

# Feuille de TD n°2

Space Shooter

M2 GAMAGORA

5 octobre 2025

*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 Création du joueur

1. Créez une nouvelle scène **Player** :
  - Node racine : **Area2D** (nommé **Player**)
  - Enfants :
    - **Sprite2D** (Utilisez une image de vaisseau ou remplacez par un noeud tel que texture rect pour faire un placeholder)
    - **CollisionShape2D**
2. Ajoutez un script pour gérer le mouvement du joueur (déplacement avec les touches ou une manette).
3. Pour cela il va falloir créer des actions dans les paramètres du projet puis ensuite verifier si ces actions sont appuyé dans le script.

### 1.2 Création de cibles statiques

1. Créez une scène **Cible** :
  - Node racine : **Area2D**
  - Enfants :
    - **Sprite2D** (rectangle rouge  $48 \times 48$  px)
    - **CollisionShape2D**

### 1.3 Création du projectile

1. Créez une scène **Projectile** :
  - Node racine : **Area2D**
  - Enfants :
    - **Sprite2D** (avec image ou placeholder rectangle jaune  $8 \times 16$  px)
    - **CollisionShape2D**
2. Ajoutez un script pour faire avancer le projectile vers l'avant (localement). Par exemple en utilisant `tranlate_object_local`.
3. Ajoutez la détection de collision pour supprimer le projectile .
4. Gérez la suppression automatique du projectile hors écran.

### 1.4 Mise en place du tir

1. Sur le joueur, détectez l'appui sur une touche (ex : clic gauche).
2. Instanciez un projectile et positionnez-le sur le joueur.
3. Orientez le projectile dans la direction de la souris.
4. Orientez en permanence le joueur vers la souris pour indiquer la direction de tir.
5. Faire en sorte que le joueur ne puisse pas tirer en continu (ex : délai entre chaque tir).
6. Faire en sorte que le projectile ne se deplace pas avec le joueur. utilisez `add_sibling` à la place de `add_child` pour l'ajouter au même niveau que le joueur.

### 1.5 Création de la scène principale

1. Créez une scène **Main** :
  - Node racine : **Node2D**
  - Ajoutez le joueur et plusieurs cibles.
  - Gérez la collision entre le joueur et les cibles.
  - Affichez un écran de défaite si le joueur se heurte à 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 Faire en sorte que la caméra suive le joueur

1. Ajoutez un node **Camera2D** comme enfant du joueur.
2. Vous pouvez modifier les parametre pour que le mouvement soit plus fluide (damping, drag margin, etc).
3. Activez la caméra en cochant la case **Current** dans l'inspecteur.
4. Ajustez les limites de la caméra si besoin (**Limit Left/Right/Top/Bottom**).
5. Changer la condition de suppression des projectiles pour qu'ils soient supprimés lorsqu'ils ont dépassé une durée de vie.
6. Bonus : ajoutez un fond avec parallax pour l'étendre autant que besoin <https://www.youtube.com/watch?v=FtKdg4gKL3w>

### 2.2 Ajouter un indicateur des cibles hors écran

1. Créez un système d'indicateurs (ex : flèches ou icônes) qui apparaissent sur les bords de l'écran pour signaler la direction des cibles hors champ.
2. Utilisez la position de chaque cible et comparez-la à la zone visible pour afficher ou masquer l'indicateur.
3. Bonus : animez ou colorez l'indicateur selon la distance de la cible.
4. <https://www.youtube.com/watch?v=iny4KAtVMCg>

### 2.3 Ajouter des effets

Vous avez plusieurs possibilités pour faire une "explosion" :

- Utiliser un sprite simple d'explosion (cercle rouge) et l'animer en taille (s'agrandit puis disparais en transparence).
- Déclencher un effet de particules (**GPUParticles2D**) pour simuler l'explosion (si vous n'avez jamais réaliser ce genre d'effet cette vidéo peut etre un bon point de départ : <https://www.youtube.com/watch?v=7IxxZgepCdY>).
- Jouer une animation d'explosion en utilisant un **AnimatedSprite**. (souvent le plus joli mais demande des assets le prévoyant, vous pouvez chercher sur interet des assets ayant une explosion)

Vous pouvez aussi ajoutez un effet de feu derrière le vaisseau, pour les sorties de moteur. <https://www.youtube.com/watch?v=SqLVJx17bNw>

### 2.4 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. Cela améliore les performances, surtout si le jeu comporte beaucoup de tirs.

### 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 des cibles mobiles

1. Modifiez le script des cibles pour qu'elles se déplacent (trajectoire simple).
2. Ajoutez des vaisseaux ennemis avec des projectiles pouvant tuer le joueur.
3. Ajoutez des patterns de déplacement pour varier la difficulté.

### 2.7 Ajouter des boosts (power-ups)

1. Créez des objets bonus (ex : augmentation de vitesse, tir multiple, invincibilité temporaire).
2. Ajoutez une détection de collision entre le joueur et les boosts.
3. Appliquez l'effet du boost sur le joueur pour une durée limitée ou stockez le boost pour utilisation quand le joueur le souhaite.

**Astuce :** Vous pouvez ajouter d'autres idées d'améliorations (boss, niveaux, effets visuels, etc.) pour aller plus loin !