

## Programmation jeux vidéo

Space Shooter

Devoir noté

Pour ce devoir, vous allez créer un petit Space Shooter 2D en pixel art. L'objectif n'étant pas de vous mettre en difficulté mais simplement d'appliquer ce que vous avez déjà appris, les mécaniques sont ici très proches de ce que vous avez déjà réalisé dans les précédents TD.

Même si le jeu est en 2D, je vous conseille d'utiliser **Unity 3D**.

Sur Moodle, téléchargez le package **space\_shooter** et importez-le sur Unity. Vous y trouverez :

- GRAPHICS
  - ANIMATIONS : les animations du jeu
  - SPRITES : les Sprites et SpriteSheets du jeu

Une fois le devoir terminé, vous devrez me l'envoyer sous forme de package par mail à [belthoise.felix@gmail.com](mailto:belthoise.felix@gmail.com), ayant pour objet "GAMAGORA : Evaluation" + votre nom.

Pour créer un package, il suffit de faire un clic droit sur votre scène de travail, puis de cliquer sur "Export Package", puis Export. Unity intégrera automatiquement toutes les dépendances de la scène sélectionnée (scripts, prefabs, sprites, ...).

Vérifiez bien que le package fonctionne en le ré-important sur un nouveau projet !  
Celui-ci devra être nommé avec votre prénom, votre nom et la version de Unity utilisée (ex: felix\_belthoise\_v6000.0.34f1).

### Attention :

- Le jeu ne doit pas crasher, avoir de boucle infini ou autre bug bloquant
- Votre codebase doit être claire, le nom de vos scripts explicites et le code bien commenté aux endroits qui peuvent poser question à la lecture
- Les attributs doivent respecter la même nomenclature partout
- La console doit être propre, c'est-à-dire afficher ni erreurs ni informations inutiles.
  - Vous pouvez laisser des logs si ceux-ci sont pertinents, explicites et peu nombreux
  - Certain warnings de Unity sont parfois inévitables, ceux-ci sont donc autorisés

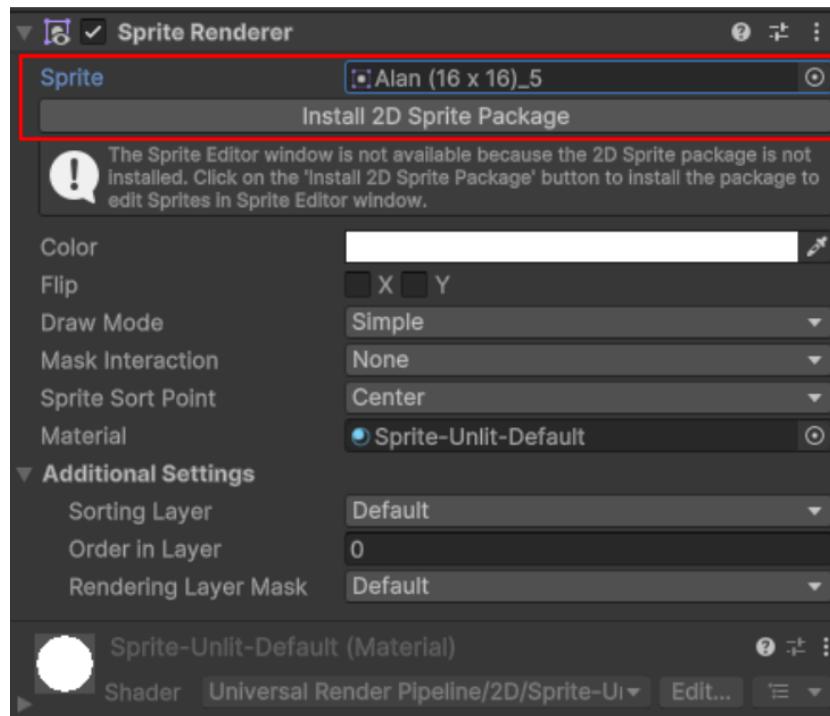
- La hiérarchie de votre scène doit également être explicite, les gameObject nommés par une nomenclature commune et être rangés de manière logique et intuitive

Pensez bien à terminer chaque feature avant de passer à la prochaine. Ce n'est pas grave si vous ne terminez pas tout, tant que ce qui a été réalisé jusque là a été fait correctement et proprement.

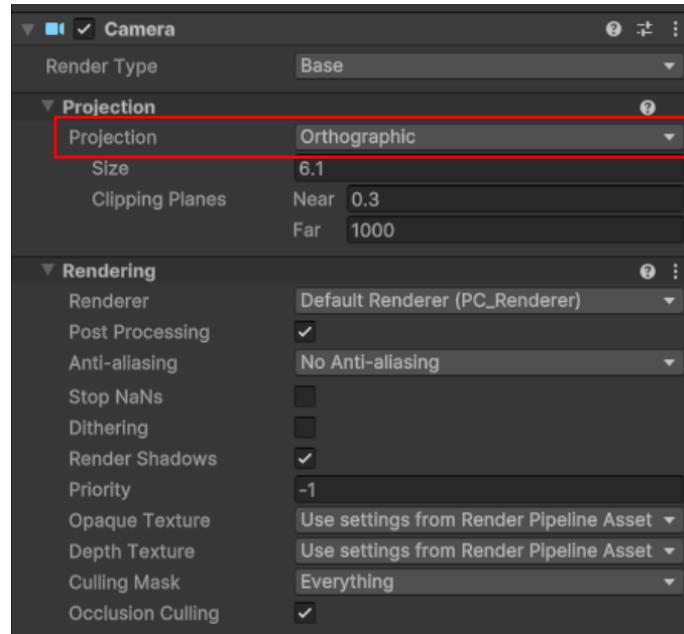
Comme d'habitude, n'hésitez pas à m'envoyer un message si besoin. La meilleure compétence d'un-e développeur-euse est de savoir quand demander de l'aide, je reste donc disponible pour répondre à tout blocage ou questionnement. Vous n'en serez pas pénalisés, bien au contraire.

#### Instructions préalables :

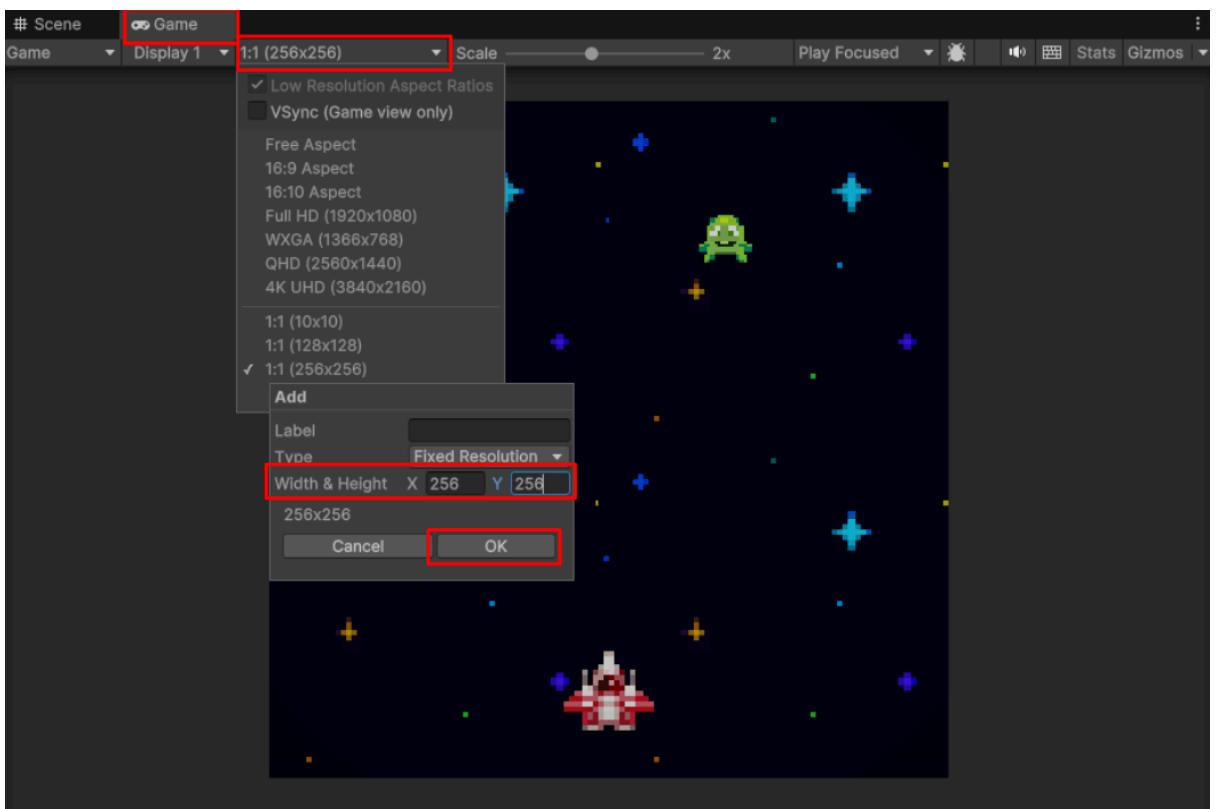
- Comme le jeu est en 2D, il vous faudra utiliser le composant **SpriteRenderer** pour afficher un sprite 2D dans une scène. Le package n'est pas toujours installé par défaut en fonction des versions de Unity, si ce n'est pas le cas vous pourrez l'ajouter en cliquant sur le bouton "install 2D package" qui apparaîtra sur le composant SpriteRenderer. **Il vous faudra renseigner un sprite dans le champ juste au-dessus pour débloquer le bouton.**



- Pensez à mettre la caméra en mode de projection **Orthographic**, pour éviter que vos sprites soient distordus par la perspective.



- Pour régler le ratio et la résolution de la caméra dans l'onglet Game, il vous faudra sélectionner le ratio en haut à gauche de la fenêtre, cliquer sur le "+" et régler la Width & Height. Je vous recommande une caméra 256 x 256 pixels pour ce projet.



## Instructions :

- Ajouter un background à l'aide des sprites **Space\_BG**
- Ajouter le vaisseau du joueur en bas de l'écran, centré horizontalement, qui possède le sprite **Player\_ship**
- Faites en sorte que le joueur puisse bouger de gauche à droite, dans les limites de l'écran, à l'aide d'un rigidbody
  - Il ne doit pas pouvoir bouger verticalement, ou ce de manière très limitée
  - La vitesse de déplacement doit être modifiable depuis l'inspecteur.
- Faites apparaître des ennemis à un rythme régulier, partant d'une position aléatoire en haut de l'écran et se dirigeant vers le bas
  - L'ennemi de base possède le sprite **Alan**
  - Le mouvement de base d'un ennemi doit être vertical, mais sentez-vous libre d'ajouter de la variété si vous le souhaitez
  - Les paramètres d'apparition et la vitesse des ennemis doit être modifiable sans ouvrir le code
- Ajoutez un système de points de vie au joueur
  - Celui-ci perd des points de vie si un ennemi le touche, mais également si un ennemi atteint la limite basse de l'écran
  - Le nombre de points de vie (ou une barre de vie) doit être affichée sur l'écran
  - Les points de vie maximum du joueur et les dégâts infligés par les ennemis doivent être modifiables sans ouvrir le code
- Ajoutez de quoi tirer : cliquer sur la barre d'espace doit lancer un projectile (sprite **Player\_beam**) depuis le vaisseau vers le haut, verticalement.
  - Si celui-ci touche un ennemi, le projectile doit disparaître (détruit ou recyclé) et l'ennemi perdre des points de vie.
  - Si les points de vie de l'ennemi tombent à 0, l'ennemi doit disparaître (détruit ou recyclé)
  - Les dégâts du projectile et le nombre de points de vie maximum d'un ennemi doivent être modifiables sans ouvrir le code
  - Maintenir la barre d'espace doit continuer d'envoyer des projectiles à un rythme régulier
- Ajoutez un système de score, qui s'incrémente lorsqu'on détruit un ennemi, affichés quelque part sur l'écran
- Lorsque le nombre de points de vie du joueur tombe à 0, le jeu doit afficher un Game Over (sprite **GAME\_OVER**) et il ne doit plus être possible de bouger ni de tirer.

## Bonus :

Si vous souhaitez aller plus loin, voilà de quoi améliorer votre prototype.

Cette partie est complètement optionnelle, mais ne sera évaluée **que si la section précédente est complète et fonctionne parfaitement**. Veillez à ce que tout fonctionne correctement avant de vous attaquer aux bonus, ou je n'en tiendrai pas compte.

- Faites défiler le background de manière infinie
- Ajoutez deux autres type d'ennemis :
  - Sprite **Bon\_Bon** : plus de points de vie mais plus lent
  - Sprite **Lips** : moins de points de vie mais plus rapides
- Ajoutez un tir chargé, en maintenant la barre d'espace (sprite **Player\_charged\_beam**), qui fait plus de dégâts mais prend quelques secondes à se lancer.
- Ajoutez les 2 animations situées dans le dossier ANIMATIONS, où vous le souhaitez nécessaire