

Rapport de Soutenance 2

Evasion

- Les Professionnels/ 2013-2014 -



Lenny "*Le Noob*" Danino - danino_l

Louis "*El Parain*" Kédémos - kedemo_l

Anatole "*Totonut*" Moreau - moreau_a

Khalis Chalabi - chalab_k



Table des matières

1	Introduction	1
2	Avancements	2
2.1	Louis " <i>El Parrain</i> " Kedemos	2
2.1.1	Expérience personnelle	2
2.1.2	Avancement de la 3D	2
2.1.3	Travail sur les collisions	4
2.2	Khalis Chalabi	5
2.2.1	Avancement du projet	5
2.3	Lenny " <i>Le Noob</i> " Danino	6
2.3.1	Personnages/Décors	6
2.3.2	Et après ?	10
2.4	Anatole " <i>Totonut</i> " Moreau	11
3	Conclusion	12

1 Introduction

2 Avancements

2.1 Louis "*El Parrain*" Kedemos

2.1.1 Expérience personnelle

Avant cette soutenance, nous avons disposé de deux semaines de vacances, ou du moins, de deux semaines sans cours. Je pense que cela a été bénéfique pour réaliser une grosse avancée dans notre projet. En effet, nous n'avons pas eu cette fois-ci le stress lié aux révisions ou aux cours. Se réunir pendant ces deux semaines a été plus facile qu'en période de cours.

Comme à la première soutenance, j'ai dû laisser de côté certaines idées de réalisations. J'espère avoir le temps, pour la troisième soutenance, de les réaliser. En revanche, certaines choses, que je n'avais pas prévu à l'origine, ont été développées.

2.1.2 Avancement de la 3D

Notre jeu est développé en 3D. Mettre de côté ce point pendant le développement du jeu est impensable. C'est pourquoi de gros progrès ont été réalisés à ce niveau. Au moment de la première soutenance, pour créer un personnage ou un mur en 3D, il fallait faire tout un tas de déclarations dans le fichier du jeu principal. Il devenait urgent de rendre la création 3D plus facile, pour permettre des phases de tests le plus tôt possible. J'ai donc écrit des classes qui rendent possible l'instanciation, l'affichage et la manipulation des modèles 3D aisés :

```
Evasion.Affichage._3D.Perso_Model michael;  
michael = new Affichage._3D.Perso_Model(Content, new Vector3(20,  
0, 20), viewMatrix, aspectRatio, graphics, 1);  
michael.Update();  
michael.Draw();
```

Ces quatre lignes, placées au bon endroit, nous permettent de manipuler un personnage en 3D. On peut le faire se déplacer dans les quatre directions, on peut le faire tourner sur lui même.

Les déplacements ont été l'un des principaux problèmes que j'ai rencontré. Pour modifier la position ou la rotation d'un modèle dans son espace 3D, l'utilisation de matrices est nécessaire. Il faut effectuer plusieurs produits matriciels à la suite. En premier, on multiplie la matrice de position du squelette du modèle par l'échelle souhaitée. Cela permet un redimensionnement. On multiplie ensuite le résultat par trois matrices de rotation, une pour chaque axe. Enfin, on multiplie le résultat par la translation que l'on souhaite faire faire au modèle. Tout cela permet d'obtenir une bonne rotation et une bonne translation du modèle 3D. Or, le produit matriciel n'est pas commutatif. C'est à dire que pour deux matrices A et B, le produit AB est différent du produit BA. Pour

la première soutenance, je n'avais pas suivi le même ordre que celui énoncé précédemment. Le résultat obtenu n'était donc pas celui espéré. Les cours de math qui ont suivi la première soutenance portaient sur les matrices. J'ai ainsi pu me rendre compte de mon erreur. La première illustration montre comment le personnage se déplaçait lors de la première soutenance :

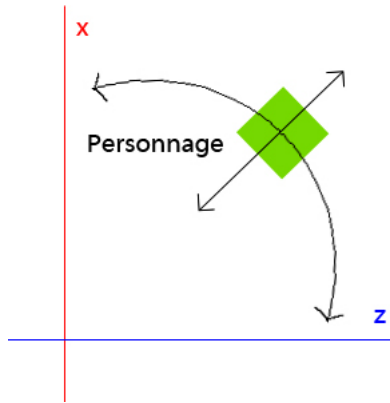


FIGURE 1 – Déplacement première soutenance

On peut voir que le personnage ne pouvait que se rapprocher ou s'éloigner du centre et tourner par rapport au centre. Régler le problème des matrices a permis de rendre le déplacement plus naturel. Le personnage peut maintenant se déplacer dans quatre directions et tourner sur lui-même. La seconde figure illustre ces déplacements :

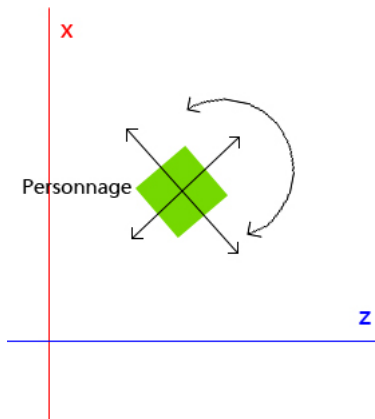


FIGURE 2 – Déplacement deuxième soutenance

2.1.3 Travail sur les collisions

Après avoir régler l’affichage et le déplacement des modèles, il a fallut s’occuper des collisions. Dans le monde du jeu vidéo, les collisions sont ce qu’il y a de plus important. Si les collisions sont mal gérées, lors d’un combat par exemple, l’ennemi peut nous frapper mais pas le contraire.

Je suis allé chercher des informations concernant les collisions sur internet. Les résultats ne manquaient pas. De nombreuses techniques sont expliquées. Il y a par exemple celle utilisant des BOUNDINGBOX. Des coordonnées définissent une boîte en trois dimensions contenant le modèle. Ensuite une méthode permet de savoir si deux BOUNDINGBOX sont en collisions. Une autre méthode consiste à utiliser les coordonnées d’un modèle et à regarder où il se situe par rapport aux autres modèles. Si deux coordonnées sont trop proches, alors on estime qu’il y a collision.

Nous avons choisi de retenir la méthode utilisant les BOUNDINGBOX. Elle permet de tester les collisions de manière souple et rapide. Voici une illustration qui montre comment sont gérés les BOUNDINGBOX par XNA :

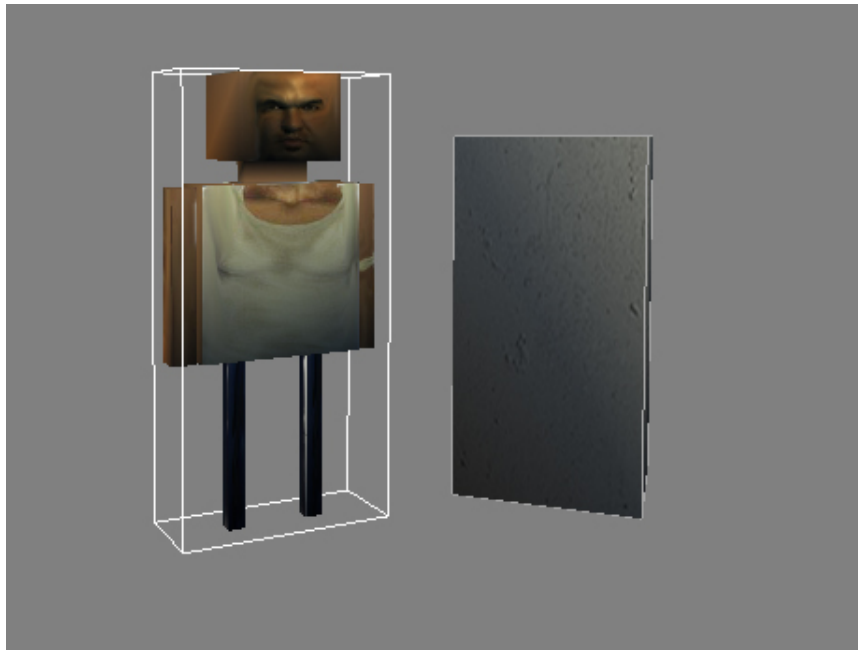


FIGURE 3 – Gestion des BOUNDINGBOX

Les tr

2.2 Khalis Chalabi

2.2.1 Avancement du projet

2.3 Lenny "Le Noob" Danino

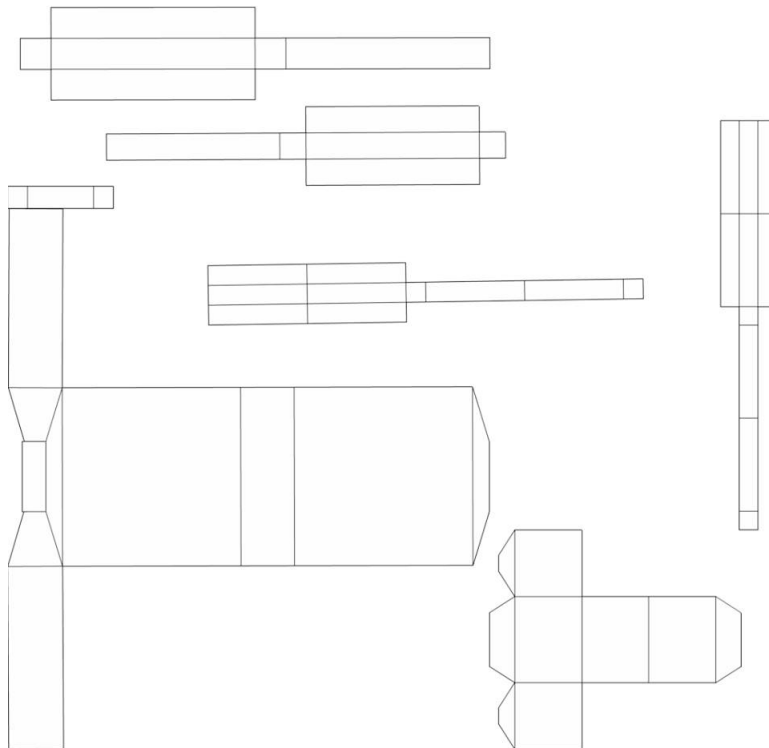
2.3.1 Personnages/Décors

Les textures des personnages sont des éléments très important dans le jeu car elles sont ce qui est immédiatement regardé par les joueurs. Il faut qu'elles soient jolies et agréables sinon les joueurs ne voudront pas jouer au jeu. Ainsi nous avons décidé qu'il serait préférable de dessiner par-dessus des textures provenant d'images sur internet. Cela apporte un degré de réalisme non négligeable.

Réalisations :

Les textures et le code :

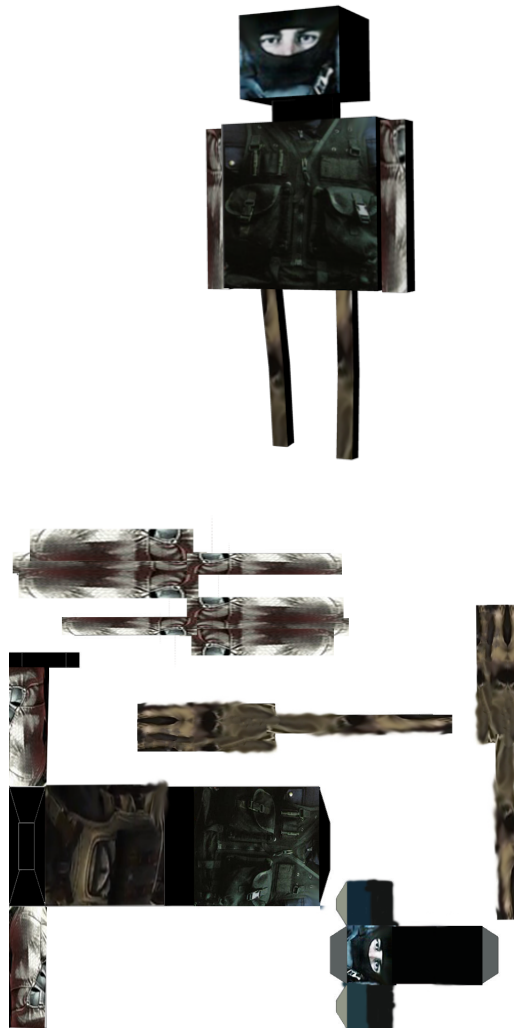
Premièrement il a fallu réaliser un modèle de base pour les personnages découpé avec Blender. Ce logiciel est extrêmement utile mais je n'y avais jamais touché et il m'a donc fallu l'aide de Louis et de tutos pour parvenir à me débrouiller suffisamment. Nous avons préféré rendre notre personnage cubique. C'était intéressant à faire car cela nécessitait de visualiser l'ensemble dans l'espace comme lorsqu'on découpe un cube.



En me basant sur le modèle j'ai pu créer les textures de 2 personnages. J'ai utilisé Photoshop, qui comme Blender m'était entièrement inconnu, pour la correction des images et la découpe. Puis j'ai visualisé le résultat avec Blender. Je vous présente donc le Pnj des prisonniers :

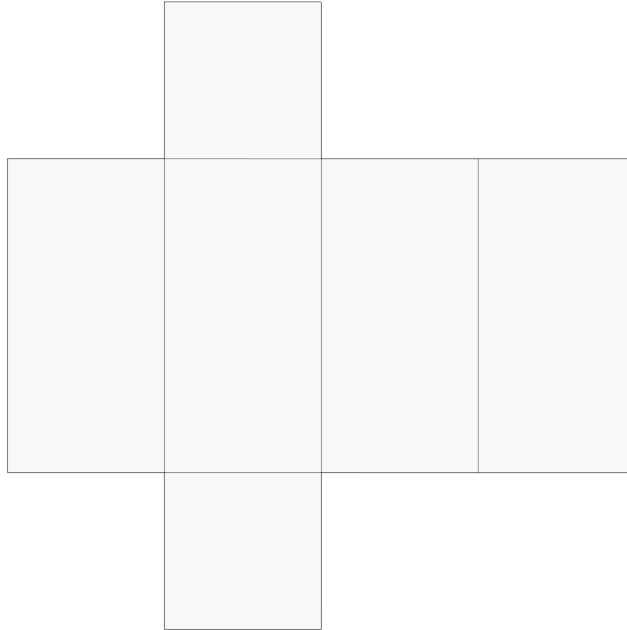


Et voici un des ennemis en dehors de la prison :



Cependant dessiner des personnages reste un aspect graphique. J'ai dû en effet toucher au code des classes pour pouvoir les insérer. Avec l'aide de Khalis, qui s'est occupé d'autres personnages, j'ai donc terminé les classes Pnj, Objets Utilitaires et Outils. De même je m'étais occupé dans la première soutenance des classes Décors, Murs et Objets. Pour parfaitement les terminer j'ai dû effectuer les mêmes étapes que pour les personnages.

Voici le modèle des murs :



Voici une des textures utilisées :



2.3.2 Et après ?

Pour notre troisième soutenance je vais me fixer quelques objectifs à atteindre :

Le multijoueur

Je compte terminer le multijoueur car même si aujourd’hui il est fonctionnel, je compte l’améliorer. Cela me permettra de progresser dans le code aussi.

Les sons

Je compte rajouter des sons pour que l’ambiance du jeu soit complète lors de la troisième soutenance avec l’aide d’Anatole.

L’éditeur de map

Enfin, je compte participer à l’édition des maps pour donner une dimension esthétique à notre jeu.

Remerciements

Comment ne pas terminer encore une fois sur des remerciements à mon groupe qui reste soudé malgré nos différents ? C’est donc pour cela que je remercie notre leader Louis qui soude quotidiennement notre équipe et nous encourage lorsque certains problèmes sont rencontrés, notre visionnaire Anatole qui aide à la projection du projet ainsi qu’à son aboutissement et toujours Khalis qui amène la bonne humeur tous les jours.

2.4 Anatole "*Totonut*" Moreau

	Première soutenance	Deuxième soutenance	Troisième soutenance
Codage décors	70%	100%	100%
Codage objets	70%	100%	100%
Codage personnages	70%	90%	100%
Graphismes 2D/3D	-	60%	100%
Site web	70%	100%	100%
Son	20%	80%	100%
Collisions	-	50%	100%
Affichage	70%	100%	100%
Boucle de jeu	-	50%	100%
Interaction entre éléments du jeu	20%	70%	100%
Menu	50%	100%	100%
Réseau	-	100%	100%
Multijoueur	-	60%	100%

3 Conclusion