M1103 : Projet de Programmation

Monster

Collin Simon -- Lapicardise Romain
Groupe C''

Année universitaire

Enseignant Responsable : Mme. Dutour

Table des matières

1. Introduction

1.1. Contexte

Cette sous-section devra présenter le contexte dans lequel s'inscrit le projet (matière, entité d'enseignement, délais, etc).
1 page maximum
Ce projet de Monster nous est demandé en Algo-Prog (M1103)

En binômes
Apprendre à manipuler la bibliothèque SDL.

Deadline : jeudi 15 décembre à 23H59
A rendre : une archive (aux deux noms du binôme) contenant tout votre travail (répertoire projet Qt Creator complet + mini-rapport format pdf)

1.2. Objectifs

Cette sous-section devra présenter les objectifs du projet (mini-cahier des charges fonctionnel).
1 à 2 pages maximum

Délivrables rendus à l'heure
Respect des critères de présentation des fonctions (zone d'information décrivant les entrées les sorties, les auteurs, ...)
Qualité des commentaires dans votre code source
Evaluation de la démo (le jeu contient-il les fonctionnalités minimales demandées ?)
Impression visuelle laissée par le jeu (les étudiants ont-ils fait un e ort pour rendre le jeu "regardable" ?)
Qualité des algorithmes (présence de bug, algorithmes non optimaux,...)
Qualité des structures réalisées
Qualité du découpage fonctionnel
Présence de fuites mémoire (ou autres lenteurs dues au code des étudiants)
Le jeu contient-il quelques fonctionnalites supplementaires ?
Il faut réveiller tous les monstres sur le plateau
Un monstre peut se déplacer uniquement horizontalement ou verticalement
Si un monstre sort du plateau, le niveau est perdu
Il existe deux sortes de monstres : bleu qu'on ne peut pas bouger, et rouge que l'on peut bouger
Si un monstre se déplace à côté d'un bloc de glace, ce dernier

explose
Si un monstre se déplace sur une eche, sa direction est
modifiée automatiquement
Tout autre obstacle arrête un monstre
Un écran de menu avec deux boutons (un pour jouer, un pour
quitter le jeu)
Un écran de jeu compose de l'image de fond et les monstres
places sur la grille (bleus et rouges)
Au moins un niveau doit être jouable (en respectant les règles
du jeu énoncées sur le site)
Mettre au moins les obstacles de type livre et glace
Gérer toutes les animations graphiques : monstre qui glisse,
glace qui explose (disparaît)
Proposer plusieurs niveaux de jeu
Intégrer les flèches

1.3. Outils utilisés

Cette sous-section devra présenter les différents outils que vous avez utilisés dans votre projet (langage, bibliothèques, IDE, gestionnaire de version, etc...).
1 page maximum

Langage : c++.
Bibliothèques : SDL, math.h, cstdlib, string, sstream.
IDE : *Qt* Creator
Pour communiquer et travailler à deux, nous avons principalement utilisé Skype et quasi jamais Git.

2. Implémentation

2.1. Architecture générale

Cette sous-section devra présenter l'architecture générale de votre projet. En particulier, elle contiendra la description des différents fichiers et leur rôle dans le projet.
1 page maximum

Nous avons essayé de “découper” notre programme de façon logique et de nommer nos fichiers de façon équivoque.

Nous avons découpé notre programme en plusieurs fichiers sources :

- le main.cpp est le fichier “principal”.
- affichage.cc contient les fonctions et actions permettant l’affichage a l’écran de tous les éléments du jeu mais aussi la disparition (destruction) de la glace.
- fichiers.cc contient les fonctions et actions permettant l’ouverture et l’édition du fichier score.txt et ainsi manipulent les deux tableaux score et rang.
- level.cc contient l’action permettant l’initalisation de chaque niveau avec les positions de chaque élément. Contient aussi les fonctions convertissant les pixels en numéro de case et inversement (utilisé dans l’initialisation des niveaux)
- menu.cc contient les actions d’apparition et de disparition en fondu (du menu), l’affichage de l’écran de vistoire et de fin de jeu et bien sûr le menu en lui même avec la gestion des boutons et leurs fonctionnalités (dans l’action menu).
- mouvement.cc contient seulement la fonction qui gère le mouvement des monstres dont les collisions avec les obstacles.
- musique.cc permet, grace à la fonction musique de jouer une musique de fond.
- SDL.cc contient toutes les fonctions propres à la SDL (affichage d’images, de texte...)
- sprite.cc mais aussi l’initialisation de tous les sprites (tout étant dans le sprite.bmp)
- tableau.cc contient les fonctions et actions permettant la détection de la direction donnée au monstre ; la détection du premier obstacle lors d’un déplacement de monstre; la suppression d’un obstacle; le réveil d’un monstre; détection de la victoire ; la fusion des tableaux de monstres et d’obstacles et le test de score (pour le faire apparaitre ou non dans le classement).

Nous avons bien entendu créer des fichiers d'en-tête associés à ces fichiers sources (sauf le main.cpp).

2.2. Structure de données

Cette sous-section devra présenter les différentes structures mises en place dans votre projet.
1 à 2 pages maximum

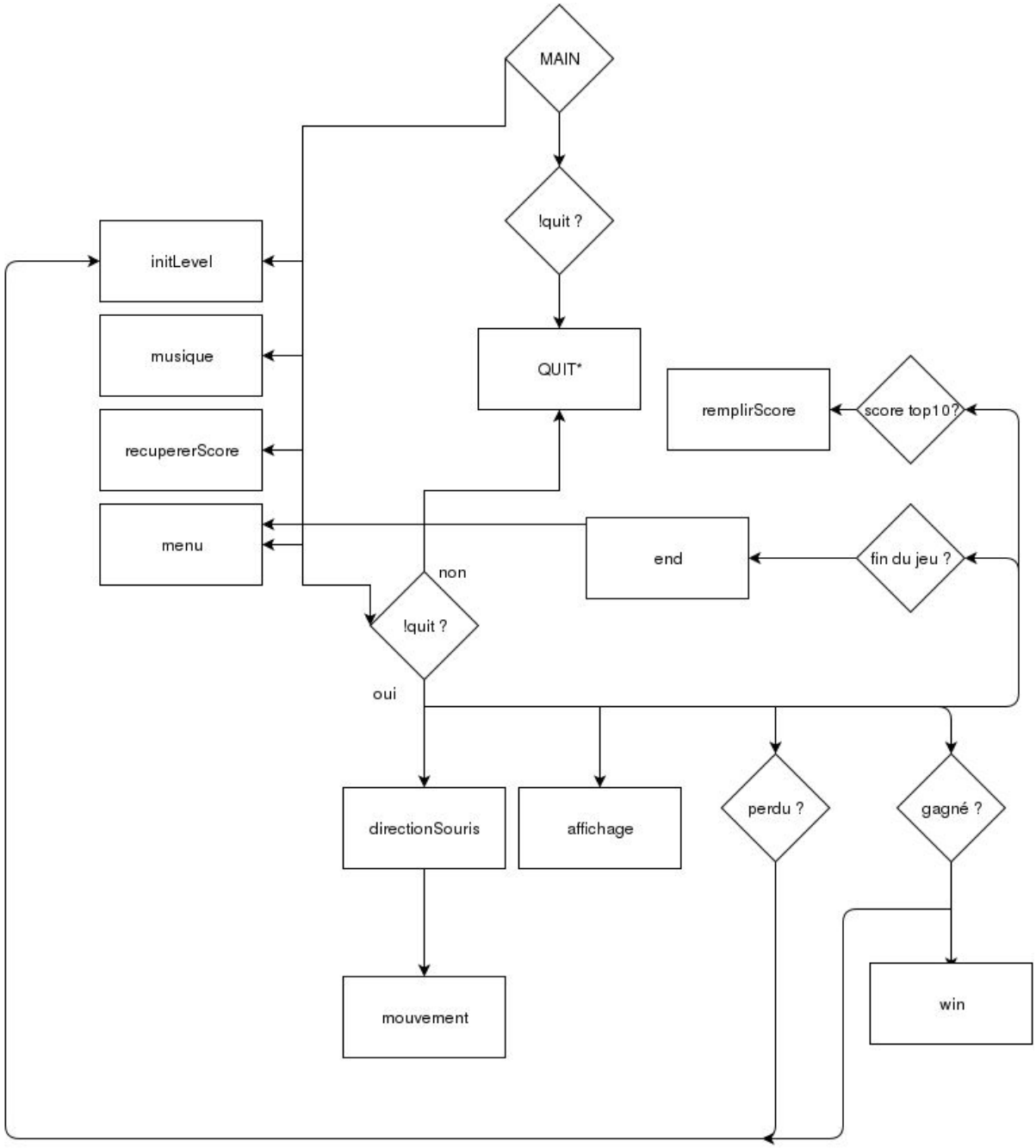
Nous avons créé deux structures dans notre code :

- la structure Level contenant pour chaque niveau un tableau de monstres et un tableau d'obstacles.
- la structure sprite contenant pour chaque "objet" du plateau de jeu un type (monstre, glace, monstre endormi, flèche du bas, du haut, de gauche, de droite et "autre"), des positions (x et y), une hauteur et une largeur, une image et sa lecture.

2.3. Organigramme d'appel des fonctions

Cette sous-section devra présenter sous forme d'organigramme simplifié l'enchaînement de l'appel des principales fonctions du projet.
1 page maximum

https://drive.google.com/file/d/0B_bsLl03tI7GNGI2dmRCUWlhSUE/view?usp=sharing



3. Aspects Algorithmiques

Cette section devra présenter quelques points algorithmiques intéressants de votre projet.
(on n'attend pas ici une description de l'algorithme de gestion du menu, mais plutôt des algorithmes de fonctionnalités du jeu).

3.1. Algorithme 1

Cette sous-section devra présenter le problème que votre algorithme permet de résoudre et l'algorithme que vous avez proposé ainsi que sa description.
1 page maximum

3.2. Algorithme 2

Cette sous-section devra présenter le problème que votre algorithme permet de résoudre et l'algorithme que vous avez proposé ainsi que sa description.
1 page maximum

4. Difficultés rencontrées

Cette section devra présenter les difficultés rencontrées pendant le développement de votre projet, qu'elles soient d'ordre technique, organisationnel ou humain.
1 page maximum

Nous avons rencontré quelques problèmes lors de la conception de notre Monster.
Principalement :

- l'affichage de caractères. Déjà, la police que l'on a choisi n'affiche pas les minuscules et j'ai mis du temps à m'en rendre compte.
- les pseudos associés au score. Cette idée à été abandonnée au profit du rang des scores.
- l'affichage des scores dans la bulle (menu) n'était pas correcte et provoquait un scintillement des scores et/ou de la bulle (suivant ou était le flip). Tout d'abord je pensais que cela venait de la boucle for (lecture des scores et rangs dans leurs tableaux) mais cela venait du fait qu'une succession de else if était plus adapté qu'une succession de if pour la gestion des boutons du menu.
- comme beaucoup le déplacement des monstres s'est avéré assez compliqué pour de nombreuses raisons. Par exemple
- la direction donnée par la souris avec un "glissé-déposé" nous a aussi demandée beaucoup de réflexion. Nous avons d'abord codé le jeu avec des directions donnés par les touches du clavier. Puis, nous avons directement compris qu'il fallait les coordonnées de la souris au clic et au relâché mais nous faisions avec un while; or event.type == SDL_MOUSEBUTTONDOWN ne dure pas dans le temps. Nous sommes donc passé par un booléen et event.type == SDL_MOUSEBUTTONUP.

5. Conclusion

Cette section devra faire écho à l'introduction : avez-vous fait ce que l'on vous a demandé de faire ? Avez-vous respecté les délais ? etc
1 à 2 pages maximum

Nous avons fait tout ce qui nous a était demandé et quelques éléments en plus mais nous n'avons pas réussi et surtout pas eu le temps de faire certains "bonus" comme l'éditeur de niveau ou la possibilité de changer le thème en jeu.
Tout le travail a été très bien organisé et réparti dans le temps et nous a permis de ne pas tout faire au dernier moment.
Nous sommes donc fier de vous présenter notre Monster!