

LOG725 - Ingénierie et conception de jeux vidéo

Labo 3 - Développement de jeux avec Godot Engine

Gabriel C. Ullmann

École de Technologie Supérieure, Hiver 2024



Le génie pour l'industrie

Objectifs d'apprentissage

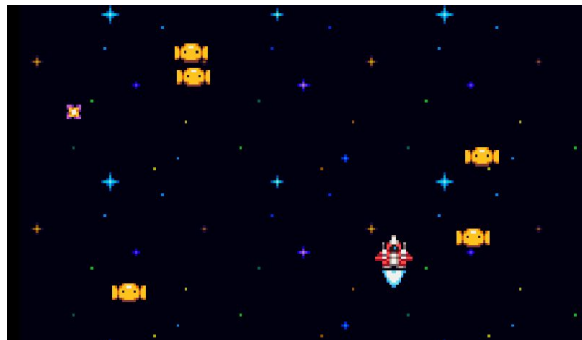
- Comprendre, de manière générale, Godot et ses composants.
- Comprendre comment les patrons de conception sont utilisés dans Godot.
- Comprendre les concepts de :
 - Nœud
 - Scène
 - Script
- Identifier et corriger les bugs dans un jeu

Activités

```
43 func _on_player_died():  
44     # print("game over")  
45     playing = false  
46     get_tree().call_group("enemies", "queue_free")  
47     game_over.show()  
48     await get_tree().create_timer(2).timeout  
49     game_over.hide()  
50     start_button.show()  
51  
52 func new_game():  
53     score = 0  
54     $CanvasLayer/UI.update_score(score)  
55     $Player.start()  
56     spawn_enemies()  
57     playing = true
```

Introduction

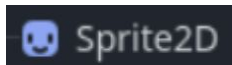
Nous allons créer des nœuds,
des scènes et des scripts.



Debugging

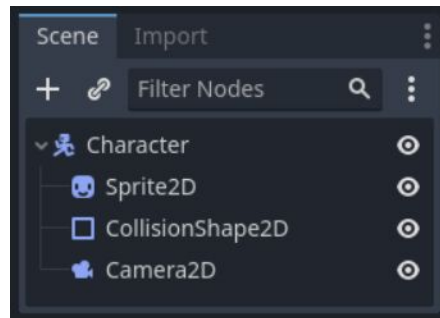
Nous allons examiner 2 bugs dans un
jeu, et aussi implémenter des
améliorations.

Les entités dans Godot Engine



Nœud

- Les "blocs de construction" du jeu.
- Ils représentent les fonctionnalités.
- Ils sont organisés de manière hiérarchique (arbre).

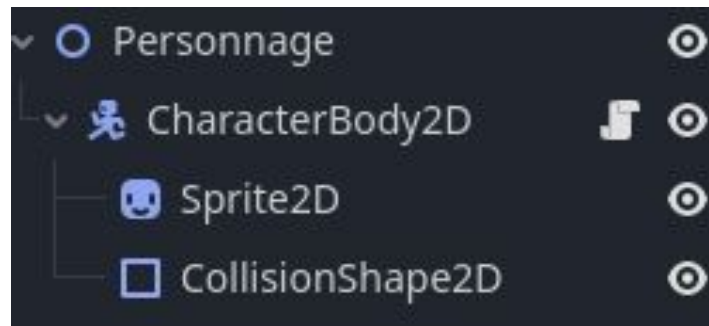


Scène

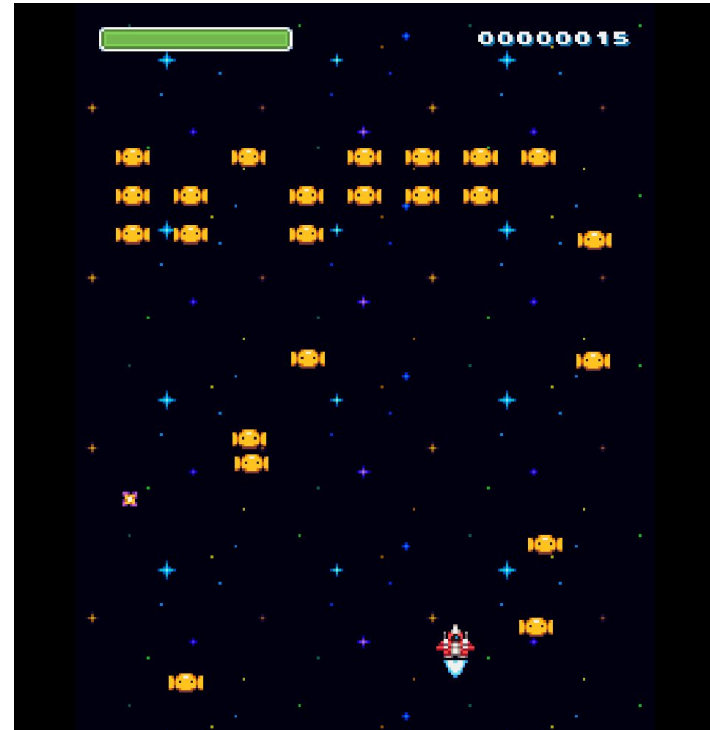
- Les groupes de nœuds.
- Les scènes sont elles-mêmes des nœuds (composition).
- Peuvent être lancées de manière interactive dans l'éditeur.

Nœuds et Scènes

1. Créer un nouveau projet.
2. Configurer les entrées.
3. Créer une node "Personnage".
4. Créer une scène "Main".
5. Élaborer des scripts pour permettre au "CharacterBody" de se déplacer.



Cas d'étude: Shmup



Scripts - Cycle de vie

```
# similaire à Python
func _init():
    print("Constructor")

func _ready():
    print("Le nœud a été ajouté à l'arbre")

func _process(delta):
    print("À chaque frame, liée au framerate")

func _physics_process(delta):
    print("Fréquence fixée")
```

Scripts - Méthodes et commandes importantes

Méthode/commande	Description
<u>Input.get_axis(action_negative, action_positive)</u>	Spécifier une action pour retourner une valeur négative et l'autre pour retourner une valeur positive.
<u>Input.is_action_pressed</u>	Vérifier si une touche d'action est appuyée, retourne true/false.
<u>move_and_slide()</u>	Implémenter un système de déplacement de style plateforme. Cela fonctionne également pour les jeux en vue top-down.
<u>print(someText)</u>	Écrire dans la console.
F10, F11, F12	Parcourir le code pendant le débogage (Step into, Step Over, Continue).

Scripts - Portée des variables

- Les variables sont visibles uniquement dans le script où elles ont été créées.
- Nous pouvons accéder aux variables dans les autres scripts 'au-dessus' de nous dans l'arbre de scène en utilisant `get_node('..')`.
- Nous pouvons utiliser `@export` pour faire une variable visuellement éditable dans l'éditeur

Bug 1 - Le vaisseau fantôme

Comportement actuel : Après le "game over", le vaisseau spatial peut encore tirer, même s'il est invisible.

Résultats attendus : Après le "game over", le vaisseau spatial ne doit plus pouvoir tirer.

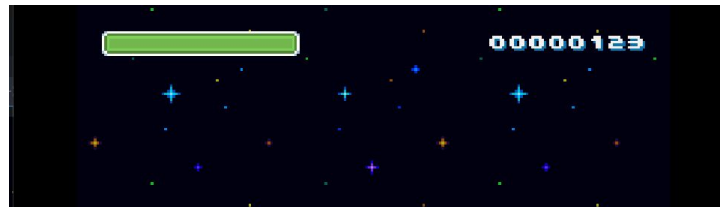
Bugfix : Recherchez la fonction responsable du tir et ajoutez une condition pour l'annuler si le jeu n'est pas commencé.



Bug 2 - Mauvaise score

Comportement actuel : Lorsque nous démarrons le jeu, le score est "123".

Résultats attendus : Lorsque nous démarrons le jeu, le score doit être "000".



Bugfix : Recherchez la fonction responsable de l'initialisation du score. Assurez-vous de définir le score à zéro au début du jeu.

Amélioration 1 - Démarrer le jeu

Comportement actuel : Il faut cliquer sur 'Start' pour démarrer le jeu.

Résultats attendus : Nous devrions également pouvoir démarrer le jeu en appuyant sur la touche 'Enter'.

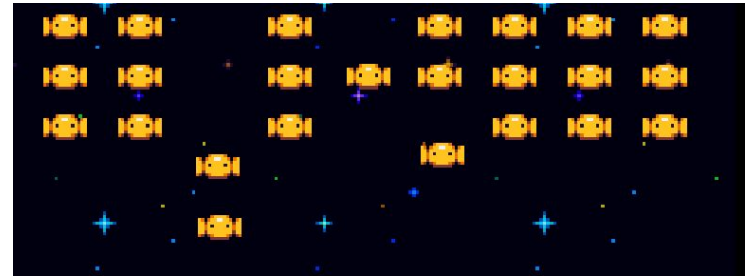


Solution : Ajoutez une nouvelle commande 'start' dans le 'Input Map'. À chaque frame, vérifiez si la touche est appuyée. Appelez la fonction '_on_start_pressed' lorsqu'elle est appuyée.

Amélioration 2 - Monter la difficulté

Comportement actuel : Les ennemis attendent longtemps avant de charger vers le joueur.

Résultats attendus : Les ennemis devraient attendre de moins en moins à mesure que le score augmente.



Solution : Changer la fonction qui définit le temps d'attente des ennemis.

Conclusion

- Les nœuds représentent les objets de jeu et les fonctionnalités.
- Les nœuds sont composés de plusieurs autres nœuds.
- Un groupe de nœuds forme une scène.
- Le game loop et la gestion des entrées sont intégrées dans le système de scripts.
- Le système de scripts utilise une syntaxe similaire à Python.
- L'éditeur nous permet d'identifier et de corriger les bugs dans les scripts.

LOG725 - Ingénierie et conception de jeux vidéo

Labo 3 - Développement de jeux avec Godot Engine

Gabriel C. Ullmann

École de Technologie Supérieure, Hiver 2024



Le génie pour l'industrie