**Projet Personnel Secondaire 5**

Par **Frédéric Gosselin**

Concentration Informatique

École Secondaire Samuel-De-Champlain

1. Introduction

2. Fonctionnalités

3. Commandes

4. Logiciels utilisés

5. Liens internet

**1. Introduction**

Depuis que j'ai découvert le logiciel RPG Maker au primaire, je me suis intéréssé à la programmation de jeux vidéo. Déplacer des événements et construire ses propres cartes était en soi une expérience bien agréable, mais je me sentais un peu limité à vivre aux dépends des commandes de base définies par le logiciel et me suis donc intéréssé aux scripts.

RPG Maker utilise un moteur de jeu bien agréable, le RGSS (Ruby Game Scripting System),

couplé au puissant langage de programmation Ruby, qui reste malheureusement peu connu en

dépit de son rival le Python qui est de plus en plus enseigné. En utilsant le RGSS, on peut obtenir d'intéressants projets (Pokemon SDK, FE XP, Victor Engine, etc), mais on reste encore beaucoup restreint par les limites du logiciel. Notons que dans RPG Maker, le jeu descend rapidement en FPS (Frames Per Second) dès qu'on est lancé dans un jeu trop gourmand en Bitmaps. En plus d'être payant, le RGSS n'est pas très portable. Même si le Ruby peut être utilisé sous plusieurs OS, le RGSS ne fonctionne que sous Windows. Ce sont ces points qui m'ont amenés à porter mon jeu sous le Ruby uniquement, parce que, tout d'abord, la version actuelle du Ruby est jusqu'à 5 fois plus rapide que le RGSS1 normal et que, ensuite, c'est le langage qui est le plus fexible pour moi qui est sans cesse en train de modifier mon code. Pour ce qui est de l'affichage des graphiques et des sons, puisque Ruby ne vient pas avec des fonctions de base pour créer un jeu, j'ai installé le moteur de jeu Gosu. Pour ce qui est des mécaniques du jeu, je me suis basé sur *Lego Battles* dont j'ai apprécié le système de combat.



**2. Fonctionnalités**

Nom du projet : Nouka

Type du projet : Jeu RPG « Defender-like »

Le principe du jeu est simple. Le personnage principal possède des « Entités » (class Entity) qu’il pourra faire combattre à celles du joueur adverse. Selon le niveau, il faudra soit éliminer des bâtiments ou des entités spécifiques, soit défendre son territoire pendant *n* temps.

En sélectionnant une entité de votre équipe, un menu se déroule pour vous afficher les actions qu’elle peut exécuter : *{Déplacement, patrouille, attaquer}.*

Pour certaines entités spéciales, par exemple le héros, on aura accès à *{invoquer}* ou *{bâtir}* pour les ouvriers.

En utilisant les flèches directionnelles, on a la possibilité de visiter les parties de la carte qu’on a visitées.

Le système fonctionnera avec des points de mana, d’argent et d’entités maximum.

**3. Commandes**

Clic simple : Sélection entité ou validation fenêtre.

Déplacement souris + clic. Sélection de plusieurs entités.

Flèche directionnelles : Déplacement du champ visible de la carte dans la fenêtre de jeu.

Roulette souris : Zoom Avant/Arrière de la carte du jeu.

Touche Échap : Accéder au menu pause.

**4. Logiciels utilisés**

Voici une liste des logiciels utilisés cette année dans mon projet :

* Ruby.exe

Interpréteur Windows pour le langage Ruby qui permet de lancer des fichiers « \*.rb ».

Pas de soucis pour exécuter le jeu; il suffit de le démarrer avec le fichier « run.cmd ».

* Bibliothèque Gosu

Librairie pour le Ruby utilisant Open Gl qui permet d'afficher des images dans une fenêtre. (Après avoir installé Ruby, il faut utiliser la commande :

« gem install gosu » dans l’interpréteur de commandes Windows.

* Marshaled Instances Reader MIRABANDONNÉ.

Un logiciel fait par moi cette année, écris en ruby, qui utilise Qt, et qui permet d'éditer

les fichiers qui se trouvent dans le répertoire « /Data/GameData/\*.MIF ».

(Et les fichiers RPG Maker, au passage \*-\*.)

* ScriptIndexing.rb

Indexer les fichiers \*.rb se trouvant dans le répertoire « /Data/Scripts/\*.rb »

(L'index est très important! Il indique au jeu quel script sera chargé avant un autre.)

* Tiled Editor

Il s’agit de l’éditeur de carte utilisé dans mon jeu, après l’abandon de MIR.

**5. Liens internet**

**Notions vues cette année, utilisées dans le développement des scripts :**

« Quadtree » :

<https://gamedevelopment.tutsplus.com/tutorials/quick-tip-use-quadtrees-to-detect-likely-collisions-in-2d-space--gamedev-374>

« Pathfinding » :

<http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html>

« PriorityQueue » :

<http://www.brianstorti.com/implementing-a-priority-queue-in-ruby/>

« Light&Sight » :

<http://ncase.me/sight-and-light/>

**Ressources du jeu :**

« DawnLike Pack » :

<https://opengameart.org/content/dawnlike-16x16-universal-rogue-like-tileset-v181>

**Logiciels :**

Langage Ruby

Site officiel : <https://www.ruby-lang.org/fr>

Installateur : <https://rubyinstaller.org/>

Documentation : <http://docs.ruby-doc.com/docs/ProgrammingRuby/>

Moteur de jeu Gosu

Site officiel : <https://www.libgosu.org/index.html>

Documentation : <http://www.rubydoc.info/github/gosu/gosu>