



Piscine Unity - d08

Les animations

Staff staff@staff.42.fr

Résumé: Ce document contient le sujet du jour 08 de la piscine Unity de [42](#).

Table des matières

I	Consignes generales	2
II	Préambule	4
III	Exercice 00 : Mise en place du décor	5
IV	Exercice 01 : Maya	7
V	Exercice 02 : Ennemis et non intelligence artificielle	9
VI	Exercice 03 : Les stats, parcequ'il en faut	11
VII	Exercice 04 : Mort et interface	13
VIII	Exercice 05 : Level up !	15

Chapitre I

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 II

Préambule

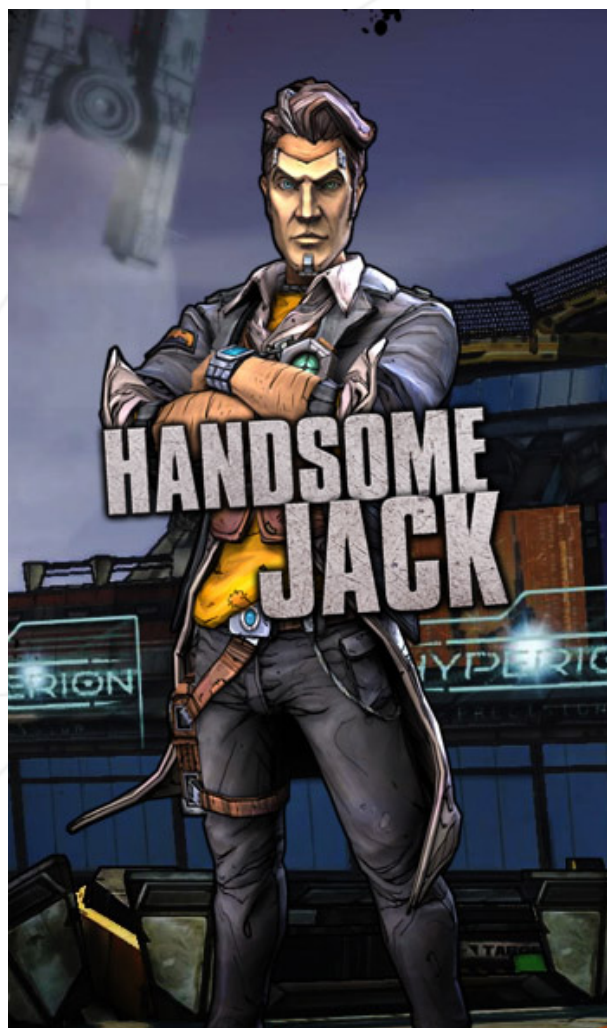



FIGURE II.1 – "Ma journée à moi ? Pas mal, je me suis acheté un poney en diamant parce que j'ai de la maille. Enfin tu vois, c'est cool, OK, bye."

Chapitre III

Exercice 00 : Mise en place du décor

	Exercice : 00
Exercice 00 : Mise en place du décor	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : La scène "ex00" et tout ce qui vous semble utile	
Fonctions interdites : Aucune	
Remarques : n/a	



ATTENTION ! ATTENTION ! Maintenant que j'ai votre ATTENTION ! quelques précisions sur les exercices de la journée. Aujourd'hui nous allons faire une démo technique. Donc : ne perdez pas de temps sur le level design et le décor ou alors faites le à la fin de la journée. Les assets aujourd'hui sont relativement cool c'est vrai, mais pas d'inquiétudes vous aurez d'autres occasions de les utiliser !



Pour tout le projet travaillez à échelle humaine. Utilisez un cube de 1 unité de côté et mettez sa hauteur à 2 pour avoir une idée de la taille d'un personnage et utilisez ça comme gabarit pour créer vos décors.

Pour ce premier exercice vous allez devoir créer un terrain simple, globalement plat, avec quelques bosses et des pentes plus ou moins abruptes qui serviront tout à l'heure.


Enfin vous devez créer un bâtiment comme un hangar par exemple. Sa taille doit être raisonnable, pas comme une petite maison, et pas comme une cathédrale non plus ! Un hangar fera l'affaire. Il faut bien entendu qu'il ait une porte, ouverte.



Si vous bossez vite en 30 minutes vous devez avoir terminé cet exercice. Si vous y passez plus d'une heure vous êtes vraiment mal partis pour la suite de la journée !

Chapitre IV

Exercice 01 : Maya

	Exercice : 01
Exercice 01 : Maya	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : La scène "ex00" et tout ce qui vous semble utile	
Fonctions interdites : Aucune	
Remarques : n/a	



Les assets fournis aujourd'hui ne sont pas tous égaux au niveau de leur échelle et ne sont pas tous prêts à l'emploi tels quels. A vous de les modifier pour leur donner une unité en utilisant tout ce que vous avez appris jusqu'ici.

Maintenant que le décor de base est placé vous devez y mettre Maya, notre personnage principal, que le joueur pourra contrôler.

Le modèle est fourni dans les assets ainsi qu'une série d'animations. Vous devez lui créer un animator avec 4 états : un idle qui sera l'état par défaut lorsque le joueur ne fait rien, un run qui sera déclenché lorsqu'on clique sur le terrain, un attack qui sera déclenché lorsqu'on clique sur un ennemi et qu'il est à portée et un dead qui sera déclenché si les points de vie de Maya tombent à 0.

On reparlera des ennemis dans l'exercice suivant donc pour le moment prévoyez juste un state "attack" qui sera déclenché lorsque Maya attaque.


Vous devez également placer la caméra pour qu'elle soit toujours centrée sur Maya et qu'elle soit plongeante. L'idée est de créer une caméra "Hack and Slash", cherchez des images ingame de Diablo ou Torchlight sur google pour vous faire une idée.

Vous devez créer un navmesh et mettre un NavMeshAgent sur Maya pour gérer ses déplacements. Votre navmesh doit bien être réglé pour que le personnage se déplace de façon logique. Vous devez avoir des pentes infranchissables sur votre terrain pour obliger le personnage à faire le tour. Si ce n'est pas le cas n'hésitez pas à retoucher votre terrain.

Une fois votre NavMesh crée vous devez gérer le déplacement du personnage et son changement d'animation de idle à run lorsque le joueur clique sur le terrain. Bien entendu la caméra doit suivre le mouvement.

Chapitre V

Exercice 02 : Ennemis et non intelligence artificielle

	Exercice : 02
Exercice 02 : Ennemis et non intelligence artificielle	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : La scène "ex00" et tout ce qui vous semble utile	
Fonctions interdites : Aucune	
Remarques : n/a	

Passons aux ennemis et au combat ! Pour simplifier les choses nous allons faire dans l'originalité : des zombies.

Créez des EnemySpawners. Ce sont des GameObjects vides avec un script qui prend en paramètre une liste de préfab d'ennemis et qui en font spawner un au hasard parmi cette liste. Vous devez créer deux préfab d'ennemis qui utiliseront les modèles lambent male et female, fournis dans les assets.

Ces ennemis doivent :

- Avoir un NavMeshAgent pour gérer leurs déplacements dans le décor.
- Avoir un animator avec 4 états : idle, run, attack, dead.
- Avoir une zone de détection pour savoir quand le joueur s'approche et courir vers lui pour l'attaquer.
- Avoir un script pour gérer tout ça avec également une gestion simpliste des points de vie. Chaque ennemi possède 3 points de vie.

Vous devez placer ces EnemySpawners sur la map et faire en sorte que lorsqu'un ennemi est tué un autre respawn quelque temps après au spawner concerné. Le "quelque temps après" est laissé à votre discrétion, le but étant de ne pas être trop court pour éviter que le joueur reste devant le spawn pour farmer les ennemis mais aussi de ne pas

être trop long pour que votre map soit suffisamment peuplée.


Vous devez également gérer le combat. Pour l'instant Maya est immortelle mais vous devez faire en sorte que les ennemis perdent un point de vie à chaque fois qu'elle attaque, et donc meurent au bout de 3 attaques. Arrangez vous pour que les dégâts soient bien appliqués au moment où l'arme frappe et non pas au début de l'animation d'attaque.

Pour attaquer il faut cliquer sur un ennemi. Si Maya est à portée d'attaque elle attaque automatiquement au corps à corps. Si elle est trop loin de l'ennemi elle doit d'abord courir vers lui avant de l'attaquer. Si le joueur garde le bouton de la souris enfoncé après avoir cliqué sur un ennemi, Maya doit attaquer l'ennemi jusqu'à ce qu'il meurt même si le curseur de la souris sort de la hitbox de l'ennemi.

Lorsqu'un ennemi n'a plus de points de vie, lancez son animation de mort. Vous devez créer une coroutine pour qu'une fois l'ennemi à terre depuis 2 secondes celui-ci s'enfonce dans le sol progressivement pour le détruire proprement lorsque le joueur ne le voit plus, plutôt que le faire disparaître d'un coup. N'utilisez pas d'autre façon de détruire vos ennemis, les faire disparaître dans le sol va vous poser certains problèmes dont la résolution est justement le but de l'exercice.

Chapitre VI

Exercice 03 : Les stats, parcequ'il en faut

	Exercice : 03
Exercice 03 : Les stats, parcequ'il en faut	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : La scène "ex00" et tout ce qui vous semble utile	
Fonctions interdites : Aucune	
Remarques : n/a	

Maintenant que vous avez le gameplay de base on va passer aux choses sérieuses et créer le système de stats et de combat avancé.



A partir de cet exercice nous allons voir de façon assez détaillée les mécaniques de gameplay d'un hack and slash. Certaines mécaniques étant assez difficiles à décrire de façon concise, si vous avez le moindre doute essayez la démo fournie pour vous faire une idée du comportement attendu. Si vous avez déjà joué à DiabloII/Torchlight vous devriez être en terrain connu. Si vous ne connaissez pas du tout les hack and slash ou seulement des plus récents comme DiabloIII ou PoE testez la démo, vraiment.

Chaque personnage dans le jeu (Maya comme les ennemis) doit avoir les caractéristiques suivantes :

- 3 caractéristiques principales : Strength(STR), Agility(AGI), Constitution(CON)
- Une stat Armor qui sera attribuée subjectivement pour le moment et pourra être modifiée plus tard par de l'équipement.
- Une stat HP qui est égale à $5 * CON$
- Une stat minDamage qui est égale à $STR / 2$, et maxDamage qui est égale à

minDamage + 4. Comme l'armor ces stats basiques seront modifiables par l'équipement.

- Une stat Level qui correspond à leur niveau.
- Une stat XP. Pour Maya c'est l'xp qu'elle a accumulé en tuant les ennemis, pour les ennemis c'est leur valeur individuelle en xp.
- Une stat money. Pour Maya c'est les crédits qu'elle a accumulé, pour les ennemis c'est leur valeur moyenne en crédits.

Vous pouvez choisir les chiffres que vous voulez pour vos personnages/ennemis mais pour le niveau 1 une bonne base est entre 10 et 20 points par stat.

Vous devez aussi rajouter une variable sur Maya qui précise le nombre de points d'xp nécessaires pour qu'elle passe qu niveau suivant.

Maintenant que les personnages ont des vraies stats on va pouvoir retoucher les combats pour les rendre moins basiques. Vous devez intégrer les formules suivantes dans vos combats : (et ne passez pas des heures à les affiner/ améliorer, c'est hors sujet)

Calcul des chances de toucher : $\text{hit} = 75 + \text{AGI} - \text{Target.AGI}$


Calcul des dégats de base : Random entre minDamage et maxDamage

Calcul des dégats finaux : $\text{baseDamage} * (1 - \text{Target.Armor}/200)$. Pour des raisons évidentes l'armor ne dépassera donc pas 150-170. Pas de diminution des dégats logarithmique pour le moment, on verra ça plus tard.

Maya est toujours invincible donc ne lancez pas la Méthode qui applique les dégats lorsqu'un ennemi l'attaque.

Chapitre VII

Exercice 04 : Mort et interface

	Exercice : 04
Exercice 04 : Mort et interface	
Dossier de rendu : <i>ex04/</i>	
Fichiers à rendre : La scène "ex00" et tout ce qui vous semble utile	
Fonctions interdites : Aucune	
Remarques : n/a	

Vous devez maintenant créer toute l'interface pour que le joueur puisse avoir un meilleur feedback sur les combats :

- Une barre de vie pour Maya, avec son nombre exact de points de vie affiché à côté.
- Une barre d'xp, qui se remplira au fur et à mesure que Maya gagnera de l'expérience. Vous devez aussi afficher sur cette barre les points d'xp actuels de Maya + "/" + les points d'xp nécessaires pour passer au niveau suivant.
- Le niveau actuel de Maya à côté de sa barre de vie.
- La barre de vie de l'ennemi sélectionné (voir ci-dessous) en haut de l'écran, ainsi que son nom et son niveau.

Un ennemi peut être "sélectionné" dans deux cas :


- Si on passe la souris dessus. Dans ce cas ses informations s'affichent en haut de l'écran. Elles doivent disparaître dès que le curseur de la souris n'est plus sur l'ennemi.
- Si Maya est en train de l'attaquer. Dans ce cas ses informations apparaissent également en haut de l'écran, mais doivent rester affichées (même si le curseur ne vise plus l'ennemi) jusqu'à ce que il soit mort, ou que Maya arrête d'attaquer.

Maintenant qu'on a enfin un feedback visuel de notre personnage on va enfin pouvoir la faire mourir ! oui parceque le God Mode ça va bien 5 minutes... Vous devez donc rendre Maya mortelle et lui faire perdre le bon nombre de points de vie lorsqu'elle se fait attaquer par les ennemis. Si Maya meurt vous devez afficher une interface/un message l'indiquant et surtout faire en sorte que rien ne plante !

Ne faites pas des ennemis qui frappent trop fort pour éviter que Maya se fasse tuer en 4 coups puisque pour le moment elle n'a aucun moyen de regagner de la vie, on voit ça dans l'exercice suivant. Idéalement il faudrait qu'elle tue un ennemi en 2-4 coups et qu'elle puisse encaisser une bonne vingtaine d'attaques pour avoir quelque chose de jouable.

Chapitre VIII

Exercice 05 : Level up !

	Exercice : 05
Exercice 05 : Level up !	
Dossier de rendu : <i>ex05/</i>	
Fichiers à rendre : La scène "ex00" et tout ce qui vous semble utile	
Fonctions interdites : Aucune	
Remarques : n/a	

C'est l'heure des petites touches finales. Commençons par finir le côté interface. Vous devez créer une fenêtre qu'on peut faire apparaître en appuyant sur "c" et qui rassemble les éléments suivant : Le nom du personnage, ses 3 caractéristiques principales, ses attaques min/max, son armure, son niveau, son xp, l'xp qu'il lui manque pour passer au niveau supérieur ainsi que ses crédits.

Vous devez donc maintenant gérer l'xp du joueur. A chaque fois que Maya tue un ennemi elle doit gagner de l'expérience et lorsque son xp atteint le niveau requis elle doit gagner un niveau. Dans ce cas vous devez créer un effet visuel pour que le joueur en soit informé de façon évidente (ne passez pas une heure dessus pour autant), vous devez également remonter la vie de Maya à son maximum et lui donner 5 points de compétence à répartir par le joueur. Son score d'xp est réinitialisé et le score à atteindre pour passer au niveau suivant augmente de 150 pourcents.

Un bouton cliquable doit apparaître sur le HUD à coté de la vie de Maya s'il y a des points de compétence restants à répartir. Lorsqu'on clique dessus la fenêtre du personnage apparaît, exactement comme si le joueur avait appuyé sur "c". Un bouton cliquable est présent à côté de chaque caractéristique principale. Lorsqu'on clique dessus un point de compétence est déduit du total disponible et est ajouté à la caractéristique correspondante au bouton cliqué.

Vous devez également gérer le niveau des ennemis (de façon simple). A chaque niveau supérieur à 1 l'ennemi doit avoir des stats boostées de 15 pourcent. Par exemple un ennemi niveau 3 aura 130 pourcent des stats d'un ennemi niveau 1.

Même si cette pratique est inintéressante d'un point de vue game design vous devez faire en sorte que les ennemis qui Spawnent soient du même niveau que Maya, pour avoir une progression dans le jeu rapide à mettre en place. Rappelez-vous qu'il s'agit d'une démo technique.

Enfin mettez en place un système de globe/potion de vie qui tombent aléatoirement quand un ennemi est tué. Ils soignent automatiquement Maya de 30 pourcent de sa vie max lorsqu'elle marche dessus.



Les exercices de cette journée (bien chargée!) s'achèvent ici mais s'il vous reste du temps vous êtes libres de mettre en place un système de loot et d'équipement d'objet. Vous pouvez également implémenter la gestion des crédits que les monstres droppent en mourant. Il y a déjà les stats toutes prêtes dans les scripts de Maya et des ennemis. Sachez juste que tout ce que vous ajoutez est du bonus perso et ne sera pas pris en compte lors de la correction.