

Examen Spécialisation Jeux avec Unity 1^{ère} année

Rédigé par COLLOT Florent et Thierry PROST

A réaliser par groupe de 1 à 3 étudiants. Pas de soutenance prévue.

Conditions d'utilisations : SUPINFO International University vous permet de partager ce document. Vous êtes libre de :

- Partager — reproduire, distribuer et communiquer ce document
- Remixer — modifier ce document

A condition de respecter les règles suivantes :

Indication obligatoire de la paternité — Vous devez obligatoirement préciser l'origine « SUPINFO » du document au début de celui-ci de la même manière qu'indiqué par SUPINFO International University – Notamment en laissant obligatoirement la première et la dernière page du document, mais pas d'une manière qui suggérerait que SUPINFO International University vous soutiennent ou approuvent votre utilisation du document, surtout si vous le modifiez. Dans ce dernier cas, il vous faudra obligatoirement supprimer le texte « SUPINFO Official Document » en tête de page et préciser notamment la page indiquant votre identité et les modifications principales apportées.

En dehors de ces dispositions, aucune autre modification de la première et de la dernière page du document n'est autorisée.

NOTE IMPORTANTE : Ce document est mis à disposition selon le contrat CC-BY-NC-SA Creative Commons disponible en ligne <http://creativecommons.org/licenses> ou par courrier postal à Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA modifié en ce sens que la première et la dernière page du document ne peuvent être supprimées en cas de reproduction, distribution, communication ou modification. Vous pouvez donc reproduire, remixer, arranger et adapter ce document à des fins non commerciales tant que vous respectez les règles de paternité et que les nouveaux documents sont protégés selon des termes identiques. Les autorisations au-delà du champ de cette licence peuvent être obtenues à support@supinfo.com.

© SUPINFO International University – EDUCINVEST – Avenue du Maine, 33 – 75015 Paris FRANCE. www.supinfo.com

Option de Spécialisation Jeux-Vidéos #1

Avec Unity

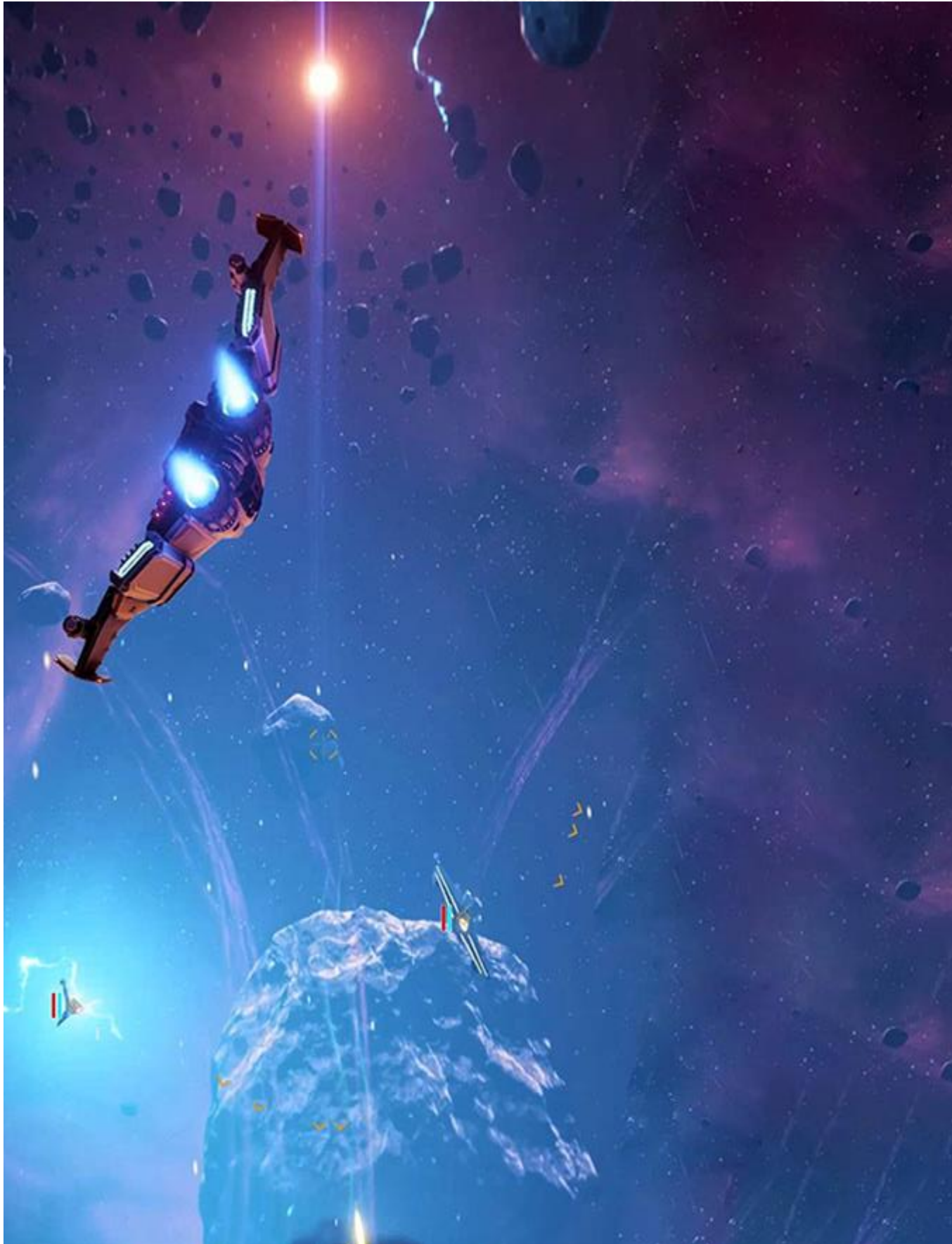


Table of contents

1	RENDU DE VOTRE PROJET	5
2	INTRODUCTION	6
3	GAME-DESIGN	7
3	ENNEMIS & GAMEPLAY.....	8
4	L'INTERFACE (UI).....	9
5	ENVIRONNEMENT.....	10
6	CONTRÔLES	11
7	COLLISIONS & PARTICULES.....	12
8	PROCESSUS DE DEVELOPPEMENT	13
9	NOTATION.....	13

1 RENDU DE VOTRE PROJET

A l'aide d'Unity vous aurez **jusqu'au 18 août 23h59** pour réaliser un Space Shooter en 3D en groupe de 1 à 3 personnes.

Vous pourrez trouver les ressources graphiques nécessaire à la création du projet gratuitement sur le [store Unity](#). Vous êtes libre dans le choix des ressources. Votre professeur aura pu également vous partager quelques « assets » lors de votre dernier cours avec lui, disponible dans l'onglet « Fichiers » sur le canal Teams de votre spécialisation.

Vous n'êtes pas autorisé à héberger publiquement votre projet en ligne, utilisez le code de quelqu'un d'autre, copiez - collez du code trouvé sur internet.

Si quelqu'un est surpris en train de tricher, il recevra une note de 0 et aura une mention « cheater ».

Votre rendu sera le dossier racine de votre projet Unity.

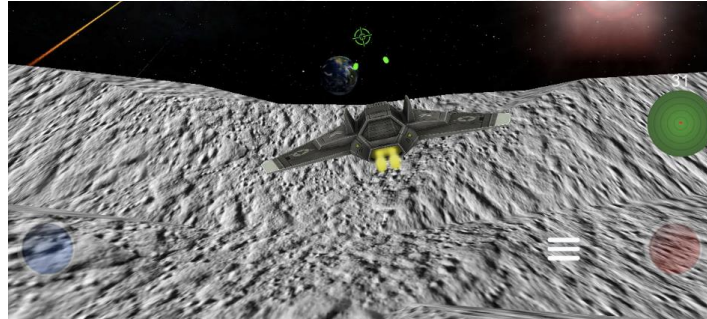
Il sera à uploader sur OneDrive (du fait de la taille du projet qui risque d'être trop important pour être inclus dans un courriel, il est préférable d'utiliser OneDrive à partir de Teams, onglet « Fichiers » puis « OneDrive » puis Créer un dossier et nommez-le comme c'est indiqué ci-dessous. A partir de la liste des dossiers, sélectionner votre dossier et ouvrez le dans OneDrive. Uploadez vos fichiers et partagez-le avec le « bouton Share » à Mr PROST et Mr COLLOT).

Vous devez ensuite copier le lien de votre dossier et l'envoyer par mail à M. PROST (Thierry.PROST@supinfo.com) et à M. COLLOT (Florent.COLLOT@supinfo.com).

Le dossier de rendu DEVRA être nommé ainsi :
« Dev-VideoGames_IDMEMBRE1_IDMEMBRE2_IDMEMBRE3 »
(par exemple : Dev-VideoGames_224563_214752).

Si le dossier partagé est vide ou bien que le nom du dossier ne respecte pas la nomenclature demandée, **votre devoir pourra être sanctionné d'un 0/20.**

1 INTRODUCTION



L'objectif de ce devoir est de réaliser un MVM (Minimum Viable Product) d'un « [Shoot'em up](#) » en 3D dans un niveau unique présentant un nombre croissant d'obstacles et d'ennemis et où l'objectif sera de détruire ces ennemis avec un niveau de difficulté accru avec la présence de boss éventuels (partie BONUS) à certains moments clés du déroulement de votre scène. Un bon exemple de ce type de jeu pourrait être l'excellent Starfox64 de Nintendo : <https://www.nintendo.fr/Jeux/Nintendo-64/Star-Fox-64-1093854.html>

Bien évidemment, vous devrez introduire une gestion des scores basée sur la destruction des ennemis et boss rencontrés, vous devrez traiter les conditions de dégâts subis par votre vaisseau (avec une barre de vie inhérente), le nombre de vies que votre joueur possèdera ainsi que la condition de « Game Over » si votre joueur ne possède plus de vies. Un écran d'interface de début avec un menu et de fin de jeu devra également être implémenté.

Lors de la réalisation de votre projet, vous serez amené à utiliser tous les thèmes que vous avez rencontrés lors de vos cours, à savoir planifier votre projet et son « Game-Design », manipuler un environnement, définir de la physique pour les différents éléments constituant votre jeu, utiliser des animations et des effets de particules, définir des contrôles, implémenter un réticule de visée, une gestion des tirs, des différents types de dégâts et leurs munitions ainsi que leurs effets, traiter la bande sonore, utiliser l'outil de « Timeline » de Unity, définir des interfaces, gérer divers effets de forces (déplacement horizontal et vertical, roulis, accélérations, etc...), peut-être implémenter un système de boucliers sous forme de bonus à récupérer qui disparaît avec le temps ou bien selon le nombre de dégâts reçus, etc.. le tout dans un rendu de qualité et un caractère ludique prononcé.

Features And Requirements

- **Player Movement:** Horizontal and vertical movement.
- **Shooting:** Player and enemies can shoot bullets which do damage to opponent. Speed, rate of fire and damage should be tunable.
- **Enemy Paths:** Enemies should travel on paths. Type of enemies, number of enemies and rate of spawn need to be tunable. Difficulty scaling and randomisation of paths is needed.
- **Score:** Points are given for killing enemies.
- **Camera Rail:** Path through the level that the camera follows.
- **Menu system:** Start menu, pause screen, death screen.

2 GAME-DESIGN

Level Design - What To Look For

- Layout to support your beats.
- Visual variation.
- Interesting dodging.
- Line of sight for spawning.
- Entry points for enemies.
- Room for enemy behaviour.
- Multiple paths.



Tout projet doit se doter d'une planification précise concernant le résultat que l'on souhaite atteindre. Tout ceci se fait à travers un « Game-Design ».

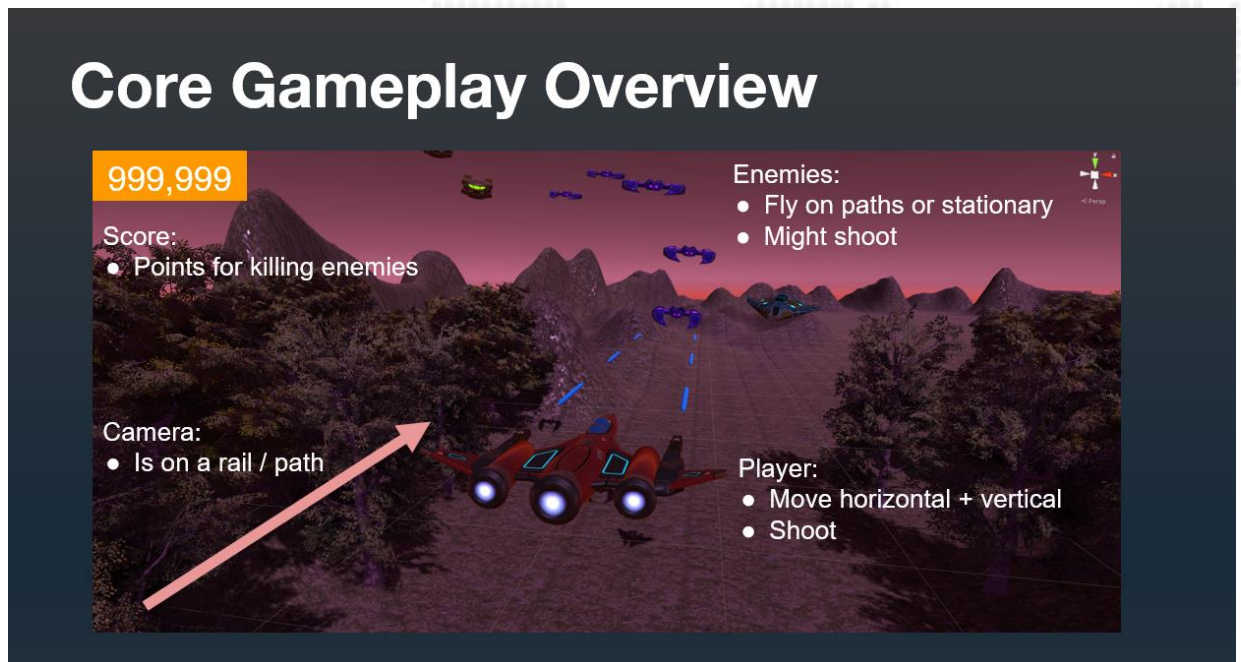
Dans notre cas présent, cela peut se définir en 3 axes principaux :

1. Expérience de jeu du Joueur
 - On pourrait être par exemple dans un monde de chaos où l'on doit détruire l'ennemi directement sur sa planète d'origine afin de supprimer la menace que cet ennemi constitue pour notre monde
2. Mécaniques de jeu
 - Mécaniques d'évitement et de tirs / shoots à la troisième personne
3. Boucle de jeu principale
 - L'objectif pour le joueur est d'arriver à aller le plus loin possible sans être détruit afin de réaliser le score le plus important possible et redémarrer au début en cas d'échec.

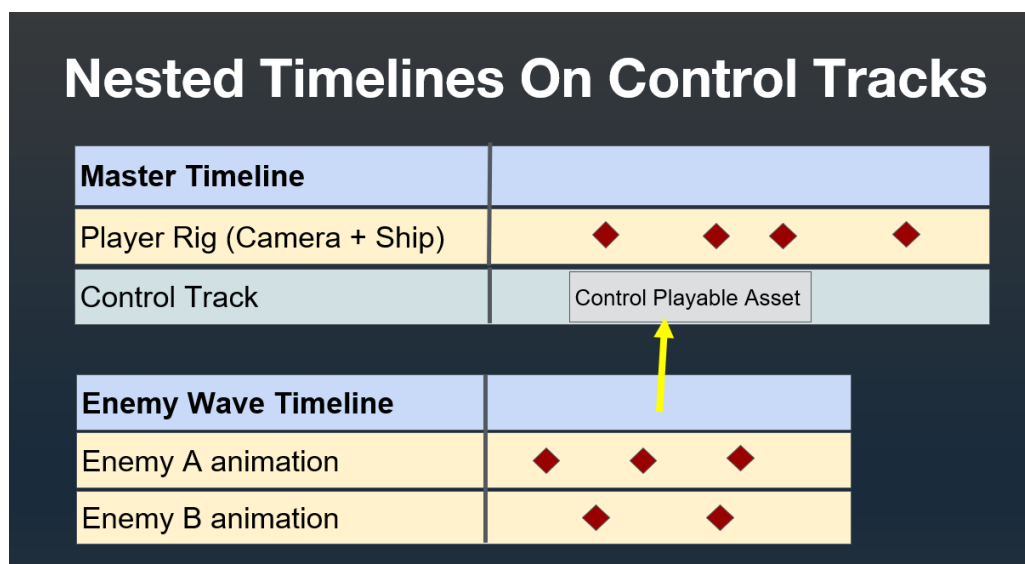
Beaucoup de choses peuvent être imaginées et nous vous laissons un champ relativement libre concernant leurs élaborations. Tout ceci doit faire l'objet d'un **premier document de « Game-Design »** présentant les points détaillés ci-dessus (voir plus) et **un autre document technique** présentant les technologies utilisées propres au développement de ce projet.

Une partie importante de votre note sera concernée par la qualité et le détail de votre « Game-Design ».

3 ENNEMIS & GAMEPLAY

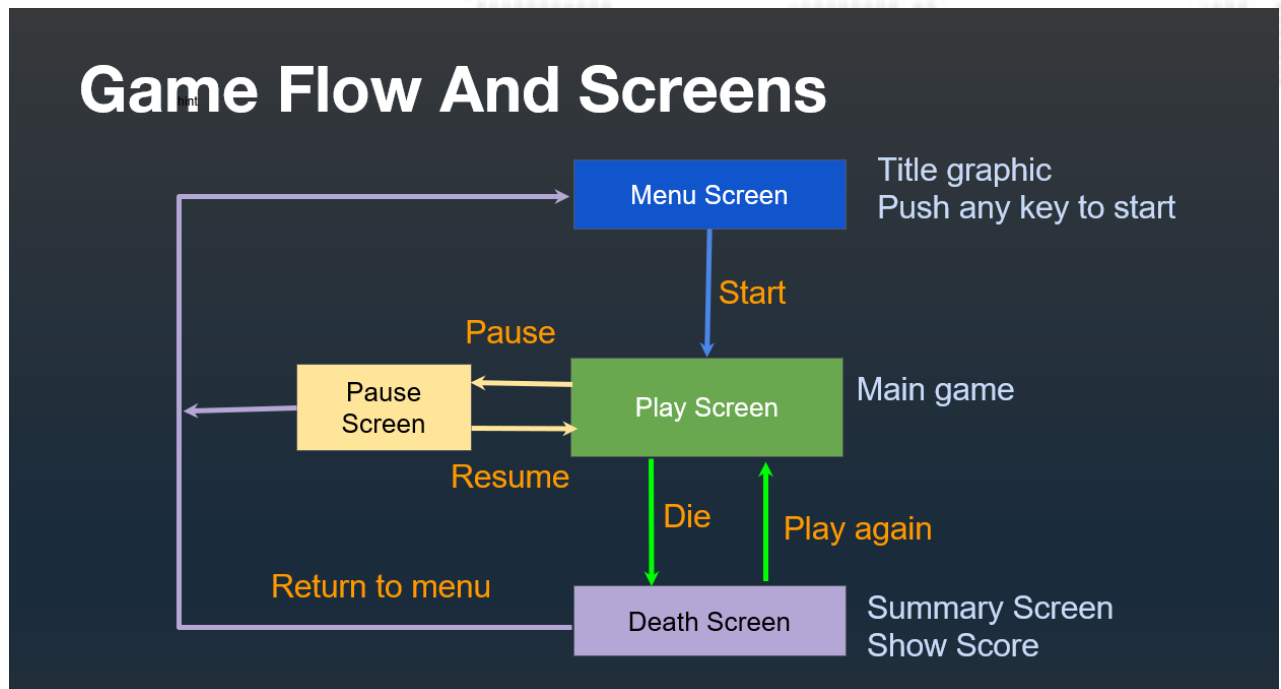


Votre gameplay devra être défini avec exactitude concernant l'évolution de votre joueur, sa manière de se déplacer et d'interagir avec les ennemis (tirs, évitement). Votre joueur suivra une trame fixe, comme le long d'un rail. Vos ennemis utiliseront le même système (merci à l'outil Timeline de Unity) et les différentes vagues d'ennemis apparaîtront à des moments clefs (triggers).



Cette section est laissée volontairement courte afin de laisser libre court à votre imagination !

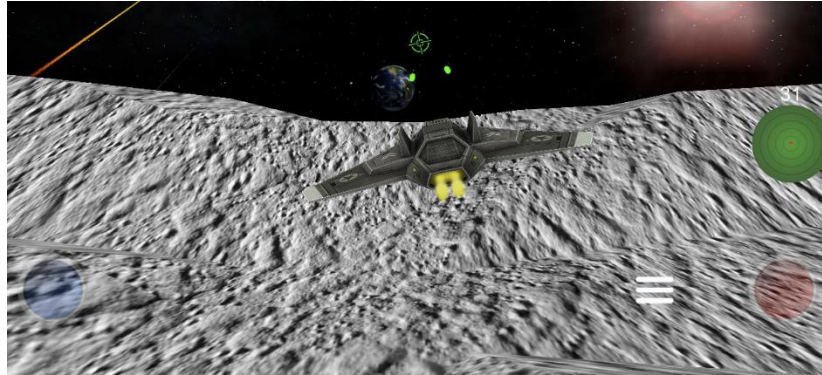
4 L'INTERFACE (UI)



Votre jeu devra se doter de plusieurs interfaces ayant chacune des utilités diverses :

- Menu d'accueil
 - o Afin de démarrer une partie
 - o Ou bien quitter le jeu
- Interface du jeu
 - o Comprends l'aire de jeu principale
 - o Les indicateurs (nombre de vies, jauge de dégâts du vaisseau)
 - o Boutons d'interface divers (quitter partie, paramètres, etc ...)
 - o Affichage du score
- Menu pause
 - o S'affiche lors de la mise en pause
 - o Permet de reprendre le cours de la partie
- Écran « Game Over »
 - o Indique que la partie est perdue et terminée
 - o Permet de recommencer une nouvelle partie
 - o Ou bien un retour au menu d'accueil

5 ENVIRONNEMENT



Vous allez avoir à définir dans un premier temps votre terrain avec sa forme, ses montagnes et ses divers éléments graphiques ainsi que quelques éléments de décors (biome particulier, etc...). Ce « level » représentera une planète que vous survolerez à ras du sol.

Il sera également nécessaire d'importer un « Skybox » en vous servant des assets « gratuits » disponibles sur le Unity Store ou bien à partir des ressources fournies par votre professeur.

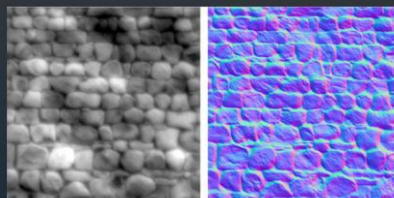
Stretch Goals - Damage and Shields

- **Player Shield:** When damaged, the player's shield depletes. Can be increased by picking up a shield pickup.
- **Pickups:** Flying over a shield pickup will increase the player's shields.
- **Momentary invulnerability:** After taking damage, the player cannot take damage again for X time.

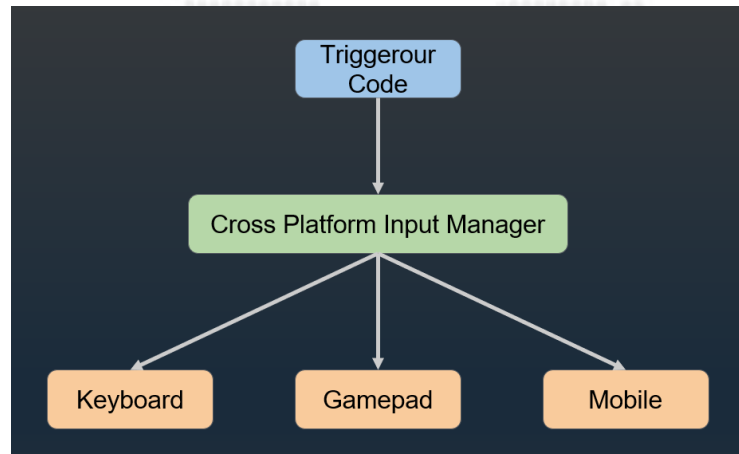
Tout au long du parcours, vous pourrez définir des objets Bonus qui pourront modifier vos armes, fréquence de tirs, fournir une accélération temporaire ou bien la récupération d'un bouclier qui disparaîtra après un certain nombre de dégâts reçus ou bien après un certain temps.

Enfin vous devrez choisir avec soins les textures qui composeront votre terrain, voir repeindre certaines parties de votre terrain avec les outils que Unity met à votre disposition. Vous devrez traiter l'éclairage de votre scène et la bande sonore.

Bump Map, Height Map, Normal Map

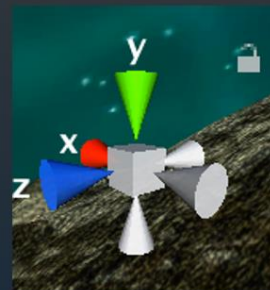
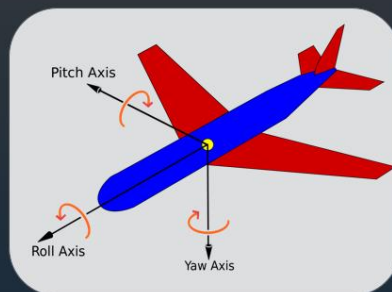


6 CONTRÔLES



Votre vaisseau devra pouvoir être contrôlé horizontalement et verticalement mais vous aurez à traiter également la possibilité de se déplacer selon 3 axes (roulis, pitch, yaw) afin d'animer votre vaisseau lors d'événements particuliers (mouvement d'évitement avec un roulis complet sur 360° par exemple, un effet de pitch lorsque le vaisseau se déplace verticalement ou bien encore un roulis léger lors des déplacements horizontaux). Soyez créatifs !

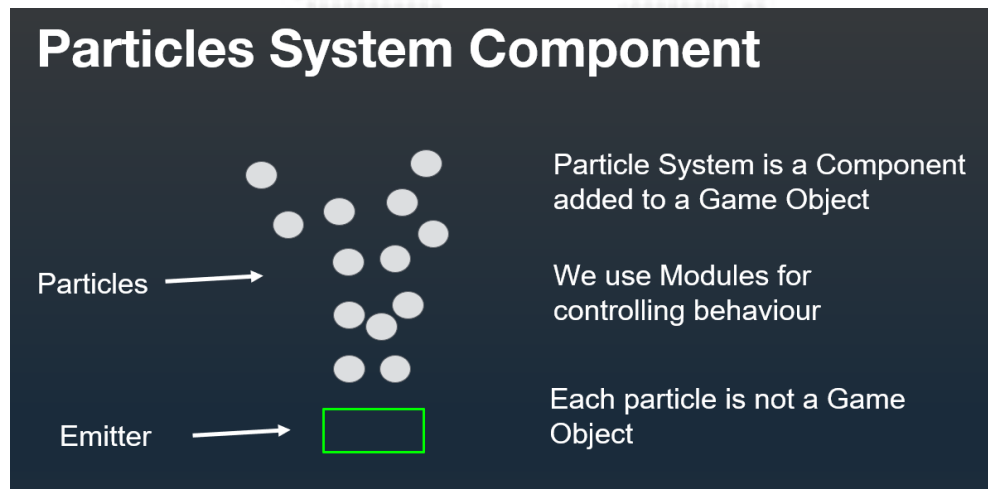
Setting Local Rotation



```
transform.localRotation = Quaternion.Euler( x , y , z );  
transform.localRotation = Quaternion.Euler(pitch, yaw, roll);
```

Vous pouvez vous servir encore une fois des ressources Unity Store pour le choix de votre / vos vaisseaux ou bien les ressources fournies par votre professeur.

7 COLLISIONS ET PARTICULES



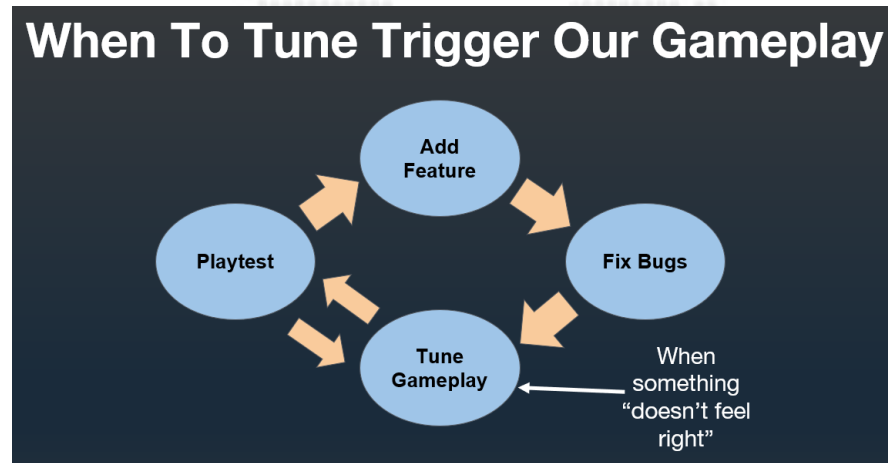
La gestion des collisions sera une partie importante de votre devoir à implémenter. Cette gestion devra certainement être utilisée sur votre terrain, le vaisseau du joueur, les ennemis, et les tirs émis.

Votre vaisseau aura une barre de vie limitée en termes de dégâts reçus / obstacles rencontrés. Tout ceci sera déterminé par vos collisions.

Concernant les particules, elles pourront être visible sur les moteurs des vaisseaux (celui du joueur ou les ennemis), lors du « Game Over » lorsque le vaisseau du joueur est détruit, lorsque les tirs ennemis vous touchent, ou bien le contraire lorsque des ennemis sont touchés ou détruits ; mais encore lorsque l'on heurte un obstacle ou un ennemi.

	Static Collider	Rigidbody Collider	Kinematic Rigidbody Collider	Static Trigger Collider	Rigidbody Trigger Collider	Kinematic Rigidbody Trigger Collider
Static Collider		Collision			Trigger	Trigger
Rigidbody Collider	Collision	Collision	Collision	Trigger	Trigger	Trigger
Kinematic Rigidbody Collider		Collision		Trigger	Trigger	Trigger
Static Trigger Collider		Trigger	Trigger		Trigger	Trigger
Rigidbody Trigger Collider	Trigger	Trigger	Trigger	Trigger	Trigger	Trigger
Kinematic Rigidbody Trigger Collider	Trigger	Trigger	Trigger	Trigger	Trigger	Trigger

8 PROCESSUS DE DEVELOPPEMENT



Votre travail, surtout s'il est réalisé en équipe, devra certainement être maintenu sur un dépôt GIT (privé, car tout dépôt public constitue une infraction aux règles de ce devoir).

Vous pouvez également utiliser la méthodologie Agile afin de vous répartir les tâches entre vous. Pensez bien au cycle de développement que l'on a abordé en cours :

- Ajout de fonctionnalité
- Corriger les bugs
- Affiner / Polir votre résultat
- Tester

... et on recommence sur une nouvelle fonctionnalité.

9 NOTATION

- Game-design et documentation technique	5 points
- Élaboration de l'environnement (terrain, skybox, décors, biome, etc...)	2 points
- Animations & Contrôles du vaisseau (roulis, pitch, yaw, accélération, etc...)	3 points
- Gestion des ennemis et de leur apparition par vagues	3 points
- Collisions et physique (terrain, vaisseaux, tirs, etc...) / Effets de particules	2 points
- Interface (écran démarrage, pause, « Game-Over », interface principale)	2 points
- Éléments bonus pour le joueur ainsi que le développement de leurs effets	3 points
Total 20 points	

- PARTIE BONUS : Créativité et imagination démontrée / Système de boss / autres...

Etonnez-nous ! Votre projet pourra être publié et présenté (supinfo.com, lab, twitter, facebook)

- Jusqu'à 5 points bonus supplémentaires