*420-SCD Intégration des apprentissages en SIM*

***Rapport de projet***

28 avril 2014

Par

Ahmed Ibrahim Mbengue et Tobi Décary-Larocque

Présenté à

Caroline Houle

Collège de Maisonneuve

Date de remise : 1er Mai 2014

1. **Introduction**  
   Le présent document est un rapport sur projet intégrateur du cours d’intégration des apprentissages en sciences, informatiques et mathématiques. Il contient de l’information sur l’évolution du projet depuis la première soumission jusqu’à la remise finale. De plus, il inclut les différentes taches informatiques et scientifiques faites par chacun des membres de l’équipe ainsi qu’une section démontrant les difficultés rencontrées au cours la réalisation du projet et les ressources utilisées à sa réalisation. Une appréciation personnelle des membres de l’équipe est jointe à la fin de ce document.
2. **Une section sur l'évolution de votre projet**

* **Menu Éditeur de vaisseau :**

Pour ce qui est de l’éditeur de vaisseau, nous avions prévu pouvoir modifier des valeurs, telles que la couleur du vaisseau et le choix de plusieurs parties différentes pour le vaisseau. Nous avons dû abandonner certaines de ces idées, car donner trop de possibilités de vaisseau faisait ralentir l’application et nous avons décidé de donner des textures métalliques au vaisseau personnalisé au lieu de changer la couleur.

De plus, nous avons remplacé le concept de boutons radios pour changer les différentes parties par des formes que l’on « drag and drop » dans une zone d’affichage et nous avons ajouté la possibilité de changer manuellement la masse, la force de poussée et la vitesse de rotation du vaisseau, et ce, de manière plus accessible.

* **Menu Création de niveau**

Notre concept de départ pour le créateur de niveau était d’avoir une barre sur le côté où l’utilisateur pouvait cliquer sur des formes déjà conçues dans une barre située sur le côté et les déplacer à sa guise dans un niveau. Puisque ce modèle aurait entrainé plusieurs contraintes, telles qu’empêcher l’utilisateur de changer la forme de l’obstacle et qu’une barre sur le côté aurait pris beaucoup trop de place, nous avons changé le modèle pour un « pointeur » qui donne un aperçu de l’obstacle qui va être mis en place et dont la taille, la couleur et la forme peuvent être modifiés.

Par contre, nous avons dû abandonner la possibilité de changer la gravité du niveau et de choisir la position de départ du vaisseau, puisque changer la masse et la force de poussée du vaisseau affectait déjà la gravité et que changer la position de départ n’était pas très pertinent à notre avis.

* **Zone de jeu**

Dans la zone de jeu, nous avons ajouté quelques fonctionnalités, telles que le « side scrolling », qui permettaient de faire des niveaux plus longs et de donner une meilleure impression à l’utilisateur.

Par contre, nous avons enlevé le concept de vitesse d’atterrissage du vaisseau, car cela rajoutait une difficulté qui n’était pas nécessaire et empêchait l’utilisateur d’atterrir ailleurs que sur la zone d’atterrissage, ce qui pouvait être frustrant pour les débutants.

* **Choix de niveau**

Pour ce qui est des choix de niveaux, nous avons décidé d’ajouter un niveau de difficulté suite aux commentaires de ceux qui avaient testé notre application et qui demandaient un niveau de grande difficulté.

1. **Tableau de la répartition des tâches**

|  |  |  |
| --- | --- | --- |
|  | Tâches informatiques accomplies | Tâches scientifiques accomplies |
| Ahmed Ibrahim Mbengue | * Écouteurs personnalisés qui permettent à l’application principale d’être à l’écoute des différents menus. * Classes qui dessinent les différentes formes utiles à l’animation de l’application (vecteurs, obstacle, etc.). * MouseListener permettant les déplacements dans l’éditeur de vaisseau. * Commentaires Javadocs des classes et méthodes de l’application. * Conception du menu d’aide * Classe « Éditeur de vaisseau » qui donne la possibilité à l’utilisateur de monter son vaisseau et de choisir ses propriétés * Création de différents niveaux de difficulté | * Calcul derrière l’animation pour permettre au vaisseau personnalisé de s’afficher. * Dessin de forme complexe en utilisant des calculs géométriques. * Transformation de donnée en monde réel dans la zone de jeu. |
| Tobi Décary-Larocque | * Écouteurs personnalisés permettant l’ajout de plusieurs boutons qui aident à naviguer entre les différents menus de l’application et l’échange d’information entre les menus. * KeyListeners qui permettent à l’utilisateur d’interagir avec l’application grâce au clavier. * Classe « Zone de jeu », qui dessine le vaisseau et son entourage et lui permet de faire des actions à travers certains niveaux. * Création de différents niveaux de difficulté. * Classe « Création de niveau» qui permet à l’utilisateur de créer son propre niveau en plaçant des formes configurables à travers un niveau vide. * Dessin du vaisseau de base. | * Calcul des contrôles du vaisseau, tels que l’accélération, la vitesse et la gravité. * Calcul de la relation entre la masse et la force de poussée pour les ajouter aux calculs de la vitesse. |

1. **Complexité du projet**

Pour ce qui est de la complexité du projet, les plus grands défis se retrouvaient dans les classes de la zone de jeu, de l’éditeur de vaisseau et du créateur de niveau. Pour la zone de jeu, les plus grands problèmes se retrouvaient dans le calcul de la position du vaisseau, qui devait prendre en compte la vitesse, l’accélération et la gravité qui tirait le vaisseau vers le bas en tout temps. De plus, trouver des valeurs qui rendaient le déplacement fluide, facile à contrôler et réaliste n’a pas été facile. Pour l’éditeur de vaisseau, la plus grande complexité se retrouvait au niveau des écouteurs de souris parce qu’il fallait s’assurer que les bonnes valeurs soient modifiées après chaque évènement, et aussi au niveau des modifications de la masse, de la force de poussée et de la vitesse de rotation, qui devaient être communiquées aux autres classes et qui devaient représenter le plus possible celles d’un vrai vaisseau spatial. Le menu de création de niveau a pris beaucoup de réflexion pour bien fonctionner, et trouver une façon de placer des obstacles à volonté, tout en donnant beaucoup de possibilités à l’utilisateur, et d’ensuite créer un niveau personnalisé en plaçant les obstacles à la bonne place fut difficile.

1. **Description des ressources utilisées**

Pour ce qui est du code utilisé, il est majoritairement créé par nous-mêmes, en s’inspirant seulement de quelques sources externes, comme les contrôles du vaisseau par exemple, pour lesquelles nous avons pris un autre jeu de simulation d’hélicoptère pour nous donner une idée de comment coder l’accélération et comment la relier aux contrôles du clavier. Sinon on s’est servi du site <http://doc.oralce.com> pour s’informer sur les classes et les méthodes utiles à la réalisation de notre projet.

1. **Appréciations et conclusion**

Pour ce qui est du déroulement général du projet, nous sommes plutôt satisfaits de comment cela s’est passé. Au cours de ce projet, nous avons appris les difficultés et les défis de ce domaine de travail et nous avons dû gérer notre temps relatif au projet malgré les autres évènements qui sont arrivés au cours de notre session. Nous avons aussi appris à travailler en équipe et à bien communiqué avec son partenaire de travail. Nous avons également beaucoup amélioré nos connaissances dans le domaine de programmation, car nous avons dû trouver nous-mêmes comment créer cette application du début à la fin et nous avons dû trouver comment coder des choses que nous n’avions jamais faites auparavant. Si nous avions eu 50 heures de plus pour travailler sur le projet, nous aurions pu ajouter une plus grande variété de choix dans le menu d’éditeur de vaisseau et peut être donné le choix à l’utilisateur de sélectionner la texture de son choix. Nous aurions aussi pu ajouter une fonctionnalité qui permettrait de « zoom out » au cours du jeu, ce qui aurait pu ajouter une meilleure vue d’ensemble aux niveaux et aurais donné plus de possibilités quant aux différents niveaux créés, car pour le moment notre zone de jeu n’est pas très grande. Mais au final, ce projet nous donne une bonne idée d’à quoi nous attendre dans le futur et nous donne une idée de ce qu’on est capable d’accomplir avec nos connaissances dans le domaine de la programmation, ainsi que le lien qu’on peut tirer avec les autres matières.

**1093 mots**