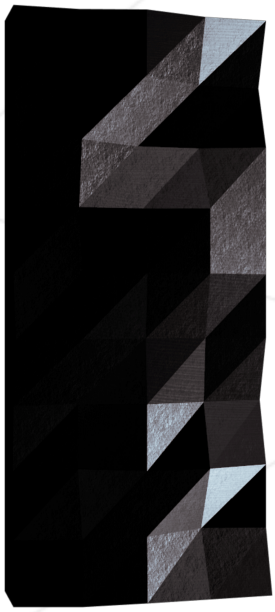

Piscine Unity - Rush01

Hack and Slash

Staff staff@staff.42.fr

Résumé: Ce document contient le sujet du rush01 de la piscine Unity de 42.

Table des matières

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 régulièremment.
- 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 defaut 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 builder 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 choisissez 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 evidement 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, faîtes-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 ! Refléchissez ! ! !
-

Chapitre II

Consignes spécifiques à la journée

Voici le dernier rush de votre piscine Unity, c'est le moment ou jamais de montrer tout ce que vous avez appris et tout ce que vous savez faire. Donc pour cette fois vous avez le droit à ABSOLUMENT TOUT ce que vous voulez. C'est à dire que vous avez le droit à : l'asset store dans son intégralité (oui oui même les scripts), tous les modèles 3d / sons / musiques / etc que vous pouvez trouver sur le net, ainsi que tous les assets que nous vous avons fourni depuis le début de la piscine.

Le but c'est juste de faire le jeu le plus stylé possible en deux jours, soit dit en passant, le temps moyen d'une GameJam.

Chapitre III

Préambule

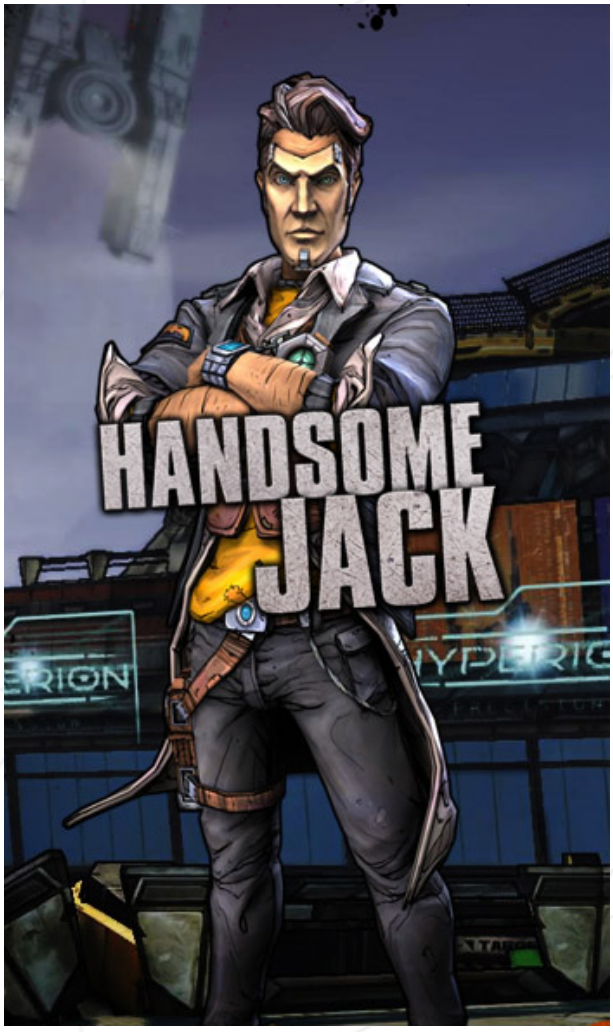

FIGURE III.1 – "Je me creuse la tête pour trouver un nom à ce poney en diamant que j'ai acheté. J'avais pensé à "Face de pet", en ton honneur. Mais bon, ça fait un peu immature. Que penses-tu de "Étalon du cul" ? Nan, c'est encore pire. Je vais y réfléchir un peu."

Chapitre IV

A propos de la collaboration - le retour

Voici un récapitulatif des bonnes pratiques à suivre pour travailler en groupe avec Unity. Afin de vous eviter 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.

Chapitre V

Partie obligatoire

J'imagine une légère frustration de votre part à la fin du d08 puisque nous avions une super présentation de Hack and Slash mais aucun gameplay, ce qui est plutôt triste il faut le dire. Pour remédier à celà, le rush final sera donc de terminer votre Hack and Slash en le rendant totalement classe. Par exemple en créant des vraies maps, un systeme de loot, des skills, un peu plus que 2 ennemis ... etc. Voyons tout ça en détails.

V.1 La base

On reprend tout à zero, voici la liste de ce qu'il faut dans votre hack and slash de base :

- Une map avec un navmesh.
- Une caméra en vue plongeante, fixée sur le héros contrôlé par le joueur (vous pouvez garder Maya ou prendre n'importe quel personnage, pourvu qu'il soit riggé/animable).
- Un héros qu'on contrôle à la souris : clic gauche sur le terrain pour se déplacer, clic gauche sur un ennemi pour l'attaquer, le héros attaque en boucle si on maintient le clic sur l'ennemi.
- Des ennemis, qui poursuivent et attaquent le héros si celui-ci entre dans leur zone de détection.
- Des animations de idle/course/attaque/mort pour tous les personnages présents dans le jeu.
- Un système de combat avancé basé sur des stats : force qui augmente les dégats, agilité qui augmente les chances de toucher et d'esquiver, constitution qui augmente la vie.
- Un système d'xp, chaque ennemi tué rapporte X points d'expérience. Lorsque le héros a suffisament d'xp il gagne un niveau et obtient 5 points de compétence à répartir dans ses stats principales.
- Les ennemis qui apparaissent doivent être du même niveau que le héros, que ce soit lvl 1 ou lvl 50.

- Un HUD qui affiche la vie du héros/de l'ennemi sélectionné, leurs niveaux respectifs ainsi que l'xp du héros.
- Une fenêtre de personnage qui reprend toutes les stats : FOR, AGI, CON, degatsMin, degatsMax, xp, xpToNextLvl, hpMax ...
- Des potions de vie que les ennemis laissent tomber aléatoirement et qui soignent le héros de 30 pourcents de sa vie max lorsqu'il marche dessus.

V.2 Les skills

Pour que notre Hack and Slash soit un peu plus intéressant qu'un simple clic sur les ennemis il va falloir y ajouter quelques skills ainsi qu'un super arbre de talents !

- A chaque Level Up, le héros gagnera aussi un point de talent. On peut en voir le total inutilisé dans une fenêtre d'arbre de talent.
- En appuyant sur la touche "N" l'arbre de talent s'ouvre. Il est possible de dépenser des points de talent pour débloquer des skills. L'arbre de talent est constitué de plusieurs paliers qui se déverrouillent tous les 6 niveaux. C'est à dire, du niveau 1 à 5 seuls les skills du premier palier peuvent être appris, du niveau 6 à 11, les skills du premier et second palier, ...
- Chaque skill peut être amélioré plusieurs fois (admettons 5). A chaque point de talent dépensé en plus du premier dans un skill, celui-ci augmente de niveau et devient plus efficace.
- Vous devez au minimum créer 6 skills dont au moins : une attaque de zone à poser au sol en cliquant (en utilisant un gabarit pour viser) avec la souris, une attaque de zone apparaissant autour du héros et suivant ses mouvements, un sort de dégats direct (style boule de feu), un skill passif pour améliorer un aspect du héros (hormis ses compétences de base), un sort de soin.
- Chaque skill doit avoir une tooltip expliquant son fonctionnement, ses dégats ..., ainsi que ses stats au niveau suivant (pour que le joueur sache s'il souhaite l'améliorer ou non).
- Vous pouvez créer une ressource spéciale pour vos skills, comme du mana ou de la rage, ou utiliser des skills sans ressources mais avec un temps de rechargement (le temps de rechargement est plus simple à mettre en place mais vous êtes libres de choisir ce qui vous plait le plus).
- Les skills doivent avoir un effet de particules ou autre pour que le joueur puisse avoir un retour visuel de leur zone d'action.
- Vous devez également créer une barre de skills, pouvant en contenir un nombre limité (admettons entre 4 et plus que 4), pour que le joueur doive faire un choix et ne puisse pas tout utiliser en permanence. Vous devez donc créer un moyen d'équiper ces skills sur cette barre, et de permettre au joueur de les lancer en appuyant sur les touches 1, 2, ...

V.3 Les loots

Un Hack and Slash sans loots, c'est un peu comme un pong sans les raquettes : c'est pourri. Donc on va maintenant rajouter les objets à ramasser et l'inventaire pour les stocker.

Vous devez créer un système d'objets équipables. Par objets on entend Armes pour commencer. Lorsqu'un ennemi meurt il a une chance de dropper une arme. Ces armes doivent être générées aléatoirement ; voici quelques consignes quant à cette génération :

- Tout d'abord les armes ne devront pas avoir toujours la même apparence. Le model de l'arme doit lui aussi être choisi aléatoirement parmis une liste de modèles prédéfinis.
- Chaque arme devra posséder au minimum des dommages et un vitesse d'attaque mais vous êtes libre de rajouter d'autres stats selon vos envies.
- Les stats de l'arme seront générées aléatoirement mais pas n'importe comment. Vous devrez prendre en compte le niveau actuel du joueur pour influencer la génération et les valeurs de ces stats. Les armes ne doivent donc pas avoir des stats disproportionnées face aux ennemis du niveau du joueur.
- Les armes doivent avoir une tooltip qui s'affiche lorsqu'on passe le curseur de la souris dessus, qu'elles soient au sol ou dans l'inventaire. Cette tooltip doit résumer les stats de l'arme.
- Créez également des niveaux de rareté. Certaines armes doivent être bien plus dures à obtenir que d'autres. Par exemple : Commun, Non-Commun, Magique, Légendaire.

Mais ce n'est pas tout. C'est cool d'avoir un générateur d'armes, c'est bien plus cool de pouvoir les équiper. Un inventaire va donc se révéler utile pour pouvoir transporter celles-ci. Vous devez créer un système qui permet de ramasser les armes en cliquant gauche dessus. Ces armes apparaissent alors dans la fenêtre d'inventaire accessible en appuyant sur la touche "i". Une fois dans l'inventaire on peut réorganiser les armes comme on le souhaite et changer l'arme actuellement équipée pour une autre. On doit également pouvoir jeter notre arme au sol en la glissant en dehors de l'inventaire. Bien entendu la taille de l'inventaire doit être limitée (12 cases semble être une bonne moyenne).

V.4 Un peu de variété

Maintenant que le système de jeu est un peu plus complet c'est le moment d'être créatifs ! Vous devez créer une map en extérieur, intéressante à parcourir, ainsi qu'un donjon sur plusieurs étages (admettons 3). Au dernier étage de ce donjon, vous devez créer un Boss qui possèdera beaucoup de vie, différents skills d'attaque (vous pouvez reprendre certains des skills du héros pour gagner du temps) et qui promettra un combat épique !

Ici on vous laisse carte blanche donc éclatez-vous !

V.5 La bande son

Pour rendre le jeu encore plus immersif vous devez créer sa bande son. Encore une fois nous vous laissons carte blanche, donc choisissez les bruitages et les musiques qui vous plaisent pour donner un max de style à votre jeu ! C'est aussi le moment de tester les fameuses reverb zones dont nous avons parlé plus tôt dans la semaine.

V.6 Un cheat code pour les gouverner tous

L'intérêt d'un Hack and Slash c'est de progresser sur plusieurs niveau. Cependant pour gagner du temps et éviter que les gens aient à jouer 2 heures à votre jeu pour tester les skills, implémentez une touche de triche qui permet de level up instantanément, et une autre pour faire apparaître une arme du niveau du héros. Votre jeu doit être équilibré, que ce soit au niveau 1, 20 ou 50. D'ailleurs les corrections demanderont de tester le jeu à plusieurs niveaux d'avancement du héros.

Chapitre VI

Partie bonus

Si vous avez fait tout ce qui était demandé et que le rush n'est toujours pas fini, vous êtes `DES GRANDS MALADES !`

Mais si vous avez encore faim, voici quelques idées de bonus possibles :

- Des objets lootables autre que les armes, par exemple des armures, casques, bottes, anneaux, amulettes, ...
- Différentes classes de personnage.
- Des armes à distance.
- Des effets élémentaires sur les armes, par exemple une épée enflammée avec des effets de particules qui vont bien, et des stats en conséquence !
- La gestion de la monnaie : les ennemis droppent des crédits qui permettent d'acheter des items aux NPC/Distributeurs ...
- Un écran de titre et un système de sauvegarde gardant en mémoire la progression du joueur pour pouvoir recommencer la partie en gardant son niveau et son équipement.
- Implémentez "Étalon du Cul".