



Rapport de stage à Incenti Studio

Programmeur Gameplay sur le jeu EtherShelter

Stage M1 de 10 semaines, du 18 Juin au 30 Aout 2018

Formation : Master « jeux et médias interactifs numériques » cohabité par le conservatoire national des arts et métiers, l'Université de Poitiers et l'Université de La Rochelle.

Maitre de stage
Localisation

BEYNET Maxime
3, 236 Tower Road, Sliema, Malte



Sommaire



1. Remerciement
2. Introduction
 - a. L'Entreprise
 - b. Contexte du stage
 - c. Mes motivations
 - d. Mes objectifs à Incenti
3. Développement
 - a. La blockchain
 - b. EtherShelter
 - c. Le générateur & Spine
 - d. Les ColorMasks
 - e. Travailler avec la blockchain
 - f. La scène de combat
4. Conclusion



1. Remerciement



Pour commencer ce rapport en bon et due forme, je souhaite remercier les trois personnes qui m'ont permis d'effectuer ce superbe stage à Malte (dans l'ordre alphabétique pour ne pas faire de jaloux) : BENYAMINA Adrian, BEYNET Coralie et Maxime.

Ces personnes m'ont donné un cadre de travail, un bureau, un logement, prépayé le voyage aller-retour ainsi que la nourriture le midi. Autant dire qu'il n'y a pas meilleures conditions pour débiter son stage de M1 à l'étranger !

De plus, merci à Adrian pour sa disponibilité et ses conseils techniques qu'il m'a dispensé tout au long de mon parcours à Incenti, s'occupant en grande partie de mon suivi pédagogique. De plus, merci à Zameran, le lead programmeur de la boîte pour être en quelque sorte mon référent techniques.

D'une façon générale, je remercie tous les membres de Incenti pour leur accueil et leur bonne humeur, ainsi que mes collègues Benjamin, Dimitri, Flo, Lola, Marine et Romain avec qui j'ai travaillé !



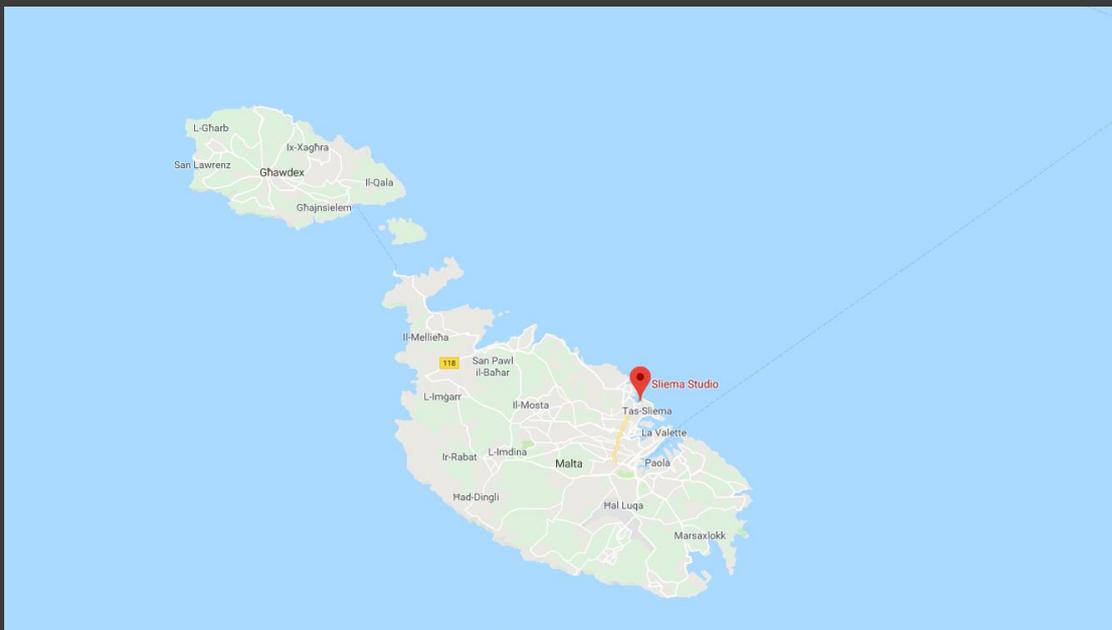
2. Introduction

a. L'Entreprise

Incenti Studio unifie des Développeurs, Game Designers et Artistes venant de tous les continents. Leur objectif est de créer et distribuer des Jeux et Applications décentralisés à l'aide de la Blockchain. La Crypto-économie est au cœur du processus de design de l'entreprise, et utilise des Contrats Intelligents (Smart Contracts), des Crypto Monnaies et des interactions Peer To Peer afin de créer des nouvelles façons d'interagir et de jouer. Le sujet phare est la décentralisation des services et des monnaies ouvertes et libres pourrait améliorer la société d'une manière sans précédent, via des jeux & médias interactif, en phase avec l'intitulé de notre formation à l'ENJMIN : Master « jeux et médias interactifs numériques »

b. Contexte du stage

Le studio est né en France à l'initiative de Maxime Beynet et Adrian Benyamina des investisseurs et supporteurs du monde de la Blockchain. L'entreprise se développe à Malte, un Pays ouvert aux technologies Blockchain et aux entrepreneurs.





c. Mes motivations

En tant que futur programmeur dans un studio de jeu vidéo, je souhaite dans un premier temps avoir une expérience supplémentaire dans l'industrie : mettre mon cœur de métier en pratique avec des professionnels, tout en pouvant apporter mes compétences, ainsi que mes avis artistiques et créatifs.

Dans un second temps, je pense que notre système monétaire & capitaliste n'est pas viable au long terme pour notre prospérité, tant au niveau des inégalités que pour l'écosystème de la planète. Sans partir sur un avis politisé qui n'est pas le débat ici, je pense qu'une économie décentralisée utilisant la blockchain ou une autre forme de technologie semblable peut émerger, mettant un terme, au moins partiellement aux systèmes bancaires centralisés actuels. En effet aujourd'hui nous avons la technologie pour permettre de ne pas mettre entre les mains d'une minorité l'argent d'une majorité.

d. Mes objectifs à Incenti

Le studio à 2 projets de jeux vidéo : **NODA**, et **EtherShelter**. Le studio a pour but de créer un jeu de type MMO en 3D (Massivement Multijoueur En ligne) appelé NODA, utilisant la technologie de la blockchain. Pour ce faire, Incenti a décidé tout d'abord de commencer petit : créer un jeu à *scope* réduit, appelé **EtherShelter**, en 2D, qui aura une base de code semblable au niveau *smart contract* : le but est de tester la viabilité d'un jeu multijoueur basé sur une blockchain publique. Mon but dans le jeu sera de participer à la création du gameplay du jeu, ainsi qu'à la mise en place de divers outils qui nous permettrons de créer un tel jeu.



3. Développement

a. La blockchain

Tout d'abord, un petit mot sur ce qu'est la blockchain. La blockchain est un système permettant de réaliser une transaction entre deux parties via un registre informatique d'informations, ouvert à tous et infalsifiable.

b. EtherShelter

Le but de EtherShelter est de tester la création d'un jeu vidéo, basé sur la technologie de la blockchain. Nous nous basons sur la monnaie ethereum pour les échanges entre les joueurs.

Le jeu est actuellement encore en cours de création et son Game Design n'est pas encore définitif. Je vais expliquer ce que j'ai compris de l'ensemble du jeu.

Le jeu contient une carte global multijoueur avec des ressources uniques, premièrement générés aléatoirement. Ces ressources et bâtiments peuvent être achetés via de l'éther (la monnaie ethereum). Un joueur débute avec des personnages générés aléatoirement. Avec ses personnages, il peut attaquer des cases de la carte pour espérer remporter le territoire et du *loot*. Toutes les actions du joueur ont un prix minime en éther, et toutes les ressources gagnées des joueurs peuvent être revendu à d'autres joueurs, ou même revendu en dehors du jeu en éther ou en euro. Le but, c'est de lancer le jeu avec des règles sur la blockchain ethereum publique, et de laisser tourner, ne pouvant plus changer la blockchain une fois le jeu lancé, impossible pour nous de modifier les règles de jeu, de récupérer l'argent des joueurs. La blockchain est open source, notre jeu fait sur Unity n'est qu'une interface, qui peut être remplacé par des mordeurs si la communauté le souhaite.

Adrian s'est occupé de l'écriture des « Smart Contrat » de la blockchain, aidé par Zameran, le lead programmeur. Mon but dans ce projet, est d'aider à mettre en place les outils, créer le générateur de personnages, et créer le système de combat du jeu.

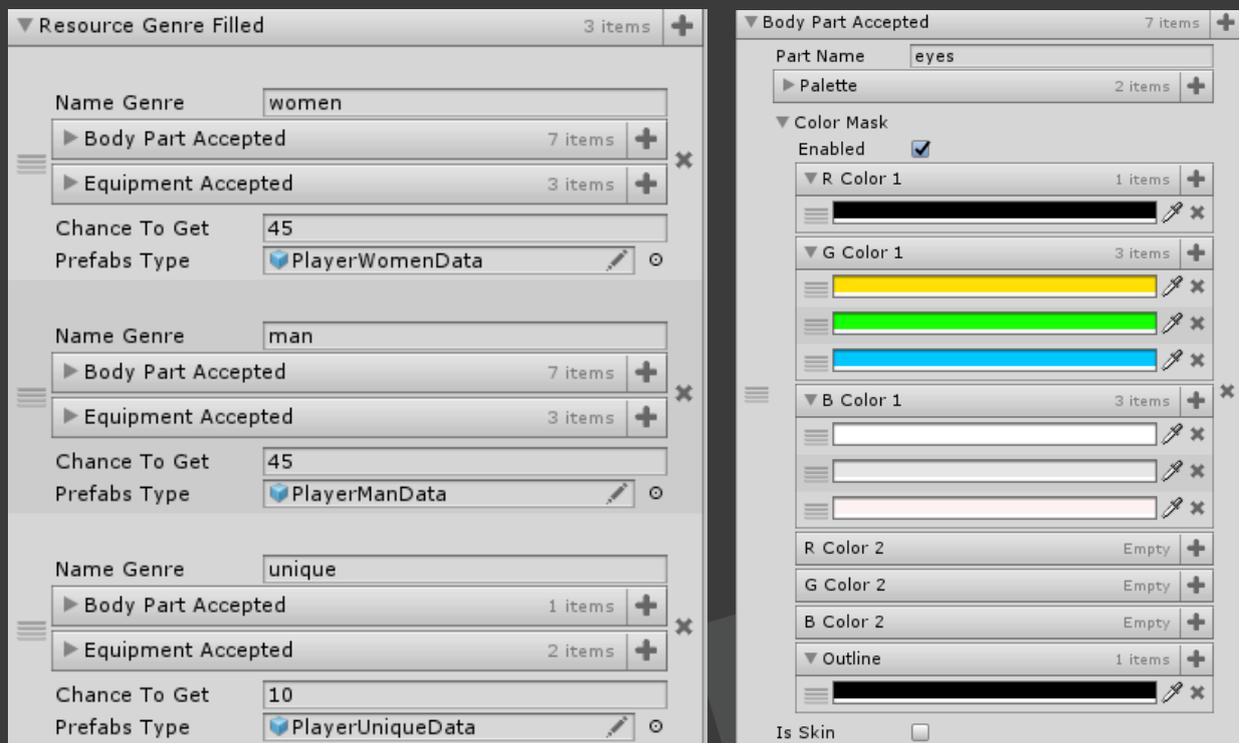


c. Le générateur et Spine

La plus grande partie de mon temps de travail a été consacré à créer un générateur de personnage. Depuis une liste d'assets graphiques de têtes de personnages (yeux, oreilles, nez), je dois pouvoir former aléatoirement un personnage sur un corps. Avec comme restriction : le générateur doit fonctionner avec un personnage généré depuis Spine : un logiciel d'animation 2D pour les jeux vidéo.

Dans un premier temps il a fallu travailler avec les graphistes pour définir un style de personnage commun, puis pour mettre en place la logistique entre leur logiciel de dessin (Photoshop) et Spine.

Dans un second temps, il m'a fallu prévoir l'architecture des dossiers où récupérer les images ainsi que la gestion des données à utiliser pour faire le générateur. Nous avons utilisé Odin, qui permet de sérialiser des données dans l'éditeur de Unity avec une grande facilité.





Ainsi, le générateur prévoit 3 types de personnages : un homme, une femme, ou un personnage prédéfini. Chaque personnage va avoir une liste de partie du corps à définir, et colorer. Ces parties du corps ont comme informations : nom, couleurs à choisir aléatoirement, est-ce que cette partie est une couleur de peau ? En effet, si nous générons aléatoirement des couleurs sur les parties du corps, il faut uniformiser la couleur de peau, pour ne pas avoir une oreille rouge et un nez bleu. Je reparlerai des ColorMasks, notre méthode de gestion des couleurs plus bas.

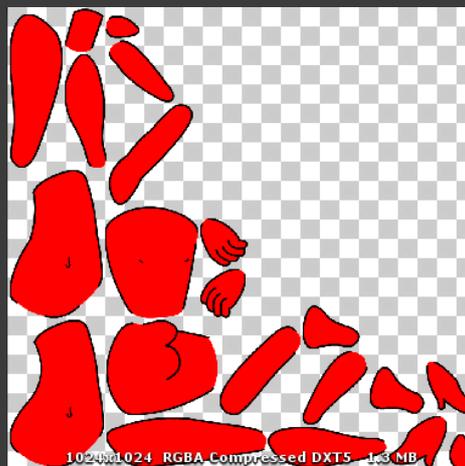
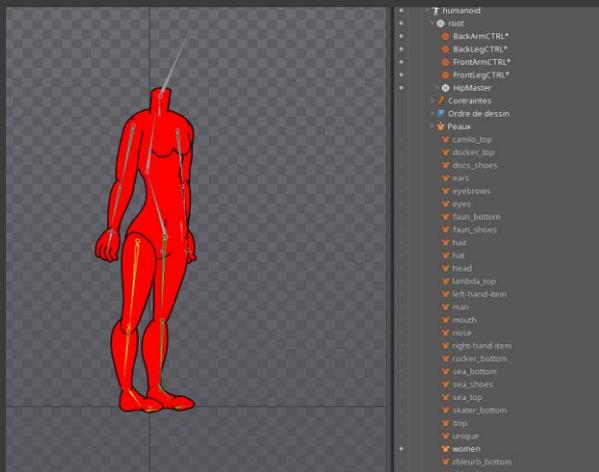
Dans un troisième temps : j'aimerais parler de notre méthode d'aléatoire en utilisant des *hashs*. La blockchain est censée être sécurisée, de ce fait je ne peux pas utiliser n'importe quel type d'aléatoire. J'utilise donc des méthodes sécurisées propres à ethereum (*Sha3Keccak*). Sans entrer dans les détails, ce que cela veut dire pour moi : à partir d'une chaîne de caractère quelconque, j'obtiens toujours le même résultat généré. Ainsi, à partir d'une chaîne du genre : « 3483423df4c6ea », j'obtiens toujours le même personnage à la fin : nez, bouche, yeux, couleur etc. Pratique pour tester.



Dans un quatrième temps, j'ai dû apprendre le fonctionnement de Spine par rapport à l'intégration dans Unity. Au niveau du logiciel lui-même d'abord, puis au niveau de l'API de Spine et du code.

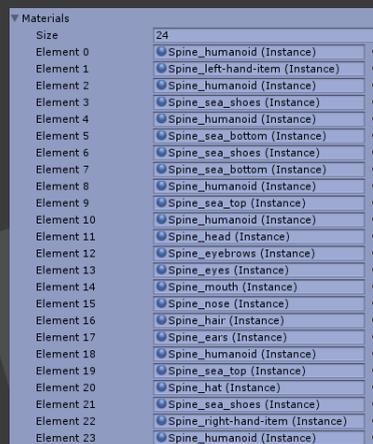
La plus grande difficulté était de trouver comment changer d'image dans le skeleton généré par Spine de façon dynamique, et rapide. Pour ce faire, j'ai passé beaucoup de temps et de recherches dans la documentation et le code source de Spine.

« On a qu'à faire que des skins », l'une des premières choses que m'a dites Adrian lorsque j'ai commencé l'intégration Spine, pour blaguer. Vous allez comprendre. Après 1 mois et plus de développement, nous avons un générateur fonctionnel, mais lent (une seconde pour générer l'ensemble d'un *skeleton* (d'un personnage venant de Spine)).

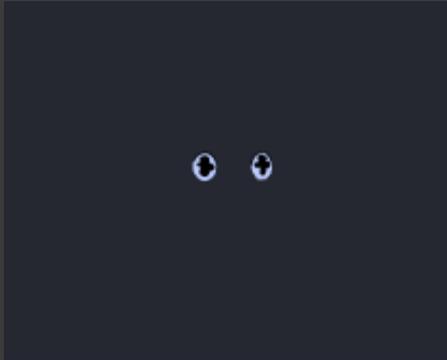


En effet, un *skeleton* venant de Spine est fait de multiple Skin (à gauche nous voyons le skin humanoïde femme). Chaque skin est « packé » par Spine dans un atlas (à droite). La méthode consistait pour le générateur à « dépacker » en runtime, changer individuellement les yeux, les oreilles et autres éléments de cet atlas, pour éventuellement repacker ensuite. Ce procédé était long et inefficace pour une gestion correcte des données. (Le simple fait de changer les yeux de « ouvert » à « fermé » demande ce processus de depacking / repacking).

La méthode a donc été de mettre dans des skins séparés tous les éléments susceptibles d'être changé. Ensuite dans Unity, selon les données générées par le générateur, je crée un skeleton, en combinant des skins : je combine le skin « yeux », « nez », « bouche », « tête », « corps femme », et éventuellement plusieurs autres skins de vêtement, d'arme. Dans cet ordre. Voici un exemple de résultat :



A droite, l'ensemble des matériaux du skeleton, chaque matériaux correspondent à un skin particulier. En dessous, 3 atlas de 3 skins :



Un skin « yeux » contenant uniquement les yeux (car élément modifiable par le générateur)



Un skin « humainoid » contenant le corps homme/femme



Un skin « sea_bottom », qui correspond à une partie de vêtement

Ces matériaux vont me servir ensuite pour changer rapidement les couleurs et images instantanément en jeu. Changer d'image ou les propriétés d'un shader dans un Materials est en effet beaucoup plus rapide que de faire de la manipulation d'atlas comme avant ! La méthode peu donc paraître lourde au premier abord en quantité d'atlas et de matériaux, mais nous gagnons en rapidité en runtime, et nos *drawcalls* sont dans la norme.

Ainsi, nous pouvons générer aléatoirement et rapidement une infinité de personnages variés avec peu d'images de base !





d. Les ColorMasks

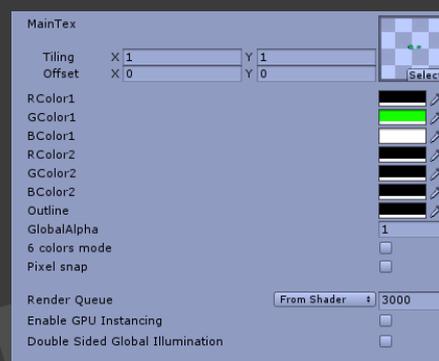
Nous voulions pouvoir colorer aléatoirement en plusieurs couleurs différentes parties d'une même image, et bien sûr, pouvoir définir quels sont les parties à colorer, avec quelques types de couleurs. Nous avons utilisé un shader créé par Benjamin. L'idée est simple : avoir des images avec 7 channels RGB précis et définis, pour ensuite pouvoir les changer indépendamment.

Celui-ci nous a posé par mal de soucis techniques et de coordinations : en effet, voici ce qu'a demandé un tel shader dans notre pipeline :

- Préparer les images à l'avance sur Photoshop en mettant les 7 channels qu'il nous faut
- Avoir les bons settings dans Spine, l'exportation a posé son lot de soucis de rendu : en effet le *packer* de Spine crée ses propres atlas de texture avec nos images.
- Gérer les données sur Unity pour les colorations et le générateur : il faut que le générateur puisse savoir quel type de couleur mettre où et comment. (Ne pas avoir de violet à la place du blanc des yeux, mais colorer entièrement les cheveux d'une même couleur par exemple)
- Garder des images propres à l'écran : Nos images doivent gérer les mip map, le filtre bilinéaire ainsi qu'une compression forte, avec un Aniso level à 16. Toute ces propriétés ont été difficiles à garder en utilisant notre shader.



Les yeux avec trois couleurs générées aléatoirement selon les règles des colorations des yeux



Le shader



e. Travailler avec la blockchain

Dans mon stage, je n'ai pas travaillé directement sur la blockchain, mais je l'ai utilisé. En effet, le pipeline Spine-Unity une fois mis en place, j'étais prêt à utiliser la blockchain pour faire fonctionner mon générateur.

L'utilisation de la blockchain me fait très fort écho au développement network, avec un client, et un serveur. Le lancement d'une blockchain locale se fait via un simple double clic sur un fichier bat. Ensuite j'écoute les données traduite par Adrian qu'il m'envoie toute les secondes, et avec ça je peux travailler sur mon client, l'interface Unity.

f. La scène de combat

Le générateur enfin fonctionnel ainsi que le pipeline Spine-Unity, j'ai pu m'atteler à la création de la scène de combat. C'est la dernière tâche sur laquelle je vais travailler à Incenti, et elle ne sera pas terminée, je dois donc mettre au propre le plus possible. Ici rien d'intéressant à dire, il y a juste beaucoup de donnée à gérer et beaucoup d'UI à relier ensemble.





4. Conclusion

À L'heure où j'écris ce rapport, je suis encore en stage pour 10 jours. Cependant, je pense avoir eu assez de détails pour en faire une conclusion.

Incenti est un studio de jeu vidéo très ouvert, qui peut se permettre, contrairement à d'autre, de tester, d'échouer et de recommencer. J'ai trouvé incroyable que le studio se base actuellement sur l'ethereum, une technologie par encore approuvée, sujet aux bugs et aux changements politiques, qui évolue et fait des mises à jour très fréquente.

C'est une chance de pouvoir travailler dans de telles circonstances, dans un studio qui se permet d'innover sur des technologies encore totalement nouvelles, aux futurs incertains, sans avoir peur de l'échec. Cela donne une ambiance sereine où travailler. De plus, j'ai trouvé passionnant à quel point notre point de vue importait aux membres du studio, que ce soit de ma part, ou des avis des graphistes, une grosse partie de la production et du jeu pouvait de ce fait être modifiée si un avis pertinent arrivait sur la table. Incenti a été une excellente expérience, et j'espère laisser derrière moi une base de code solide, et un bon souvenir à l'équipe.

