



Piscine Unity - d07

Navmesh, Physique 3D, GUI 3D

Staff staff@staff.42.fr

Résumé: Ce document contient le sujet du jour 07 de la piscine Unity de [42](#).

Table des matières

I	Préambule	2
II	Consignes generales	3
III	Règles spécifiques à la journée	5
IV	Exercice 00 : Dessine moi un bac à sable	6
V	Exercice 01 : Un tank pour tous les gouverner	7
VI	Exercice 02 : Wargames	9
VII	Exercice 03 : Pimp My Tank !	10

Chapitre I

Préambule

Il est temps de partir en expédition. A bord de votre tank, vous pourrez vous déplacer en terrain ennemi et vous frayer un chemin afin de les surprendre. Aujourd'hui on va donc s'inspirer de **Worlds of Tanks** ou autres jeux du même genre. Comme d'habitude je vous invite fortement à aller jeter un oeil à ces petites perles et vous faire une idée sur le style attendu aujourd'hui.

Enfin comme je suis sympa je vous donne une musique plutôt appropriée qui vous donnera de l'énergie et qui vous restera en tête au moins toute la journée.

Bon courage !

Chapitre II

Consignes generales

- La piscine Unity est à faire entièrement et obligatoirement en **C#** uniquement. Pas de Javascript/Unityscript, de Boo ou autres horreurs.
- L'utilisation de fonctions ou de namespaces non autorises explicitement dans le header des exercices ou dans les regles de la journee sera considéré comme de la triche.
- Pour une utilisation optimale de Unity, vous devez travailler sur le ~/goinfre, qui est en local sur le mac que vous utilisez. Pensez à bien récupérer vos projets avant de vous delog car le goinfre local est vidé régulièrement.
- Contrairement aux autres piscines, chaque journée ne demande pas un dossier ex00/, ex01/, ..., exn/. A la place pour la piscine **Unity**, vous devrez rendre votre dossier projet qui aura pour nom le nom de la journee : d00/, d01/, Toutefois, un dossier de projet contient par default un sous-dossiers inutile : le sous-dossier "projet/Temp/". Assurez-vous de ne **JAMAIS** pusher ce dossier dans votre rendu.
- Au cas ou vous vous poseriez la question, il n'y a pas de norme imposée à 42 pour le **C#** pendant cette piscine Unity. Vous pouvez utiliser le style qui vous plaît sans restriction. Mais rappelez-vous qu'un code que votre peer-evaluateur ne peut pas lire est un code qu'elle ou il ne peut noter.
- Vous devez trier les assets de votre projet par dossier. Chaque dossier correspond à un et un seul type d'asset. Par exemple : "Scripts/", "Scenes/", "Sprites/", "Prefabs/", "Sounds/", "Models/", ...
- Assurez-vous de tester attentivement les prototypes fournis chaque jour. Ils vous aideront beaucoup dans la compréhension du sujet et du travail attendu.
- L'utilisation de l'Asset Store d'Unity est interdite. Vous êtes encouragés à utiliser les assets fournis chaque jour (quand nécessaire) ou à en chercher d'autres sur le net s'ils ne vous plaisent pas, sauf bien entendu pour les scripts car vous devez avoir écrit tout ce que vous rendez (hors scripts fournis par le staff, obviously). L'Asset Store est interdit car quasiment tout le travail que vous avez à faire s'y trouve déjà sous une forme ou sous une autre. Néanmoins l'utilisation des Standard Assets de Unity est autorisée voir meme conseillée pour certains exercices.

- Pour les corrections à partir du d03 il vous sera demandé de build les jeux pour les tester. **C'est le correcteur** qui doit build le jeu vous devez donc évidemment toujours push vos projets/sources. De ce fait votre projet doit correctement configuré pour le build. **Aucun** réglage de dernière minute ne doit être toléré.
- Important : Vous ne serez pas évalués par un programme, sauf si le contraire est explicite dans le sujet. Cela implique donc un certain degré de liberté dans la façon que vous choisissiez de faire les exercices. Toutefois, gardez en tête les consignes de chaque exercice, et ne soyez pas FAINÉANTS, vous passeriez à côté de beaucoup de choses intéressantes.
- Ce n'est pas grave d'avoir des fichiers supplémentaires ou inutiles dans votre dossier de rendu. Vous pouvez choisir de séparer votre code en différents fichiers au lieu d'un seul, sauf si le header d'un exercice mentionne explicitement les fichiers à rendre. Un fichier ne doit définir qu'un et un seul comportement, pas de namespaces donc. Toute cette consigne ne s'applique bien évidemment pas au sous-dossier "projet/Temp/" qui n'a pas le droit d'exister dans vos rendus.
- Lisez le sujet en entier avant de commencer. Vraiment, faites-le.
- Le sujet pourra être modifié jusqu'à 4h avant le rendu.
- Même si le sujet d'un exercice est relativement court, ça vaut le coup de passer un peu de temps à comprendre parfaitement le travail attendu pour le faire au mieux.
- Parfois il vous sera demandé un soin particulier sur la qualité artistique de votre rendu. Dans ce cas, cela sera mentionné explicitement dans le sujet correspondant. N'hésitez alors pas à tester plein de choses différentes pour vous donner une idée des possibilités offertes par Unity.
- Par Odin, par Thor ! Réfléchissez !!!
-


Chapitre III

Règles spécifiques à la journée

- [Freedom](#)

Chapitre IV

Exercice 00 : Dessine moi un bac à sable

	Exercice : 00
Exercice 00 : Dessine moi un bac à sable	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : *.cs	
Fonctions Autorisées : Tout sauf mentionné dans les règles de la journée	
Remarques : n/a	

Bon maintenant que vous commencez à devenir des experts dans l'art du terraforming vous allez pouvoir nous concocter un magnifique terrain. Soyez imaginatif mais le thème doit rester en accord avec le jeu de la journée à savoir les tanks.

Voici donc quelques indications techniques sur votre terrain de jeu :


- Il devra faire 256 par 256.
- Essayez de le diviser en zones de jeux avec des endroits où l'on peut éventuellement se cacher, des hauteurs différentes.
- Utilisez plusieurs textures faites vous plaisir.
- Pour le type d'environnement choisissez ce que vous voulez l'exemple fourni n'est pas l'unique réponse possible.



Réfléchissez votre terrain en avance pour votre NavMesh. Evitez les terrains trop obtus et granuleux.

Chapitre V

Exercice 01 : Un tank pour tous les gouverner

	Exercice : 01
Exercice 01 : Un tank pour tous les gouverner	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : *.cs	
Fonctions Autorisées : Tout sauf mentionné dans les règles de la journée	
Remarques : n/a	

Maintenant qu'on a notre bac à sable on va pouvoir s'amuser dedans et y rajouter des jouets. Créez donc un tank, on doit pouvoir faire avancer le corps du tank en avant avec W et en arrière avec S, on peut orienter le corps du tank vers la gauche avec A et la droite avec D. Enfin on peut orienter le canon avec la souris. Le canon doit tourner uniquement sur l'axe y et doit toujours viser vers le pointeur de la souris (fiez vous à l'exemple qui est plus représentatif).

On doit également pouvoir utiliser un boost et faire avancer le tank plus vite en appuyant sur Maj Gauche. Le boost est limité et doit se recharger après un certain temps d'utilisation. Et que serait un tank sans mitrailleuse et missiles ? Ajoutez donc la possibilité de tirer avec le clic gauche de la souris avec une mitrailleuse et avec le clic droit des missiles. Les missiles sont disponibles de manière limitée après un certain nombre de tirs on ne peut plus en lancer. La mitrailleuse et les missiles doivent produire une particule au niveau de leur zone d'impact.




Pour la mitrailleuse et les missiles vous devez utiliser un Raycast Physique qui partira de votre canon et qui aura un range défini.

Pensez bien à ajouter du son à votre jeu (mitrailleuse, missiles, explosions, musiques,

...) afin de le rendre plus vivant.

Chapitre VI

Exercice 02 : Wargames

	Exercice : 02
Exercice 02 : Wargames	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : *.cs	
Fonctions Autorisées : Tout sauf mentionné dans les règles de la journée	
Remarques : n/a	


Bien maintenant qu'on a de quoi faire joujou on va ajouter du challenge. Pour ca rien de mieux que des ennemis. Créez donc une intelligence artificielle capable de repérer l'ennemi le plus proche et de l'attaquer lorsqu'elle est à portée. On entend par "à portée" lorsque le tank ciblé est dans le range et à la bonne hauteur pour être touché. Profitez en pour ajouter des points de vie aux tanks du coup. Lorsqu'un tank n'a plus de vie il explose et disparaît. Si votre tank explose la partie redémarre.

Deux trois considérations techniques maintenant :

- Vous devez evidemment utiliser un NavMesh pour déplacer les tanks contrôlés par l'IA.
- Il doit y avoir au moins 2 tanks ennemis présents et ils peuvent s'attaquer entre eux.
- L'IA ne doit pas toujours tirer de maniere régulière et peut manquer sa cible.
- Vous n'avez pas besoin de gérer le décor obstruant la visée de l'IA. Une IA qui tire dans la montagne parce que vous êtes derrière ne sera pas pénalisant.
- Si elle est bien concue l'IA ne devrait pas en arriver à rentrer dans un autre tank.

Chapitre VII

Exercice 03 : Pimp My Tank !

	Exercice : 03
Exercice 03 : Pimp My Tank !	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : *.cs	
Fonctions Autorisées : Tout sauf mentionné dans les règles de la journée	
Remarques : n/a	

Rendons le tout encore plus classe en ajoutant un GUI. Mais pas n'importe quel GUIs non, un GUI 3D qui claque. On va donc afficher quelques informations essentielles autour du tank, à savoir la vie du tank et le nombre de missile restant. Si vous voulez en rajouter d'autre libre à vous. Mettez en place un petit signe distinctif qui permettra de comprendre qu'il s'agit bien d'un GUI en 3D et qui donnera donc une utilité à la troisième dimension.

Rajoutez également un petit sprite de crosshair (viseur) et faites le changer de couleur si le joueur touche une cible.