

Rapport de Soutenance 2

- 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	Louis Kédemos	1
1.1	Présentation	1
1.2	Réalisations	2
1.2.1	Animations 3D	2
1.2.2	Création d'une intelligence artificielle	4
1.2.3	Utilisation d'objets	5
1.3	Impressions personnelles	6
1.3.1	6

1 Louis Kédémos

1.1 Présentation

Portant un intérêt prépondérant à l'informatique, j'ai intégré l'école d'ingénieur informatique EPITA en 2013. L'informatique est un domaine que je côtoie depuis de longues années. En revanche, c'est la première fois que je réalise un projet d'une telle ampleur. Jusqu'à il y a six mois, je ne connaissais que le travail individuel, de durée n'excédant jamais une semaine. Ce projet a présenté de nouveaux défis, auxquels je n'ai jamais été confronté. Ma première année à EPITA m'a permis de découvrir le côté "intellectuel" de l'informatique. Avant d'entrer dans l'école, je ne connaissais que la partie concernant le code, la programmation. Que du travail brut, pour ainsi dire.

Parallèlement à mes activités d'étudiant, je me consacre aussi à la pratique du sport et de la photographie. J'ai notamment rejoint l'association de photo de EPITA, *le club Ephemere*. Ils m'ont permis d'améliorer ma technique et de ne pas abandonner la photographie. C'est aussi durant cette première année à EPITA que j'ai commencé à exercer une "vraie" activité de photographe. J'ai en effet commencé à couvrir en tant que photographe de grands événements. Ma motivation pour cet art n'a que grandi durant cette année.

Pour revenir au projet, ce dernier m'a beaucoup fait progressé en matière d'informatique et de relations humaines. J'ai toujours aimé les défis, et ce projet en a présenté de nombreux. A chaque nouveau problème ou occasion, j'ai essayé de me surpasser. Et j'espère avoir réussi.

1.2 Réalisations

Pour cette troisième soutenance, nous n'avons pas eu assez de temps pour ajouter de nouvelles fonctionnalités à notre jeu. Nous avons dû en effet répartir notre énergie à nos partiels, nos cours et au projet. En conséquence, peu d'évolution majeures ont fait leurs apparitions entre la deuxième et la troisième soutenance. Dans la suite de cette partie, je détaille le travail que j'ai effectué, seul ou à plusieurs.

1.2.1 Animations 3D

Notre deuxième soutenance a vu l'apparition d'un générateur de carte et de l'affichage de carte. La gestion des différents éléments en 3D a aussi été rendue plus simple. Mais il manquait une chose essentielle : l'animation 3D. Les personnages 3D, lors de leurs déplacements, ne sont pas animés. Les jambes, les bras ne bougent pas. Les personnages sont simplement translatés dans l'espace. Ma première tâche a été de réaliser les animations des personnages. Une tâche ardue qui a requis de nombreuses heures de travail. Pour cela, j'ai utilisé le logiciel de modélisation 3D Blender. Il s'agit d'un logiciel libre assez simple qui permet une bonne introduction au graphisme 3D. L'animation des personnages se résume seulement à un pas complet. C'est à dire coordonner le mouvement de chaque membre du corps : l'avancement de chaque jambe et le contre-balancement des bras. Il suffit ensuite de jouer cette animation en boucle pour donner l'impression d'une marche. Chaque type de personnage à son type de marche. J'ai donc animé différemment le personnage principale, les gardiens et les autres prisonniers.

Pour l'affichage et le déplacement des modèles 3D, je me suis

heurté à un premier problème. Il faut manipuler des matrices et le produit matricielle pour juste afficher un modèle. Malheureusement, je n'avais pas encore saisie la subtilité du produit matricielle : il n'est pas commutatif. J'ai perdu énormément de temps sur ce point au début. En ce qui concerne l'animation à proprement parler, j'ai rencontré une nouvelle difficulté. Pour la comprendre, il faut comprendre comment un modèle 3D est enregistré. Un modèle est en réalité un squelette sur lequel on vient poser une image, une texture comme on le ferait avec du papier peint. Et c'est ce squelette qui est enregistré en mémoire. Chaque mouvement ou rotation de chaque os de ce squelette est aussi enregistrée pour constituer plus tard une animation 3D. Pour animer le modèle, il faut dire au programme de charger l'état suivant du squelette. Mon problème a été justement de charger la position suivante. Je ne savais pas quelle propriété du modèle utiliser, ou comment faire pour modifier l'état du squelette.

Très peu de documentation existe sur Internet. Je n'ai réussi à trouver qu'un exemple issue du site officiel de microsoft. Mais cet exemple n'avait aucune explication ou commentaire. Déchiffrer les lignes de code a été assez ardu. Au fait de mon manque d'expérience en programmation 3D, beaucoup de temps a été nécessaire pour reproduire l'exemple. Une erreur revenait souvent : la même image était chargée sans arrêt. Après de nombreux tests infructueux, j'ai finalement réussi à obtenir une animation complète des personnages.

1.2.2 Création d'une intelligence artificielle

A ce stade du développement, notre personnage est capable de se déplacer dans son environnement 3D et d'entrer en collision avec les murs, grâce au travail d'Anatole. Faire se déplacer les ennemis a donc été l'étape suivante dans le développement du jeu. La difficulté est de déterminer un chemin aléatoire dans la carte. Il faut que ce chemin soit libre de murs et d'ennemis, pour que le déplacement s'effectue sans encombre. Une fois ce chemin déterminé, il faut trouver un moyen de le faire parcourir par l'ennemi. Pour choisir un chemin aléatoire, nous définissons en premier lieu une distance maximale. Puis nous plaçons un marqueur à la position de départ de l'ennemi. Ensuite, une direction est choisie au hasard. Le programme suit ensuite cette direction jusqu'à l'intersection suivante, où il choisie une nouvelle direction au hasard. Ce processus est répété tant que la distance maximale n'est pas atteinte. Cela permet d'obtenir un parcours pseudo-aléatoire.

Vient ensuite le problème du déplacement des ennemis. Nous savons où ils doivent aller, mais nous ne savons pas comment. Etant donné que le déplacement des ennemis ne concernait pas vraiment la boucle de jeu principale, une seule idée nous est apparue, à Anatole et moi. Il s'agissait d'utiliser des processus en parallèle, ou des `THREAD`. Un thread est un outil du `C#` qui permet d'exécuter deux actions en parallèles, ou en tout cas en donne l'illusion. Un ordinateur, naturellement, ne peut effectuer qu'une seule chose à la fois. Ainsi les ennemis sont capables de se déplacer de façon aléatoire et autonome, sans ralentir l'exécution principale.

Mais se déplacer selon un chemin prédéfini en permanence n'est pas très "intelligent" pour une intelligence artificielle. C'est pourquoi Ana-

tole et moi avons implémenté une capacité repérage aux ennemis. Lorsque notre personnage se situe à une faible distance d'un ennemi et que celui-ci regarde dans la direction du personnage, alors on considère que le personnage est repéré. Dès lors, l'ennemi va poursuivre le héros, jusqu'à la mort de l'un des deux ou si le héros arrive à échapper à la surveillance de l'ennemi.

1.2.3 Utilisation d'objets

Notre jeu dispose maintenant d'ennemis capables de se déplacer et d'un personnage. Il nous manquait encore la possibilité d'utiliser des armes, telles qu'un pistolet ou un couteau. Etant donné que ce point est encore relatif à la 3D, je m'en suis occupé. La première étape a été de modéliser des armes en 3D, avec Blender. Cela n'a pas été très difficile. Notre jeu ayant adopté un style graphiste assez simpliste, je n'avais pas à créer d'armes très détaillées.

1.3 Impressions personnelles

1.3.1

Table des figures