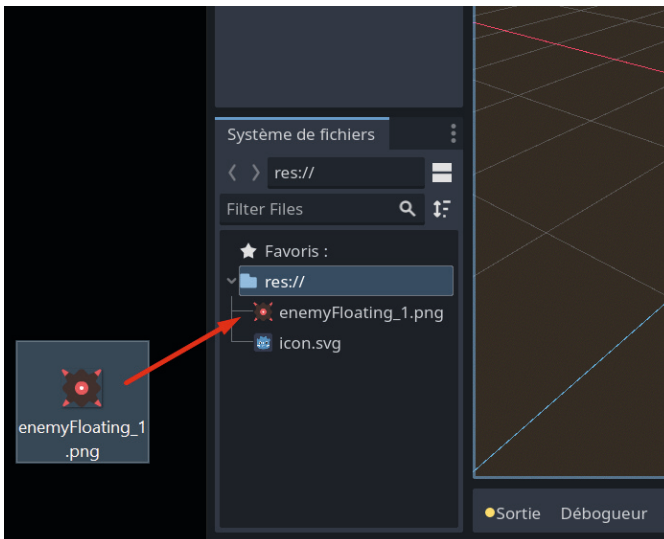


Le système de fichiers liste l'ensemble de vos ressources (textures, sons, scènes, scripts, modèles 3D...) importées. Dès lors qu'un fichier (appelé *asset*) est importé, il est utilisable dans votre projet pour concevoir votre jeu.

Pour créer des dossiers (et vous organiser), cliquez droit et sélectionnez NOUVEAU DOSSIER.

Figure 1.12 : Création d'un dossier

Pour importer des ressources comme des textures vous pouvez simplement les glisser/déposer dans la liste de vos assets ou dans un dossier.

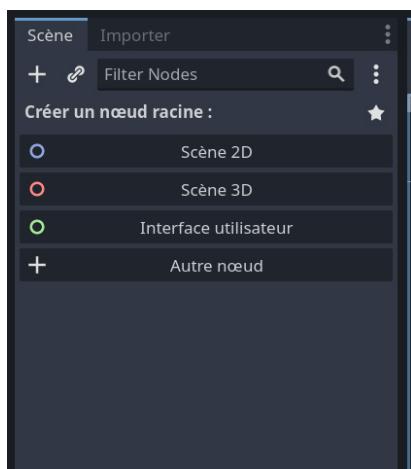
Figure 1.13 : Import d'assets

***Note >** L'image importée n'est qu'un exemple pour illustrer mes propos, il n'est pas nécessaire d'utiliser les mêmes ressources que moi dans ce chapitre. Ici nous nous contentons de découvrir le fonctionnement de base du logiciel.*

Lorsque nous créerons notre premier jeu, nous veillerons à garder une arborescence claire et propre. Nous rangerons toutes nos ressources dans des dossiers organisés.

La scène [⑤]

Figure 1.14 : La scène



En haut à gauche de l'interface, vous avez la scène. C'est ici que vous retrouverez tous les objets que vous avez utilisés dans votre niveau. Par défaut la scène est vide et des raccourcis sont présents pour vous permettre de créer votre premier élément. Via cette fenêtre, vous pourrez organiser votre scène et accéder aux propriétés d'un objet en cliquant dessus. Les propriétés s'affichent dans l'inspecteur.

L'inspecteur (inspector) [O]

Figure 1.15 : L'inspecteur

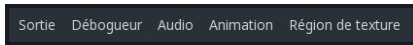


L'inspecteur vous permet d'accéder à toutes les propriétés de l'objet sélectionné. Vous pourrez modifier les propriétés visuelles, les informations de position, rotation, taille et également les animations. Pour l'exemple j'affiche sur ma capture les propriétés d'une image.

Panneau inférieur [0]

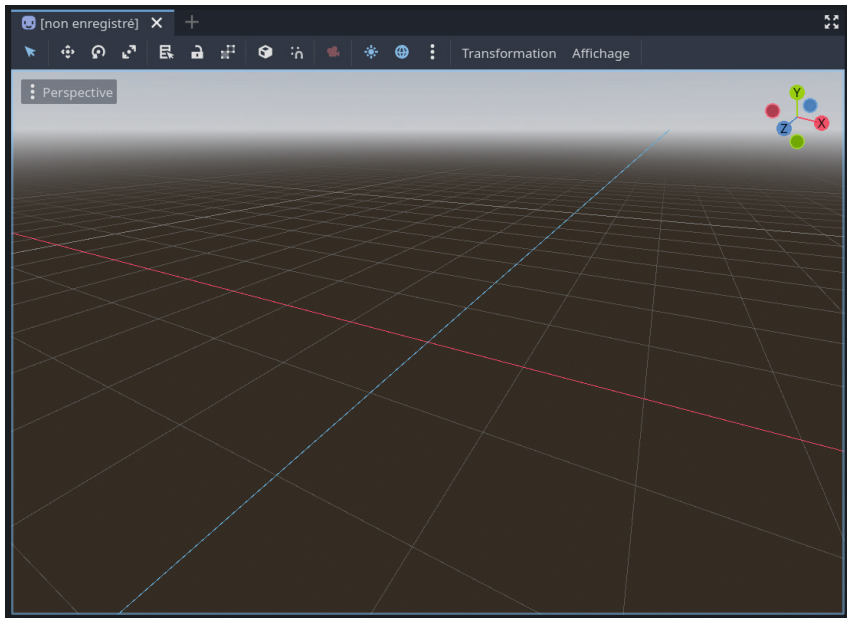
Tout en bas de l'écran se trouve le panneau inférieur. Refermé par défaut, il vous permet d'accéder à de nombreux outils comme le système de mixage audio, le système d'animation mais également au débogueur et à la console qui affichera les données du code et les erreurs.

Figure 1.16 : Panneau inférieur



Fenêtre principale (viewport) [8]

Figure 1.17 : Viewport



Au centre de l'interface se trouve le viewport, la fenêtre principale de Godot. C'est ici que nous travaillerons, que nous positionnerons les éléments décoratifs, les per-

sonnages, afin de créer les différents niveaux de notre jeu. Nous pouvons travailler en 2D et en 3D.

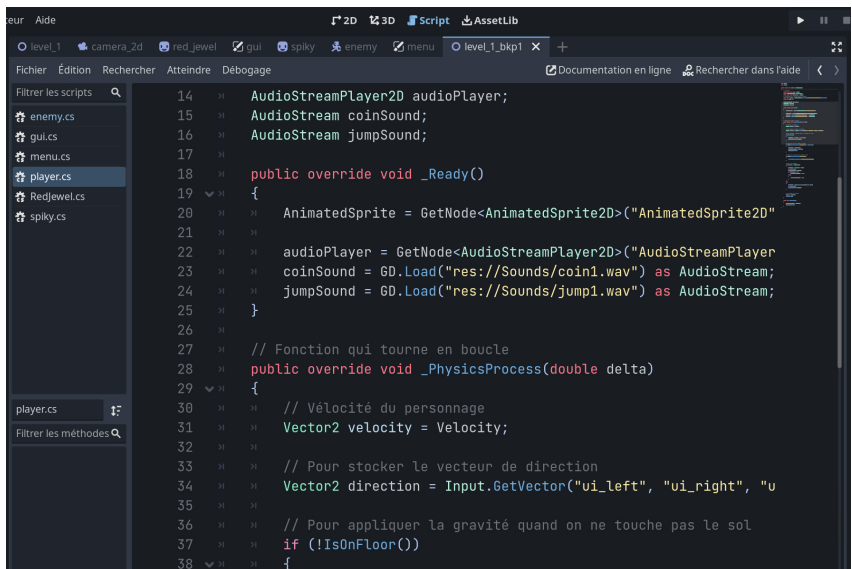
Vous pouvez naviguer dans cette fenêtre avec le clavier et la souris. Avec un clic droit, vous pouvez tourner la caméra (en mode 3D). En mode 2D, appuyez sur le bouton du milieu (la molette) pour déplacer la caméra ; et en mode 3D, maintenez la touche Maj enfoncée pendant que vous appuyez sur la molette.

La molette permet aussi de zoomer/dézoomer.

En haut de cette fenêtre, vous trouverez une barre d'outils permettant d'accéder aux outils de sélection, position, rotation et modification de taille.

C'est également sur cet espace que nous pourrons écrire du code.

Figure 1.18 : Éditeur de code



Nous verrons dans la suite de ce livre d'autres zones de l'interface au fur et à mesure de nos besoins.

1.3. Modification de l'interface

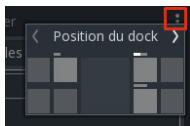
Vous pouvez adapter l'interface à vos habitudes. Les fenêtres peuvent être redimensionnées en attrapant la petite arête entre deux fenêtres.

Figure 1.19 : Agencer l'interface



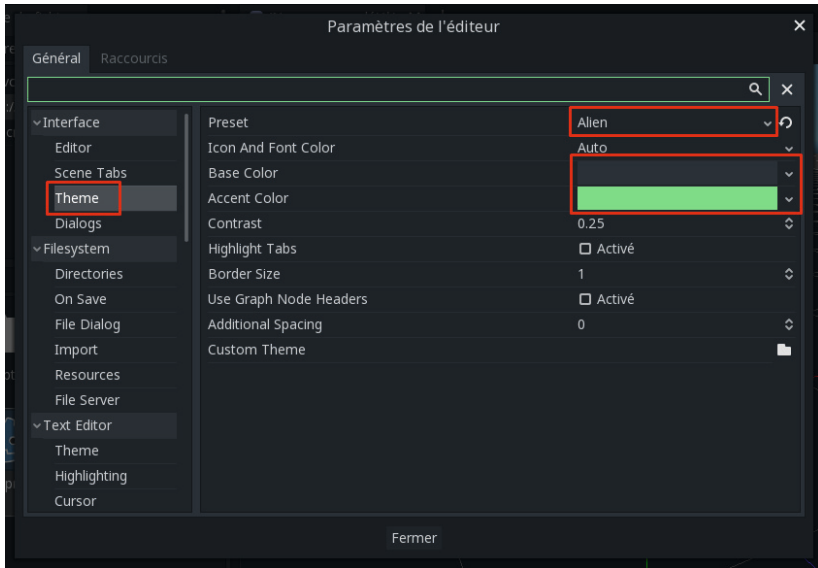
Vous pouvez aussi repositionner les éléments et les ancrer où vous le souhaitez. Pour cela, cliquez sur les points en haut à droite des fenêtres pour accéder à l'outil de position :

Figure 1.20 : Position du dock



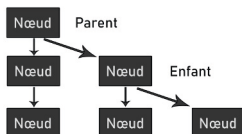
Pour ma part, je travaille avec le système de fichiers et la scène à gauche, l'inspecteur à droite et le viewport au centre.

Vous avez également la possibilité de modifier les couleurs et le thème de Godot. Pour cela, cliquez sur ÉDITEUR/PARAMÈTRES DE L'ÉDITEUR puis cliquez sur THEME et choisissez les couleurs ou le préréglage (*preset*) de votre choix.

Figure 1.21 : Modification du thème

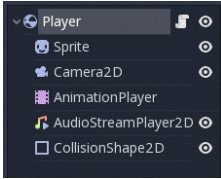
1.4. Nœuds et arbres

Godot est basé sur un système de nœuds [nodes] et d'arbres [tree]. Les nœuds sont les éléments fondamentaux pour la création de jeux. Ils ont un nom, des propriétés et peuvent être organisés afin d'avoir des enfants comme sur la Figure 1.22.

Figure 1.22 : Un arbre composé de nœuds

Lorsqu'ils sont organisés de cette façon, nous appelons cela un arbre (*tree*). Avec Godot vous aurez la possibilité de créer des nœuds, de les organiser, de les assembler afin de créer des jeux élaborés. Voici par exemple à quoi pourrait ressembler un nœud joueur :

Figure 1.23 : Exemple de nœud joueur



Comme vous pouvez le voir, le joueur a un sprite pour son visuel, une caméra pour que le joueur puisse le voir, un système d’animation, un système de son et un objet de collision. C’est typiquement le genre de nœud que nous aurons l’occasion de créer.

Dans le prochain chapitre nous verrons plus en détail comment se déroule la création d’une scène sous Godot.