

Feuille de TD n°2

Space Shooter 3D

M2 GAMAGORA

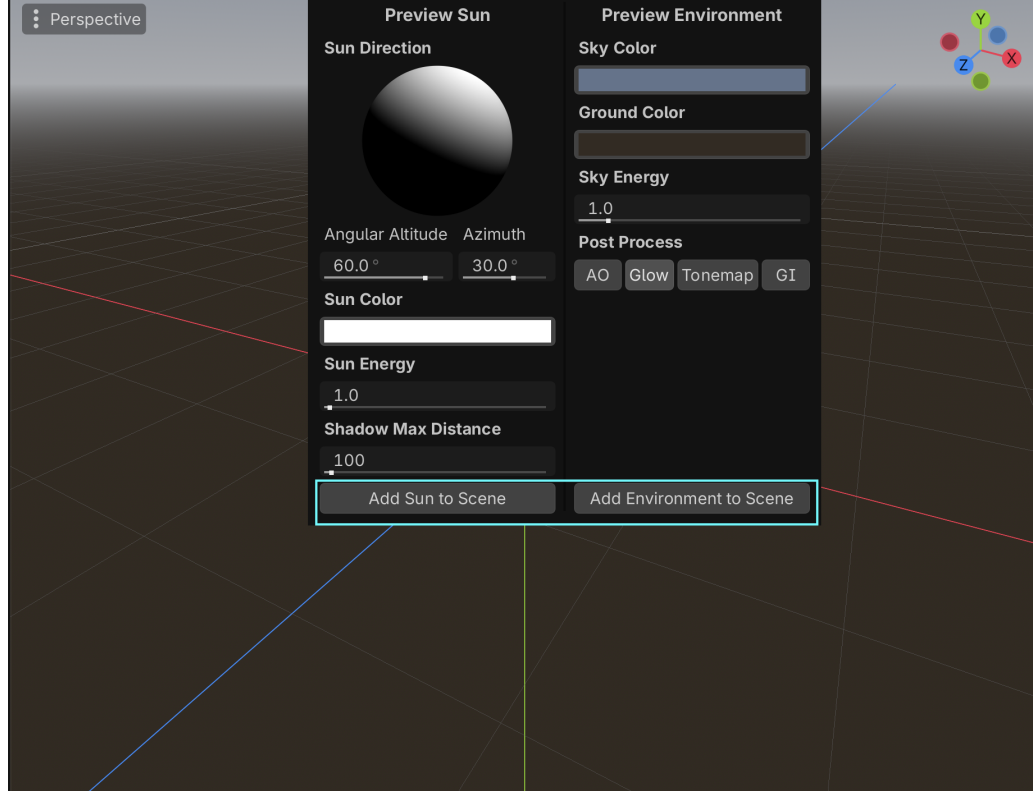
30 septembre 2026

Si vous n'avez pas fini le tutoriel dodge the creeps, commencez par le finir avant de commencer celui-ci.

1 Mise en place du projet

1.1 Skybox et soleil

- si vous lancez une scène malgré l’affichage dans l’éditeur avec une skybox rien ne va être éclairé dans le jeu
- Dans l’éditeur godot ajoute un world environment et un directional light (soleil) afin de pouvoir voir les scènes pris séparément.
- Vous pouvez ajouter ceux que godot utilise pour l’aperçu directement dans la scene principale (voir Figure ??).
- Ou ajoutez directement un node `WorldEnvironment` et un node `DirectionalLight` à la scène principale.



1.2 Création du joueur

1. Créez une nouvelle scène **Player** :
 - Node racine : `CharacterBody3D` (nommé `Player`)
 - Enfants :
 - `MeshInstance3D`
 - `CollisionShape3D` (forme `BoxShape3D` adaptée au mesh)
 - `Camera3D` (positionnée légèrement derrière le joueur)
2. Ajoutez un script pour gérer le mouvement du joueur en 3D, prenez votre temps et hésiter pas a essayer plein de chose differente car il est pas facile d’obtenir un rendu de déplacement agréable.
 - Pour que le mouvement rende bien il faut faire en sorte que la caméra ne se deplace pas parfaitement en meme temps que le joueur.
 - Pour cela il existe plusieurs possibilités :
 - Avoir la camera en sibling du joueur et faire un script qui interpole la position de la camera vers la position au repos regardant le joueur.
 - Utiliser un `SpringArm3D` entre le joueur et la caméra pour un effet de suivi amorti.

1.3 Création de cibles statiques

1. Créez une scène **Cible** :
 - Node racine : `Area3D`
 - Enfants :
 - `MeshInstance3D` (primitive `SphereMesh` rouge de rayon 1.5 unités)
 - `CollisionShape3D` (forme `SphereShape3D`)
2. Gérer la détection de collision avec le signal `area_entered` pour detruire le joueur lorsqu’il se prend un asteroid.

1.4 Création du projectile

1. Créez une scène **Projectile** :
 - Node racine : `RigidBody3D`
 - Enfants :
 - `MeshInstance3D` (primitive `CapsuleMesh` jaune 0.2×1 unités)
 - `CollisionShape3D` (forme `CapsuleShape3D`)
2. Ajoutez un script pour faire avancer le projectile vers l’avant en utilisant `translate_object_local`.
3. Ajoutez la détection de collision avec les signaux `area_entered` pour supprimer le projectile et detruire les asteroides.
4. Utilisez les layer de collision et les masque de collision pour faire en sorte que seul les bonnes collisions sont enregistré
5. Gérez la suppression automatique du projectile après une certaine distance ou durée si il n’y a pas eu de collision.
6. gerer le tir de projectile :
 - Sur le joueur, détectez l’appui sur une touche (ex : clic gauche ou barre espace) et instanciez un projectile.
 - Pour faire en sorte que le projectile ne soit pas déplacé avec le joueur, il faut l’instancier en dehors du joueur. par exemple avec `add_sibling`.
 - Comme le projectile n’est pas un enfant de joueur il faut penser a lui donner la bonne orientation et position au moment de l’instanciation. Utiliser `global_transform`.
 - Faire en sorte que le joueur ne puisse pas tirer en continu (ex : délai entre chaque tir).

1.5 Création de la scène principale

- Ajoutez le joueur et plusieurs cibles réparties dans l’espace 3D
- Affichez un écran de défaite si le joueur se prend une cible (ex : astéroïde).
- Affichez un écran de victoire si toutes les cibles sont détruites.
- Ajoutez un bouton pour recommencer la partie.

2 Améliorations

2.1 Ajouter un fond spatial

- Modifier le World Environment pour aller dans le theme spatial.
- Pour cela vous aller devoir créer une skybox.
- Vous pouvez utiliser une texture 2D equirectangulaire (360°).
- Vous devrez ensuite utilisez un `PanoramSkyMaterial`

2.2 Ajouter des effets de particules

- Créer l’explosion d’une cible.
 - Utilisez un node `GPUParticles3D` pour simuler une explosion.
 - Utilisez un `ParticlesMaterial` pour configurer les particules.
 - Dans le cas ou vous n’aurez jamais fais ca cette video peut etre un premier pas dans les particules 3D : <https://www.youtube.com/watch?v=tjSxICUXMmM>
 - Utiliser cette explosion pour l’explosion d’un asteroid suite a l’impact d’un projectile et l’explosion du joueur
- Créer un effet de propulsion pour le joueur.
 - Utilisez un node `GPUParticles3D` pour simuler la propulsion.
 - Si vous ne savez pas comment faire vous pouvez regarder cette video : https://www.youtube.com/watch?v=F27vQR_ivsc

2.3 Réaliser un pooling de projectiles

1. Implémentez un système de réutilisation des projectiles (object pool) pour éviter de créer/détruire trop d’instances.
2. Créez un script qui gère une liste de projectiles inactifs et les réactive au besoin.
3. Pour cela le plus simple est de créer un Autoload singleton.
4. Celui-ci devra pouvoir etre appele pour tirer un projectile a une position et rotation donnée.

2.4 Gerer la création d’asteroides a partir de certaines contraintes :

1. nombre d’asteroides
2. distance minimale entre les asteroides
3. position aléatoire dans un volume 3D defini par un AABB
4. taille aléatoire dans une plage définie
5. rotation aléatoire

2.5 Ajouter un système de score

1. Ajoutez un node `Label` dans l’UI pour afficher le score.
2. Incrémentez le score à chaque cible détruite.
3. Affichez le score final à la victoire ou à la défaite.

2.6 Ajouter un indicateur pour les ennemis

1. Le marqueur devra indiquez la position sur l’écran d’ennemis pour les voir meme si ils sont trop loin ou hors de l’écran.
2. Le marqueur devra se deplacer sur le bord de l’écran en fonction de la position relative de l’ennemi par rapport au joueur.
3. Pour savoir si l’ennemi est visible ou non vous pouvez utiliser la fonction `is_position_in_frustum` de la camera.
4. Si vous ne voyez pas comment realiser cela vous pouvez vous referer a cette video <https://www.youtube.com/watch?v=EKVYfF8oG0s>

2.7 Ajouter des sons spatialisés

1. Ajoutez des nodes `AudioStreamPlayer3D` pour les effets sonores spatialisés (tir, explosion, moteurs).
2. Configurez l’atténuation et l’effet Doppler pour plus de réalisme.
3. Ajoutez un `AudioStreamPlayer` pour la musique d’ambiance et l’UI.
4. Déclenchez les sons dans les scripts aux bons moments avec positionnement 3D.

2.8 Ajouter des cibles mobiles

1. Modifiez le script des cibles pour qu’elles se déplacent en trajectoires 3D (orbites, spirales).
2. Ajoutez des vaisseaux ennemis avec des projectiles.
3. Implémentez des patterns de vol en formation et des manœuvres d’évitement.

2.9 Ajouter des boosts (power-ups)

1. Créez des objets bonus flottants avec rotation automatique et effets de particules.
2. Ajoutez une détection de collision entre le joueur et les boosts.
3. Appliquez l’effet du boost avec feedback visuel (changement de matériau, particules).

Astuce : Vous pouvez ajouter d’autres idées d’améliorations spécifiques à la 3D : boss avec modèles complexes, niveaux avec géométrie procédurale, shaders personnalisés, mini-map 3D, cockpit détaillé, etc.