



# Piscine Unity - Rush00

Hotline Miami

Staff [staff@staff.42.fr](mailto:staff@staff.42.fr)

*Résumé: Ce document contient les consignes pour le rush00 de la Piscine Unity de 42.*

# Table des matières

<b>I</b>	<b>Préambule</b>	<b>2</b>
<b>II</b>	<b>A propos de la collaboration</b>	<b>3</b>
<b>III</b>	<b>Consignes relatives à la journée</b>	<b>4</b>
<b>IV</b>	<b>Consignes generales</b>	<b>5</b>
<b>V</b>	<b>Partie obligatoire</b>	<b>7</b>
V.1	Gameplay général . . . . .	7
V.2	Les armes . . . . .	7
V.3	Les ennemis . . . . .	7
V.4	Les maps . . . . .	8
V.5	Le son . . . . .	8
V.6	Les menus . . . . .	9
<b>VI</b>	<b>Partie bonus</b>	<b>10</b>

# Chapitre I

## Préambule



FIGURE I.1 – Cette image représente la collaboration sur Unity. Unity c'est comme un Jenga géant où vous piqueriez les pièces de l'autre pour construire votre partie de la tour.



FIGURE I.2 – Cette image représente Unity pendant un merge.

Mais rassurez vous, vous trouverez dans le prochain chapitre quelques conseils pour bien aborder le travail en équipe.

# Chapitre II

## A propos de la collaboration

Il est enfin l'heure de votre premier projet en groupe avec Unity. Aussi puissant soit-il Unity n'est pas toujours une partie de plaisir lorsqu'il s'agit de collaboration. Afin de vous éviter de perdre des heures/jours de boulot et de nombreuses crises de nerfs voici quelques précieux conseils pour bien travailler en groupe.

- Comme on est sympa vous trouverez dans les assets de la journée un .gitignore qui vous permettant de ne pas push sur le depot les fichiers qui doivent rester local.
- Ne travaillez **JAMAIS** en même temps sur une même scène ou sur un même prefab. Unity a encore beaucoup du mal à gérer le merge sur ce type d'asset.
- Faites attention au code de votre binôme lorsque vous modifiez un objet contenant son script. Un tag en moins peut suffire pour détruire une fonctionnalité.
- Ayez une communication irréprochable avec votre binôme sur les modifications que vous apportez au projet.
- Separez bien votre travail, essayez de ne pas toucher aux même fonctionnalités/éléments de gameplay pour éviter un maximum les conflits.
- Faites de nombreux commits. Dans le cas d'un conflit vous pourrez revenir en arrière pour aller chercher une version qui marche.
- Mettez en place des "milestones", c'est à dire une version fonctionnelle de votre jeu aux étapes clés de son développement et synchronisez vos repos git locaux avant d'attaquer une nouvelle étape. Ainsi vous aurez une version fiable si l'un d'entre vous casse quelque chose et dans le pire des cas vous pourrez push votre dernière "milestone" valide et éviter d'avoir un projet non fonctionnel.

Sur ce, bon courage, vous en aurez bien besoin.

# Chapitre III

## Consignes relatives à la journée

- Lisez vraiment le sujet jusqu'au bout. Certains bonus peuvent être réalisés simplement si vous codez votre script de base correctement dès le début.
- Pour le rush la correction ne s'arrêtera pas lors d'une erreur cependant afin de vous y retrouver et de vous organiser correctement les points du sujet sont triés par ordre de priorité.
- Pour ce rush l'utilisation du C et de la minilibx **uniquement** est autorisé. Le code devra cependant être à la norme. Afin de vous avantager un peu, vous aurez le droit exceptionnellement à 6 fonctions par fichier.
- La consigne précédente est bien entendu une blague.

# Chapitre IV

## 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 coté 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.
- Meme si le sujet d'un exercice est relativement court, ca 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 V

## Partie obligatoire

### V.1 Gameplay général

Hotline Miami est un de ces jeux au gameplay très nerveux. Votre principal objectif sera donc de restituer cet esprit. Mais commençons par les bases. Vous devrez mettre en place un joueur que l'on peut déplacer dans toutes les directions avec WASD (W -> Haut, S -> Bas, A -> Gauche, D -> Droite). Le personnage regarde toujours en direction de la souris. La caméra doit suivre sa position mais pas sa rotation.

On peut récupérer une arme au sol en appuyant sur E et dropper celle qui est actuellement équipé en cliquant droit. On peut donc évidemment tirer en cliquant gauche. Tous les personnages ne possèdent qu'un seul point de vie. Si ils sont touchés ils meurent instantanément. L'objectif est d'arriver à la fin du niveau, ou de tuer tout le monde. Si l'on est tué on perd et on doit recommencer le niveau.

### V.2 Les armes

On doit pouvoir distinguer deux types d'armes :

- Les armes à feu, qui envoient donc des projectiles en direction de la souris. Elles ont des munitions (pas de chargeurs) et peuvent s'épuiser.
- Les armes blanches, sont des armes de corps à corps. Elle ne lancent donc pas de projectiles et ne sont pas limitées par des munitions.

Lorsque l'on drop une arme elle est lancée en direction de la souris de manière plus ou moins stylée. On attends de vous au minimum 4 armes différentes.

### V.3 Les ennemis

Les ennemis et leur intelligence artificielle sont le coeur de ce projet. Ce sont eux qui feront peser la balance et rendront votre jeu fun ou non. Il va donc falloir mettre en place les comportements suivants :

- Les ennemis peuvent voir le joueur loin devant eux et un petit peu derrière eux. Lorsqu'ils repèrent le joueur ils le poursuivent en lui tirant dessus. Leur poursuite s'arrête après un certain temps ou à la mort du joueur ou de l'ennemi.
- Les ennemis ne peuvent pas traverser les murs, ils connaissent l'emplacement des portes et peuvent poursuivre le joueur sur plusieurs salles sans problème.
- Si le joueur tire à proximité d'un ennemi, il doit pouvoir l'entendre et doit se mettre à poursuivre le joueur, même si il n'est pas dans la même salle.
- Les armes des ennemis n'ont évidemment pas de munitions, ils peuvent tirer autant qu'ils le souhaitent.
- Les ennemis ne peuvent pas se tuer entre eux.
- Certains ennemis doivent patrouiller sur un chemin prédéfini.

## V.4 Les maps

Vous devez au moins rendre un niveau jouable. Chaque niveau doit être composé de plusieurs pièces. Les pièces doivent être reliés par des portes. Les ennemis et joueur ne doivent pouvoir passer a travers aucun mur ou porte. Les portes peuvent donc s'ouvrir et se fermer pour pouvoir circuler librement entre les pièces. Essayez de faire en sorte que les niveaux soient fun à parcourir.

## V.5 Le son

Que serait **Hotline Miami** sans son, il va donc falloir une fois encore y accorder un soin particulier. Vous trouverez dans le pack d'asset de la journée toute la bande originale du jeu. Vous devrez donc mettre en place de la musique presente dans tout votre jeu. Essayez de la faire varier suivant les niveaux histoire de pas se lasser.



Vous ne devez utiliser que la bande originale du jeu. Non votre super musique techno n'ira pas mieux que la BO. En même temps qui aurait envie de mettre autre chose ?

Mais ce n'est pas tout vous devrez également mettre en place des bruitages pour les éléments suivants :

- Lorsque le joueur ou un ennemi est tué.
- Lorsqu'une arme est ramassée.
- Chaque arme doit avoir un son de tir différent et plus ou moins cohérent.

- Lorsque l'on gagne et lorsqu'on perd.

## V.6 Les menus

Afin de sublimer le tout vous allez rajouter des menus. On attends de vous les éléments suivants :

- Un Menu principal ultra stylé qui claque avec un logo animé utilisant la police de Hotline Miami. Ca doit piquer les yeux mais être beau à la fois. On doit pouvoir lancer le jeu ou le quitter depuis ici.
- Un menu contextuel qui s'ouvre au dessus de la scène lorsque l'on perd ou gagne la partie. Le texte doit évidemment changer en fonction de ce dernier paramètre.
- Un GUI in-game qui nous informe du nombre de munitions restantes.



Tout ca c'est beaucoup de travail et c'est plutot précis. Sachez que ce chapitre est trié par ordre de priorité afin que vous puissiez savoir ou vous en êtes et ce qui sera plus important lors de votre correction.

# Chapitre VI

## Partie bonus

Si vous êtes arrivés jusqu'au bout de la précédente partie déjà un grand bravo. Maintenant on va pouvoir ajouter des éléments de gameplay encore plus intéressants. Vous êtes globalement libre d'ajouter tout ce qui vous semble cool, qui peut améliorer l'ambiance, le fun de votre jeu.

Au cas ou vous n'auriez aucune idée en voila une bonne petite liste :

- La possibilité d'étourdir des ennemis en leur lançant l'arme dessus, ou en leur claquant la porte dessus.
- Plus de maps, plus d'armes, toujours plus.
- Des ennemis variés qui spawnent avec une arme et une tête aléatoire.
- Des sous types d'armes à munitions (longue portée, mitraillette, pistolet, ...).
- Le katana tue les ennemis lorsqu'on le lance sur eux.

Bref les possibilités sont nombreuses, faites vous plaisir et créer le **Hotline Miami** de vos rêves.