SUPINFO International University

INSTITUTE OF INFORMATION TECHNOLOGY

Projet de groupe B2
Développement - Rampart

Table des matières

1. Fiche signalétique du groupe de projet

1.1. Membres du groupe

ID Booster	Nom	Prénom	Photo
93762	LABARRE	Gautier	
92807	MAITRE	Cyril	
95535	MIRROIR	Alexis	
92692	STROPPA	Clément	
92582	WALTER	Raphael	

2. Rapport de projet

2.1. Répartition des tâches

2.1.1. Travail réalisé par **Gautier LABARRE**

- Analyse et conception générale du projet
- Analyse et conception approfondie des classes :
  - Artillerie
  - Canon
  - ScreenEffect

2.1.2. Travail réalisé par **Cyril MAITRE**

- Analyse et conception générale du projet
- Analyse et conception de la structure du programme
- Analyse et conception approfondie des classes :
  - Partie
  - Mur
  - MurFigure
  - Carte
  - EtatTerrain
  - Objet

2.1.3. Travail réalisé par **Alexis MIRROIR**

- Analyse et conception générale du projet
- Analyse et conception approfondie des classes :
  - Aide
  - Media
- Conception du design
- Réalisation des images / « sprites »

2.1.4. Travail réalisé par **Clément STROPPA**

- Analyse et conception générale du projet
- Analyse et conception approfondie des classes :
  - Option
  - Chateau
  - Boulet
- Portage du programme sous Debian GNU/Linux

2.1.5. Travail réalisé par **Raphael WALTER**

- Analyse et conception générale du projet
- Analyse et conception approfondie des classes :
  - Menu
  - Bateau
- Portage du programme sous Mac OS X

3. Guide du logiciel

3.1. Manuel d'installation

Le projet est fourni avec une version de l'exécutable portable. Cela signifie qu'aucune installation à proprement parlé n'est nécessaire. Cependant certaines contraintes sont à respecter.

3.1.1. Installation sur Windows

Pour lancer le jeu sous Windows, rendez-vous dans le dossier *exe/windows/* et lancez "*Rampart.exe*".

L'exécutable est fourni avec les bibliothèques dynamiques (DLL) *libsndfile-1.dll* et *openal32.dll*.

Vous devrez également posséder le "*Package redistribuable Microsoft Visual C++ 2010*". Dans le cas contraire, une erreur de ce type surviendra lors de l'exécution du jeu : "*Impossible de démarrer le programme car il manque MSVCP100.dll sur votre ordinateur*"

- Package redistribuable Microsoft Visual C++ 2010 (x86) - 4,8Mo: http://www.microsoft.com/downloads/fr-fr/details.aspx?FamilyID=a7b7a05e-6de6-4d3a-a423-37bf0912db84
- Package redistribuable Microsoft Visual C++ 2010 (x64) - 5,5Mo: http://www.microsoft.com/downloads/fr-fr/details.aspx?FamilyID=BD512D9E-43C8-4655-81BF-9350143D5867

3.1.2. Installation sur Mac OS X

Pour lancer le jeu sous mac, décompressez le fichier "Rampart.zip" qui se trouve dans *exe/mac*, puis exécutez le fichier.

3.1.3. Installation sur Linux

Pour lancer le jeu, vous devrez installer SFML, qui est la bibliothèque graphique utilisée.

La version GNU/Linux du jeu a été développée et testée sur le système d'exploitation Debian Squeeze.

Cette notice est donc spécifique à ce système. Elle s'applique également, en théorie, aux distributions "Debian-like" (Ubuntu, ...).

Pour les autres distributions, renseignez vous sur l'installation de **SFML**.

- *Installation de SFML en ligne de commande*
  ```
  aptitude install libsfml-dev libsfml-doc libcsfml-dev
  ```

- *Lancement en interface graphique*

Les fichiers se trouvent dans le dossier */exe/linux/*.

Double-cliquez sur l'icône de l'exécutable du jeu: rampart-debian.x64 ou rampart-debian.x32 selon votre architecture (64 ou 32 bits).

- *Lancement en ligne de commande*, utilisez la commande suivante :
    - Depuis n'importe quel répertoire:
      `/chemin/du/jeu/rampart-debian.x64`
    - Si vous vous trouvez dans le répertoire du jeu
      `./rampart-debian.x64`
    - **NOTE:** En cas de problème, vérifiez que le fichier soit exécutable :
      `chmod +x rampart-debian.x64`

3.2. Manuel d'utilisation

3.2.1. Ecran d'accueil

Après avoir lancé le jeu, vous arriverez sur le menu principal. Ce menu vous permet de :

- Lancer le jeu en cliquant sur "*Jouer*"
- Modifier les options du jeu en cliquant sur "*Options*"
- Consulter l'aide du jeu en cliquant sur le point d'interrogation sur fond bleu en bas à droite de l'écran.
- Quitter le jeu en cliquant sur "*Quitter*"

3.2.2. Modifier les options

Vous avez la possibilité de modifier les options du jeu afin d'améliorer votre expérience de jeu. En effet ces options affecteront directement le déroulement du jeu.

Voici la liste des paramètres :

- **Musique** : Permet d'activer ou non la musique
- **Niveau** : Permet de définir le niveau du jeu. "*1*" étant le niveau le plus facile et "*3*" le plus difficile. Le niveau influe sur la vitesse de déplacement et le nombre de point de vie des bateaux.
- **Contrôles** : Cette option vous permet de changer les touches de contrôle du jeu.

3.2.3. Le Jeu - Phase Initiale : Sélection de la ville principale

La phase initiale du jeu, consiste à sélectionner sa ville principale dans un délai de 10 secondes. Cette ville sera dès le début du jeu protégée par des remparts.

Pour sélectionner votre ville principale, utilisez les touches "*Droite*", "*Gauche*" et "*Action*" ou votre souris en cliquant 2 fois sur la ville que vous avez choisie.

3.2.4. Le Jeu - Phase I : Placement des murs

Sur le terrain, vous avez 15 secondes pour placer des murs de formes aléatoires pour constituer des remparts fermés autours de vos villes afin de les protéger.

Pour positionner vos murs, utilisez votre souris ou les touches "*Haut*", "*Bas*", "*Droite*" et "*Gauche*". Utilisez le clic gauche de la souris ou la touche "*Action*" pour placer vos murs.

Un mur ne peut être placé que sur une case de terre vide.

3.2.5. Le Jeu - Phase II : Placer les cannons

Une fois les murs placés, vous avez 15 secondes pour placer un nombre défini de canons de défense, à l'intérieur des murs.

Pour positionner votre canon, utilisez votre souris ou les touches "*Haut*", "*Bas*", "*Droite*" et "*Gauche*". Utilisez le clique gauche de la souris ou la touche "*Action*" pour placer votre canon.

Un canon ne peut être placé que dans une zone protégée par des remparts (case avec un fond grisé) vide.

3.2.6. Le Jeu - Phase III : Assauts des bateaux

Ensuite, viens la phase où les bateaux attaquent les murs. Vous devez alors riposter avec vos armes pour détruire un maximum de bateaux.

Utilisez votre souris ou les touches "*Haut*", "*Bas*", "*Droite*" et "*Gauche*" pour déplacer votre viseur. Utilisez le clic gauche de la souris ou la touche "*Action*" pour tirer avec le premier canon disponible. Réitérez l'action pour tirer avec le canon suivant. Mais attention, vos canons ont un temps de rechargement.

Si un boulet tiré par un de vos canons, touche un bateau, celui-ci perd un point de vie. Si les points de vie d'un bateau atteignent 0, celui-ci est détruit. Le nombre de points de vie d'un bateau est déterminé par la difficulté que vous avez choisi (*Difficulté 0* : 1 point de vie, *Difficulté 1* : 2 points de vie, et *Difficulté 2* : 3 points de vie).

Les bateaux vont également tirer sur vos murs. Lorsqu'un boulet tiré par un bateau atterri sur un de vos murs, celui-ci est détruit et laisse une brèche dans vos remparts. La vitesse de déplacement

des bateaux est déterminée par le niveau de difficulté que vous avez choisi (*Difficulté 0* : vitesse faible, *Difficulté 1* : vitesse moyenne, et *Difficulté 2* : vitesse élevée).

Cette phase se termine à la fin du temps imparti ou lorsque tous les bateaux ont été détruits. Les bateaux non détruits à la fin de la phase réapparaîtront à la phase suivante.

3.2.7. Le Jeu - Phase IV : Reconstruction

Enfin, il faut reconstruire les murs détruits durant la phase précédente grâce aux formes aléatoires.

Les tours se succèderont ainsi de suite jusqu'à ce que les villes soient toutes protégées. Dans ce cas le joueur gagne la partie. Si aucune ville n'est protégée, alors il perd la partie.

A la fin de cette phase, le calcule du score est réalisé :

- Pour chaque mur placé : *+10 points*
- Pour chaque mur détruit: *-10 points*
- Pour chaque bateau détruit : *+100 points*
- Pour chaque case protégée : *+25 points*

3.2.8. Le Jeu - La pause

Durant le jeu, vous pouvez à tout moment activer la pause en appuyant sur la touche "*Pause*". A partir du menu pause, vous pouvez soit retourner au jeu en appuyant de nouveau sur la touche "*Pause*", ou bien quitter la partie en cours en appuyant sur la touche "*m*" (menu).

3.3. Manuel de compilation

3.3.1. Cas général

Nous utilisons la bibliothèque graphique SFML version 1.6, et en ce sens, il vous faudra l'installer (« include » et .lib) sur votre système d'exploitation pour pouvoir compiler le projet.

Vous pouvez télécharger la SFML (version 1.6) ici : http://www.sfml-dev.org/download-fr.php

En outre, vous aurez besoin, sous Windows, des DLL suivantes pour l'exécution du jeu :

- les DLL de la SFML
- libsndfile-1.dll
- openal32.dll

3.3.2. Compilation avec Visual Studio 2010

Nous avons recompilé la SFML pour l'utiliser avec Visual Studio 2010 (la version 1.6 à la date de début du projet n'était pas compatible avec Visual Studio 2010).

Les fichiers de la SFML recompilée ayant une taille non négligeable, nous ne les avons pas inclus dans le rendu de projet. Ils sont disponibles dans une archive, accompagnés d'un guide d'installation, à l'adresse suivante: http://b2-rampart.stroppa.info/SFML_1-6_recompiler_VS_2010.zip

Après l'installation de la SFML, vous trouverez dans le dossier *src/windows/Projet Visual Studio 2010 le* projet Visual Studio 2010. Il suffit d'ouvrir le fichier "*Rampart.sln*" avec Visual Studio 2010, puis de lancer la compilation (le projet est déjà paramétré pour inclure la SFML).

3.3.3. Compilation avec NetBeans (**version GNU/Linux uniquement**)

La version « Linux » du jeu a été développée à l'aide de l'IDE **NetBeans 7.0.** Ainsi pour compiler le jeu, il vous suffit de vous rendre dans le dossier */src/linux* avec NetBeans et d'ouvrir, tout simplement, le projet.

La bibliothèque graphique SFML est nécessaire à la compilation.

➔ Référez vous à la partie [*1.1.1.Installation sur Linux]* pour l'installation de SFML sous Debian.

3.3.4. Compilation avec Xcode 4

Pour compiler le jeu sous Xcode, vous devez commencer par installer la SFML sur votre machine. Pour ce faire, vous devez télécharger la bibliothèque sur le site officiel de la SFML en choisissant la version correspondant à l'architecture de votre système d'exploitation (32 ou 64 bits).

Une fois la bibliothèque téléchargée, copiez le contenu du dossier « *SFML-x.y/lib* » (ou lib64) dans le répertoire « */Bibliothèque/Frameworks* ». Vous devez ensuite aller dans le répertoire *« SFML-x.y/extlibs/bin* » puis copier, dans le cas d'une version 32 bit, les dossiers *« OpenAL.framework* » et « sndfile.framework » dans le répertoire « */Bibliothèque/Frameworks* » ou le dossier x86_64 pour la version 64 bit.

Suite à cela, il vous suffira d'aller dans le dossier *src/mac* et d'exécuter le fichier « Rampart.xcodeproj »

3.3.5. Compilation avec un autre IDE

Pour compiler le projet avec un autre IDE, installez tous d'abord la SFML sur votre IDE. Ensuite, créez un nouveau projet et paramétrez-le pour inclure la SFML.

Enfin ajoutez les fichiers source à votre projet et lancez la compilation.

Tutoriel pour installer la SFML: http://www.siteduzero.com/tutoriel-3-321241-installation-de-la-sfml.html

3.4. Manuel développeur

Le jeu a été développé de manière à être facilement paramétrable. Il vous suffira de changer la valeur de certaines directives « *#define* » pour modifier le comportement du jeu :

3.4.1. Les compteurs

- Changer le compteur de la phase initiale: `_INIT_COMPTEUR_P1` dans "*Partie.h*"
- Changer le compteur de la phase I: `_INIT_COMPTEUR_P2` dans "*Partie.h*"
- Changer le compteur de la phase II: `_INIT_COMPTEUR_P3` dans "*Partie.h*"
- Changer le compteur de la phase III: `_INIT_COMPTEUR_P4` dans "*Partie.h*"

3.4.2. Les points gagnés et perdus

- Changer le nombre de points gagnés en ajoutant un mur : `_POINT_ADD_MUR` dans "*Partie.h*"
- Changer le nombre de points perdus en perdant un mur : `_POINT_DEL_MUR` dans "*Partie.h*"
- Changer le nombre de points gagnés en détruisant un bateau : `_POINT_DEL_BATEAU` dans "*Partie.h*"
- Changer le nombre de points gagnés en protégeant un terrain : `_POINT_ADD_PROTECTED` dans "*Partie.h*"

3.4.3. La carte

- Modifier la taille de la carte : `_NB_CARRE_X` et `_NB_CARRE_Y` dans "*define.h*"
- Modifier le contenue de la carte : *carte_1* dans "*define.h*"
  Contenu disponible:
  - Terre 1: `_TERRE`
  - Terre 2: `_TERRE_2`
  - Herbe 1: `_TERRE_HERBE`
  - Herbe 2: `_TERRE_HERBE_2`
  - Mer: `_EAU`
- Modifier les figures de mur générées : `_NB_FIGURE_MUR` dans "*MurFigure.h*" et ajouter figureMur_x dans "*MurFigure.h*"

3.4.4. L'artillerie

- Temps de rechargement des canons : `_RECHARGEMENT_DELAY` dans "*Artillerie.h*"
- Temps de rechargement des bateaux : `_RECHARGEMENT_DELAY_BATEAU` dans "*Bateau.h*"
- Vitesse de base des bateaux (sans tenir compte du niveau de difficulté) : `_VITESSE_BASE` dans "*Bateau.h*"

3.4.5. Débogage

- Activer / désactiver le débogage de l'application : les directives « *#define* » de "*debug.h*"

4. Documentation technique

4.1. Technologies employées

4.1.1. Le langage

Le langage utilisé pour ce projet est le langage C++. C'est un langage orienté objet, créé en 1983 et mondialement utilisé de nos jours. Depuis près de 30 ans il a largement fait ses preuves et est utilisé aussi bien pour des logiciels administratifs que pour des jeux (le très célèbre Half-Life 2 par exemple).

Ce langage nous a donc paru être un bon choix, d'autant que nous avons eu un cursus à son sujet en début d'année.

Pour la gestion multimédia (images, sons, événement... etc.), nous avons utilisé une bibliothèque graphique. Notre choix s'est porté sur la bibliothèque SFML ("Simple and Fast Multimedia Library"). Cette bibliothèque possède plusieurs avantages :

- Elle est gratuite.
- Elle est portable (Windows, Linux, Mac OS).
- Elle possède une architecture orientée objet.
- Elle est facile à utiliser, possède une documentation très complète, ainsi que plusieurs tutoriels en Français.
- Elle propose plusieurs modules permettant de gérer l'aspect graphique, événementiel, audio, etc.

4.1.2. Les Environnements de développement intégrés (IDE)

Nous avons utilisé un IDE par système d'exploitation. Nos choix se sont portés sur des IDE connus et très utilisés dans le milieu professionnel.

- Windows : Visual Studio 2010, avec le plugin Visual Assist X qui permet une meilleur coloration syntaxique.
- Linux : NetBeans IDE 7.0
- Max OS X: Xcode

4.1.3. Système de gestion de versions

Pour améliorer et faciliter le travail en groupe, nous avons utilisé un système de gestion de versions, SVN (Subversion).

4.1.4. Editeur de dessin

- Photoshop : Editeur de dessin avancé, pour la réalisation des « sprites » du jeu
- Paint : Editeur de dessin simple pour la retouche des « sprites » du jeu