

Rapport de programmation de source libre

Membres de l'équipe:

- Jean-William Perreault
- Julien Métivier

1. Application choisie:

Supertux (<http://supertux.lethargik.org/>)

2. Langage(s) de programmation:

C++

3. Préparation requise à l'exécution à partir des sources (compilation vanille du code source):

3.1- Bibliothèques supplémentaires et/ou spécifiques requises

- La bibliothèque SDL et ses extensions
 - SDL_Image
 - SDL_Mixer
 - Mesa (optionnel)

3.2- Étapes préparatoires (configuration, make, installation, etc)

Procéder à l'installation des bibliothèques supplémentaires précédemment énumérées, lancer le fichier de configuration bash avec la commande «./configure». Par la suite, lancer la commande «make». Et finalement si l'on désire installer l'application sur le système lancer la commande «make install».

4. Rapport de compilation/exécution (vanille)

4.1- Réussite complète?

Nous avons réussi sans trop de problèmes. Lors de la première compilation, il fallut modifier une des sources du pojets, parce que ça syntaxe n'était pas acceptée par le compilateur g++ que nous utilisions.

4.1.1- Identique à l'application ciblée

Oui, puisque la compilation s'est effectuée avec succès.

4.1.2- Reproduite pour les deux membres d'équipe sur des machines différentes.

Oui sans aucuns problème encore une fois, la compilation s'est effectuée avec succès sur plusieurs machines autant à l'école que sur nos machines personnelles.

4.2- Réussite partielle?

4.2.1- Divergence avec l'application visée

4.2.2- Résultat différent entre les membres d'équipe

Non.

4.2.3- Correction pour synchroniser les deux membres d'équipe

Aucune, nécessaire.

5-. Changement(s) à l'application (reproduire cette section pour tous les changements fonctionnels envisagés)

5.1- Description du changement voulu

Étant donné que le jeu constituait un «clone» de «Super Mario Bros», initialement le joueur, à un certain moment du jeu, se voit attribué le pouvoir du feu. Nous avons dès lors décidé d'intégrer au jeu 3 nouveaux pouvoirs, l'eau, le vent et la terre soit les quatre éléments. Nous comptons aussi intégrer un menu permettant le changement de la résolution du jeu.

5.2- Identification du code source modifié

5.2.1- Nom du fichier :

Player.h et Player.cpp

5.2.2- Liste des ajouts (nouvelle classe et/ou fonction et/ou attributs)

Ajout de nouveaux attributs de type pointeur sur Sprites.

Ajout d'attribut statique et d'un accesseur statique

5.2.2.1- Description de la (micro) modification

Ajout de 4 nouveaux attributs de type pointeur sur Sprite soit

- Sprite firetux
- Sprite airtux
- Sprite watertux
- Sprite earthtux

Ajout d'attribut statique et d'un accesseur statique permettant aux autres classes d'accéder au pouvoir courant de la classe joueur:

- static std::string playerPower;
- static std::string getPower();

5.2.3- Liste des modifications (classe et/ ou fonctions changées)

Modification de la fonction Player::draw()

5.2.3.1- Description de la (micro) modification

Ajout d'un système attribuant un pouvoir aléatoirement et attribuant le Sprite correspondant au personnage.

5.2.1- Nom du fichier :

ressources.cpp

5.2.3- Liste des modifications (classe et/ ou fonctions changées)

Modification de la méthode void ressources::loadshared()

5.2.3.1- Description de la (micro) modification

Gestion du chargement des ressources pour nos ressources personnelles soit les nouvelles formes du personnage principal avec ses différents pouvoirs.

5.2.1- Nom du fichier :

setup.cpp

5.2.3- Liste des modifications (classe et/ ou fonctions changées)

Ajout d'attribut de type Menu pour constituer un menu de configuration de résolution.

5.2.3.1- Description de la (micro) modification

Ajout d'un nouveau menu de résolution permettant la configuration de la résolution du jeu. Chaque résolution supportée par la carte graphique est détectée et ajoutée dans un conteneur lu par le menu et l'ajoutant à ses différents choix.

Bien que les résolutions soient présentes dans le menu lorsqu'activée, elle ne prenne pas effet, car nous manquons de temps pour implémenter cette possibilité pleinement.

5.2.1- Nom du fichier :
title.cpp

5.2.3- Liste des modifications (classe et/ ou fonctions changées)

Modification de la fonction void title()

5.2.3.1- Description de la (micro) modification

Modification du texte de crédit pour y ajouter nos noms comme contributeurs au projet supertux.

5.2.1- Nom du fichier :
worldmap.h et worldmap.cpp

5.2.2- Liste des ajouts (nouvelle classe et/ou fonction et/ou attributs)

Ajout de nouveaux attributs de type pointeur sur Sprites.

5.2.2.1- Description de la (micro) modification

Ajout de 4 nouveaux attributs de type pointeur sur Sprite soit

- Sprite firetux
- Sprite airtux
- Sprite watertux
- Sprite earthtux

afin que le personnage lorsqu'il est sur la «world map» prenne l'apparence correspondant à son pouvoir actuel.

5.2.1- Nom du fichier :
special.cpp

5.2.2- Liste des ajouts (nouvelle classe et/ou fonction et/ou attributs)

None

5.2.2.1- Description de la (micro) modification

None

5.2.3- Liste des modifications (classe et/ ou fonctions changées)

Modifications de la fonction Bullet::draw()

5.2.3.1- Description de la (micro) modification

Comme les attaques sont liés avec l'élément du Sprite du personnage, nous avons dû utiliser une methode static pour venir «get» la valeur du personnage, soit son element. Donc, après ce «get», les if() se chargeront de decider l'attaque conforme à l'usage.

5.2.1- Nom du fichier :

supertux.strf

5.2.2- Liste des ajouts (nouvelle classe et/ou fonction et/ou attributs)

Ajout des link pour les .png de Sprite d'attaque

5.2.2.1- Description de la (micro) modification

Comme nous avons maintenant plus d'attaques et plus de .png, nous avons du linker dans le fichier .strf toutes les différentes images pour les associer à un nom qu'on utilise par la suite dans nos «.cpp».

5.2.3- Liste des modifications (classe et/ ou fonctions changées)

None

5.2.3.1- Description de la (micro) modification

None

5.3. Compilation de la nouvelle version d'application

5.3.1- Réussie ?

Oui!!!

5.3.1.1- Résultat(s) désiré(s) obtenu(s)?

Pour la majeure partie, oui.

Seulement certains changements, comme la prise d'effet menu résolution.

Worldmap le personnage ne prends pas toutes les bonnes couleurs.

5.3.1.2- Problème(s) rencontré(s), s'il y a lieu

Le seul problème fut l'intégration d'une variable statique, parce qu'il fallait l'allouer en dehors de la classe concernée seulement afin qu'elle puisse devenir utilisable. (Caprice de C++).

5.3.2- Reproduite pour les deux membres d'équipe sur des machines différentes

Oui.

