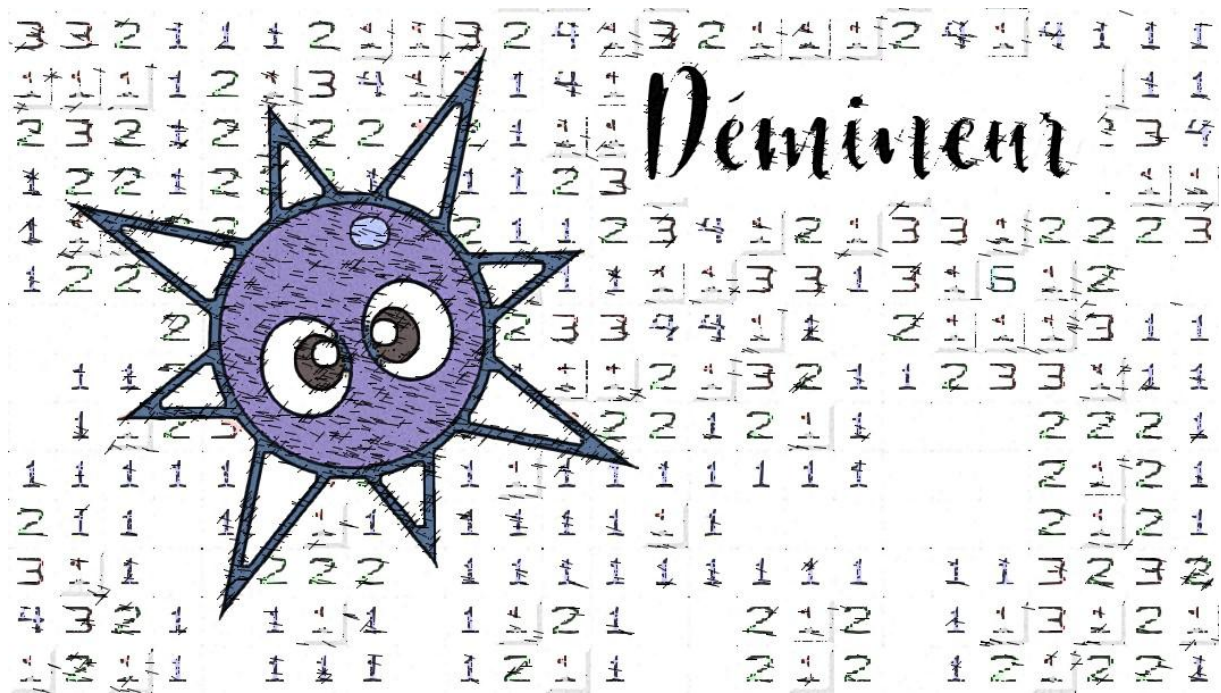


DEV4 - C++II



Remise1

Février 2018

Executive summary

Le jeu Démineur est un jeu composé d'un tableau de cases. Les cases peuvent être soit vides, soit contenir une bombe.

Le but du jeu est de découvrir toutes les cases qui ne contiennent pas de bombes en un minimum de temps, et cela sans faire exploser une seule bombe. Si le joueur découvre une case contenant une bombe, la partie est terminée.

Si le joueur soupçonne une case de contenir une bombe, il peut la marquer par un drapeau.

Dans le cadre de ce projet, le jeu possédera uniquement un mode solo.

Pour modéliser et gérer le jeu, le choix fut porté sur 8 classes : Board, Game, Observable, Observer, Player, Position, RulesEndGame, Time.

Sommaire

<i>Executive summary</i>	3
<i>Introduction</i>	5
<i>Les écarts</i>	6
L'affichage.....	6
Enregistrement des scores	6
La taille du plateau	6
Les bugs	6
Les testes	6
Les warnings.....	6
<i>Problemes rencontrer</i>	7
Problemes software.....	7
Problemes de developement	7
<i>Estimation du temps</i>	7
<i>Modification de la premiere analyse</i>	8
<i>Conclusion</i>	10
<i>Bibliographie</i>	11

Introduction

Dans ce document, nous allons aborder le développement du jeu Démineur dans le langage de programmation C++. Actuellement, nous n'aborderons que la partie modèle du programme. Cela signifie que nous allons développer les méthodes nécessaires à la gestion du jeu.

Pour chacune des classes, nous détaillerons les modifications au niveau des différentes méthodes et attributs ajouter, supprimer et modifier.

Certains bugs peuvent encore éclore car nous avons pensés à certains tests mais malheureusement dans le temps imparti nous n'avons pas pu tout tester ou nous n'avons tout simplement pas pensés à certains tests.

Les règles du jeu seront décrites brièvement afin de placer la modélisation métier dans son contexte.

Les écarts

L'affichage

Nous avons pour la classe board deux affichages. La méthode `displayBoard()` affiche le plateau sans cacher les bombes, ce qui est utile lors du débbugage et des tests. La méthode `diplayBoardForPlayer()` qui elle cache les bombes au joueur, elle n'affiche que les position explorer, 'c', celle avec des indices, '[1 à 8]', et les cases sombres, 'd'.

Enregistrement des scores

Par rapport à l'énoncé, nous avons, qu'en cas de victoire enregistrée, le nom du joueur, la taille du plateau, le nombre de bombe et le temps mis par le joueur.

Nous n'avons pas vérifié si ces paramètres ont déjà été enregistrés. Cela correspond plus à un historique des palmarès.

La taille du plateau

La taille du plateau est demandée mais n'est pas définit par défaut. Le joueur doit dans tout les cas rentrer une valeur entière.

Cette valeur est comprises entre 5 et 20, cette limite permet à avoir un affichage console agréable.

Pour la partie graphique nous pourrions changer cela selon la taille de la fenêtre avec tout de même des valeurs maximales qui garantiront un affichage agréable.

Les bugs

Ce qu'il y a comme bug rencontrer jusqu'à présent est que lorsque l on demande le coordonner x et y parfois il y a des bugs et une exception est lancée.

D'autres bugs peuvent encore être trouvée car nous n'avons pas pu tout tester par manque de temps ou simplement car nous n'y avons pas pensés.

Les tests

Nous n'avons pas eux le temps de tester la classe game.

La classe time n'as pas réussis à être testée car nous ne savions pas comment nous avons essayé avec la méthode sleep ou l utilisation de thread mais cela n'as pas fonctionné.

Les warnings

Nombreux warning ont été supprimés suite à des correction mais une partie semble avoir disparu entre les dernières compilations sans avoir été corriger. Sur les machines de l'école lors de la dernière exécution aucun warning ne sais afficher et de même sur nos machines personnel.

Problèmes rencontrer

Problemes software

Lors de l'élaboration du jeu nous avons opter pour l'utilisation du template subdir.
Malheureusement suite à l'utilisation du headers catch.hpp le programme ne compilait plus.

Après 2 semaines de tentative de modification pour corriger cela, nous avons optés pour une structure classique repartie en différents packages.

Les packages sont repartis comme suit :

- headers

 - model

 - observer

 - tests

- sources

 - model

 - observer

 - tests

Le main se trouve au même niveau que le package model, observer et tests.

Le package model contient les classe board, game, player, position, ruleendgame et timeT.

Le package observer contient les classes observer et observable.

Le package tests contient les classes boardtest, gametest, playertest, positiontest, rulesendgametest et timetest.

Problèmes de développement

Pour la partie model la plus grande difficulté fût porter sur la methode revelentpositionrecursif() qui permet d'éclairer le ou les zones aux alentours d'une position sélectionnée. Nous avons eue plusieurs version qui se rapprochaient fortement du résultat escompté sans réellement y parvenir.

Au final la méthode semble correcte et apporte le résultat escompté.

Estimation du temps

Pour l'estimation du temps nous avons perdu énormément de temps pour la tentative de correction de l'utilisation du template subdir et du header catch.hpp.

Suites a cela, il a fallut faire une réorganisation complète du projet. Nous avons attendu 2 semaines avant de faire cela.

En terme de temps, nous estimons a 80h le temps du développement du model testes compris.

Modification de la première analyse

En majeur partie, notre première analyse fut bonne.

Nous avons changé suites au debriefing après la remise, qui est observer et observer.

Pour la classe Game, c'est principalement cette classe ou plusieurs méthodes manquaient. Pour la gestion de l'entrée au clavier pour la sélection de la taille du plateau, pour le type de partie sélectionnée selon le nombre de bombes, les positions sélectionnées à jouer.

L'affichage globale pour informer le joueur également plusieurs méthodes ont été ajouter a la classe Game .

Nous avons utilisé également vos deux fichiers qui permettent la lecture au clavier de strings et d'entiers, keybord.h et stringConvert.

Conclusion

Le plus difficile fût de développer la méthode pour révéler les cases claires et placer les indices où il le faut.

Certains écarts sont présents par manque de temps.

La plus grande ressource qui manque pour le développement fut le temps.

Bibliographie
