

Compte rendue Projet Programmation

Groupe H: Paul Beziau
Clément Nerestan
Chloé Pathé

20 avril 2015

Table des matières

1	Introduction	5
2	Projet	7
2.1	Le Problème	7
2.2	L'implémentation du jeu	7
2.2.1	Le Jeu	7
2.2.2	Jouer par le biais de la console	8
2.2.3	Jouer avec l'interface graphique	8
3	Critique	9
4	Bibliographie	11

Chapitre 1

Introduction

Ce projet a pour but de développer le jeu 2048. Pour cela, on utilisera les bibliothèques NCurses et Simple directmedia layer (SDL), ainsi que les bibliothèques natives de C. Le 2048 se joue sur une grille de 4x4 tuiles. Ces dernières ont chacune pour valeur une puissance de 2. Les tuiles sont déplacées selon les quatre directions grâce aux touches fléchées : gauche, haut, bas, droite. Lorsque deux tuiles de même valeur 2^n sont assemblées lors d'un mouvement, une tuile de valeur $2^{(n+1)}$ est créée. A la fin d'un mouvement, une tuile de valeur 2 ou 4 est placée de façon aléatoire sur une case vide (une chance sur dix d'obtenir une tuile de valeur 4). Le but du jeu est d'obtenir une tuile de la plus grande valeur la plus grande possible.

Chapitre 2

Projet

2.1 Le Problème

Notre projet est effectué en deux parties : la première consiste en l'implémentation du jeu 2048, la seconde en la création de stratégies permettant de gagner le plus souvent possible. Le jeu doit être jouable avec les touches fléchées du clavier. Il y a donc plusieurs problèmes à régler. Il faut recréer le jeu : la grille, les tuiles, la gestion de leur déplacements et de leurs fusions, les conditions de déplacement et de fin de partie ; il faut aussi implémenter la gestion du clavier. Une fois tout cela terminé, il faudra implémenter des stratégies viables.

2.2 L'implémentation du jeu

Notre jeu est jouable au clavier selon deux modes. Le premier se situe directement dans la console, le deuxième dispose d'une interface graphique.

2.2.1 Le Jeu

La taille de la grille est une constante définie en dehors des fonctions utilisées. La modification de cette variable n'affectera donc pas le fonctionnement du jeu. Cependant, sa valeur a été fixée à 4, afin de respecter le jeu original. La valeur des tuiles est gérée selon leur puissance de 2. Ainsi, un 2 vaudra 1, un 4 sera 2, ainsi de suite. On leur rendra leur vraie valeur à l'affichage. Dans la première version de notre code, cette gestion des tuiles posait problème en cas de 0. En effet, 2 puissance 0 fait 1, et non pas zéro. Nous avons donc décidé qu'en cas de 0, la tuile affichée serait une case vide. Le déplacement des tuiles selon une direction se fait en plusieurs étapes. On vérifie d'abord que le mouvement est possible. Un mouvement est possible si : 1) Au moins une case sur le bord vers lequel on déplace les tuiles est libre. 2) Si on fusionne des tuiles dans la direction demandée ; et donc qu'on libère un ou plusieurs case. Si aucune de ces deux conditions ne sont remplies, le mouvement est impossible et ne sera pas réalisé. Après cette vérification, on tente de décaler les tuiles dans la direction voulue. La fusion entre deux tuiles de même valeur se fait après ce décalage. Au cours de la fusion, le score est implémenté de 2 puissance la tuile créée. Après cette fusion, un nouveau décalage est effectué. Cependant, une seule fusion est

effectuée. Ainsi quatre tuiles de même valeur N ne formeront que deux cases de valeur $N+1$, et non une seule case de valeur $N+2$. Pour obtenir cette case, un nouveau mouvement sera requis. A la fin de chaque mouvement, une tuile de valeur 2 (1) ou 4 (2) est placée aléatoirement dans une des cases vides de la grille. Il y a une chance sur 10 qu'une tuile de valeur 4 soit créée. Quand aucun mouvement n'est possible, la partie est finie.

Afin d'éviter les duplications de code, nous avons décidé de n'effectuer la vérification des mouvements, les fusions et le décalage des tuiles dans un seul sens : vers le haut. Pour cela, nous faisons tourner la grille dans le sens antihoraire. La grille tournera d'un cran pour un mouvement vers la gauche, de deux pour un mouvement vers le bas, et de trois crans pour un mouvement vers la droite. Une fois les différentes étapes du mouvement effectué, la grille tourne de nouveau afin à se replacer dans sa position originelle.

2.2.2 Jouer par le biais de la console

L'affichage du jeu par la console se fait via NCurses. A chaque mouvement, la grille est effacée et recréée. La véritable valeur des tuiles est affichée à ce moment-là. Si un «game over» est obtenu, c'est-à-dire, si le jeu est terminé car plus aucun mouvement n'est possible, le programme se coupe, et l'affichage de la grille et du score obtenu perdure pendant quelques secondes.

2.2.3 Jouer avec l'interface graphique

L'interface graphique est gérée par la librairie SDL. Afin de rendre le jeu lisible, plus la valeur d'une tuile est composée de nombres, plus la taille de la police sera faible. Ainsi, 2048 est écrit en plus petit que 64, et 64 est écrit plus petit que 2, par exemple. De plus, les tuiles ont des couleurs différentes selon leur valeur. Nous pouvons aussi jouer à la souris. L'interface fait 400x450 pixels, chaque tuile fait 100x100 pixels. Nous avons aussi ajouter de la musique qui se lance et qui s'arrête en même temps que le jeu. La seule véritable difficulté de cette partie a été de comprendre comment utiliser la librairie SDL.

Chapitre 3

Critique

Le principal problème que nous avons rencontré est un problème d'organisation. En effet, nous avons la plupart du temps travaillé en même temps sur les mêmes fichiers. Même si nous ne travaillions pas sur les mêmes fonctions, les différents «commit» sur GitHub entraînaient des duplications ou la suppression de certains morceaux de code. De plus, le fait de tous travailler sous des systèmes d'exploitation différents nous ont obligé de trouver des astuces pour pouvoir compiler les différentes bibliothèques. Nous avons décidé de passer par un système d'exploitation Unix, plus pratique.

Chapitre 4

Bibliographie

Sujet du Projet : <http://dept-info.labri.u-bordeaux.fr/ENSEIGNEMENT/edd/projet/sujet.pdf>
2048 game : <http://gabrielecirulli.github.io/2048/>
Doc SDL : <http://www.libsdl.org/release/SDL-1.2.15/docs/>
Doc SDL-ttf : http://www.libsdl.org/projects/SDL_ttf/docs/SDL_ttf.pdf