

DEV4

Projet de C++

R. Absil, J. Beleho, N. Vansteenkiste

Année académique 2017 - 2018

Ce document détaille les consignes communes relatives au projet de C++, réalisé dans le cadre de l'unité d'enseignement DEV4. Le but de ce projet est d'implémenter par groupe de deux étudiants une version du jeu « Démineur », un jeu au tour par tour à un seul joueur où le but est de détecter des mines sur un plateau rectangulaire.

Table des matières

| | | |
|----------|--|----------|
| 1 | Introduction | 2 |
| 2 | Règles du jeu | 2 |
| 2.1 | Révélation et marquage de case | 3 |
| 2.2 | Génération de plateau | 3 |
| 2.3 | Fin de jeu | 5 |
| 3 | Modalités de remise | 6 |
| 3.1 | Modélisation métier | 6 |
| 3.2 | Implémentation console | 7 |
| 3.3 | Implémentation graphique | 7 |
| 4 | Conclusion | 7 |

1 Introduction

Démineur est un « très ancien » jeu vidéo, distribué sur diverses plateformes. Une partie, disputée seule, consiste à détecter des mines sur un plateau. Le but de ce projet est d'implémenter une version complète du jeu Démineur, par groupe de deux étudiants, incluant à la fois une interface console et une interface graphique.

Le développement de ce projet est divisé en trois étapes. La première consiste à modéliser dans son entièreté le programme, la deuxième à en implémenter complètement la partie métier et son interface console, et la troisième à implémenter une interface graphique. Le but de ces remises n'est pas de forcer l'étudiant à fournir plus de travail, mais à développer de bonnes pratiques de conceptions de programmes de taille « relativement conséquente ».

En effet, plusieurs remises intermédiaires permettent au professeur de suivre le groupe dans son parcours. Ainsi, si une première remise est de qualité très médiocre, le professeur a l'opportunité de corriger les erreurs au sein du projet afin de ne pas pénaliser outre mesure le groupe lors de la suite du développement.

Les remises de ces étapes sont programmées les semaines du 19 février, 26 mars et 7 mai. Notez que bien que ces travaux soient en groupe, rien ne garantit que les étudiants au sein du groupe auront la même note, dans la mesure où leur travail peut être considérablement inégal, ou les réponses aux questions qu'ils fourniront dans des défenses potentielles pourront être de pertinences différentes.

Ce document détaille donc les règles du jeu Démineur, ainsi que les modalités de remise. Plus particulièrement, la section 2 décrit les règles du jeu dans son intégralité. Ensuite, la section 3 détaille les spécificités des remises intermédiaires. La section 4 conclut ce document et en rappelle sommairement le contenu.

2 Règles du jeu

Démineur se joue sur un plateau rectangulaire de longueur et largeur arbitraires, dont le contenu est caché. Certaines cases contiennent des mines. À chaque tour, l'utilisateur peut soit choisir de révéler une case, soit d'en marquer une comme « minée ». S'il révèle une case minée, il a perdu la partie. Il remporte la partie quand toutes les cases ne contenant pas de mines sont révélées. Il est donc, *a priori*, inutile de marquer les mines obligatoirement¹. Le but du jeu est de remporter la victoire en un temps minimum.

1. Le marquage de cases comme minées est donc, en quelque sorte, un aide-mémoire.

2.1 Révélation et marquage de case

À chaque tour, l'utilisateur peut :

1. *marquer* une case comme *minée*,
2. soit *révéler* une case non relevée et non marquée,
 - si la case révélée contient une mine, il perd immédiatement la partie ;
 - si la case révélée ne contient pas de mine, il est nécessaire de calculer le nombre total de mines contenues dans les cases *voisines* :
 - si ce nombre est non nul, il est affiché sur la case révélée ;
 - sinon, la case est laissée vide et toutes les cases adjacentes sont révélées ;
3. enlever une marque de case.

Les voisins d'une case sont définis comme l'ensemble des cases directement adjacentes à une case donnée, en incluant les voisins diagonaux. Ainsi, sur un plateau rectangulaire, les coins ont 3 voisins, les cases sur les bords qui ne sont pas des coins en ont 5, et toutes les autres cases en ont 8.

La figure 1 illustre un exemple de plateau et de révélation de case. Sur la figure 1(a), on a représenté un plateau initial, où toutes les cases sont « cachées² ». La figure 1(b) illustre un plateau en cours de jeu : certaines cases sont cachées (en gris), certaines cases sont marquées comme minées (avec un petit drapeau), certaines cases indiquent le nombre total de voisins minés (en blanc, avec un chiffre), et certaines autres sont révélées et laissées vides (en bleu), indiquant le fait qu'elles ne contiennent pas de voisins minés. La croix rouge indique la case que l'utilisateur a l'intention de révéler.

La figure 1(c) illustre le résultat de cette révélation : cette case ne contenant pas de mines, les voisins ont été récursivement révélés. Si l'utilisateur avait révélé la case située immédiatement en dessous, il n'aurait révélé que le « 1 » à l'extrême gauche du plateau. Si l'utilisateur avait révélé la case située deux cases en dessous, il aurait révélé une mine et aurait perdu. Finalement, la figure 1(d) illustre un plateau victorieux : toutes les cases ne contenant pas de mines sont révélées, et les autres sont soit marquées comme minées, soit non révélées.

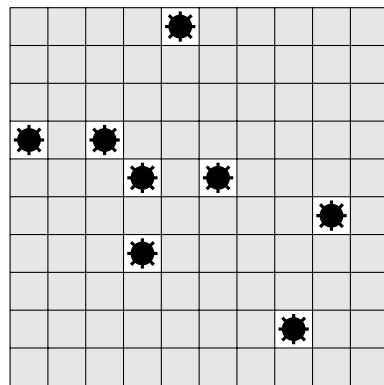
2.2 Génération de plateau

En début de partie, il convient de générer un plateau contenant un certain nombre de mines. Les contraintes à respecter sont les suivantes :

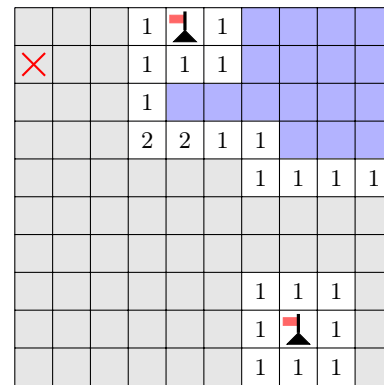
- les plateaux sont des grilles rectangulaires de longueur et largeur arbitraires (par défaut : 9×9),
- la première case révélée sur un plateau ne contient *jamais* une mine,
- les plateaux contiennent un nombre de mines arbitraire. Par défaut, un plateau de longueur m et largeur n contient

$$0,0002 \cdot (mn)^2 + 0,0938 \cdot (mn) + 0,8937 \quad (1)$$

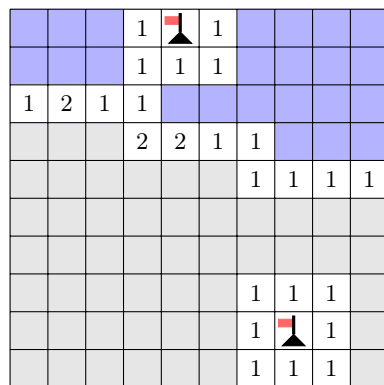
2. On a ici explicitement indiqué où se trouvent les mines, afin de savoir, pour l'exemple, si un clic fait perdre le joueur.



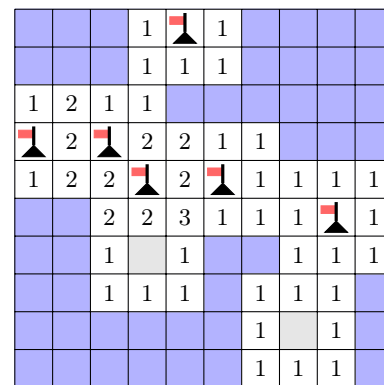
(a) Plateau initial, miné



(b) Plateau en cours de jeu, avant clic



(c) Plateau en cours de jeu, après clic



(d) Plateau victorieux

FIGURE 1 – Exemple de plateau miné et son affichage partiel et complet

mines³, arrondi à l'entier le plus proche.

Par exemple, un plateau de taille 9×9 contient

$$0,0002 \cdot (81)^2 + 0,0938 \cdot (81) + 0,8937 = 9,8037$$

mines, arrondi à 10. Ce nombre correspond aux instances « par défaut » de nombreuses implémentations de démineur.

Notez qu'à l'évidence, même si l'équation (1) donne un nombre par défaut de mines sur un plateau de taille arbitraire, vous devez également offrir la possibilité à un utilisateur de choisir lui-même le nombre de mines qu'il souhaite dans son plateau, soit en donnant cette valeur, soit en donnant la densité de mines contenues dans le plateau (par exemple : « je souhaite un plateau contenant 40% de mines »).

Variante éventuelle

Si vous disposez d'assez de temps, vous pouvez envisager une variante du plateau de jeu, en mettant des cases par exemple triangulaires, hexagonales, etc. La seule notion à adapter est donc celle de voisins (et de densité par défaut de mines). Notez que cette suggestion *ne fait pas* partie des exigences du projet : *si* vous avez rempli toutes les exigences, *alors* vous *pouvez* implémenter cette variante.

2.3 Fin de jeu

La partie est terminée

- soit quand un utilisateur révèle une case contenant une mine⁴, ce qui le fait perdre immédiatement,
- quand toutes les cases du plateau ne contenant pas de mines sont révélées, ce qui lui fait remporter la partie. Cette situation est illustrée à la figure 1(d).

En cas de victoire, le temps que l'utilisateur a pris pour finir le jeu est enregistré sur un « hall of fame » listant les meilleurs temps de joueurs en fonctions des tailles et densité de plateaux choisies. Plus particulièrement, pour chaque taille de plateau et densité de mines, vous devez enregistrer les scores (c'est-à-dire les meilleurs temps) des 5 joueurs les plus rapides. Il est sûrement utile d'offrir à l'utilisateur d'enregistrer son pseudonyme le cas échéant.

Il est probablement ergonomique d'afficher constamment le chronomètre, afin de permettre au joueur de savoir depuis combien de temps il a commencé la partie.

3. Ce nombre ne fait pas vraiment partie des règles « officielles » de Démineur, qui n'autorise pas des longueurs et largeurs arbitraires de plateau. Il a été obtenu par interpolation de la densité de mines dans les plateaux autorisés.

4. Pour rappel, la première case révélée ne peut contenir de mine.

3 Modalités de remise

Dans la mesure où ce projet est réalisé sur un quadrimestre, plusieurs remises intermédiaires sont demandées.

Le but de ces remises n'est pas seulement de forcer l'étudiant à travailler régulièrement, mais à pouvoir le corriger rapidement. Ainsi, une prise de décision significativement mauvaise au début de la conception du projet sera corrigée par le professeur et aura un impact limité sur la suite du développement.

À ce titre, trois remises sont demandées :

1. remise métier la semaine du 19 février : modélisation métier du programme ;
2. remise métier la semaine du 26 mars : implémentation des classes métiers et d'une interface en console ;
3. remise graphique la semaine du 7 mai : implémentation d'une interface graphique sur base des classes métiers de la deuxième remise ;

Remarquez qu'il vous est demandé d'implémenter le design pattern « Observateur / Observé », ou « Modèle / Vue / Contrôleur », notamment pour les interactions avec l'utilisateur. Ainsi, il est *inadmissible* d'avoir des dépendances de la partie métier de votre projet vers la partie contrôleur, console ou graphique !

Notez également que si, lors de l'implémentation de la partie console ou graphique, vous deviez vous écarter de la modélisation prévue lors de la première remise, il convient de justifier judicieusement ce changement.

Par ailleurs, si vous utilisez des bibliothèques tierces pour certaines parties de votre travail (par exemple pour effectuer des tests unitaires), vous devez faire en sorte que le travail remis soit « autosuffisant ». Ce n'est pas à votre professeur de régler les problèmes de dépendances externes (entre autres) ! Ceci exclut l'utilisation de bibliothèques non portables (`windows.h`, `unix.h`, etc.), ainsi que des chemins absolus vers des ressources, etc.

3.1 Modélisation métier

Ce travail est à remettre pour la semaine du 19 février. Vous devez, à ce stade, remettre une modélisation documentée de la partie métier (c'est-à-dire hors interaction clavier / souris utilisateur) de votre projet. Dépendant des exigences de votre maître-assistant, cette remise peut prendre diverses formes, telles que des headers documentés, d'un rapport, de diagrammes de classes, de séquence, de cas d'utilisations, etc.

En aucun cas, en l'occurrence, il ne vous est demandé de remettre du code fonctionnel. Votre travail doit, à ce stade, bien représenter les différents composants de votre projet, leurs interactions, etc.

3.2 Implémentation console

Ce travail est à remettre pour la semaine du 26 mars. Le but de ce travail est de remettre un projet complètement implémenté, avec interface console (dont enregistrement des scores), qui réponde aux fonctionnalités décrites en section 2. À chaque tour, vous devez donc afficher le plateau dans la sortie standard, et demander à l'utilisateur ce qu'il souhaite faire.

3.3 Implémentation graphique

Ce travail est à remettre pour la semaine du 7 mai. Le but de ce travail est de remettre un projet complètement implémenté, avec interface graphique, qui réponde aux fonctionnalités décrites en section 2. Vous devez réaliser votre interface graphique à l'aide du framework Qt.

4 Conclusion

L'énoncé du projet, ses consignes, dates de remises et exigences ont été décrites. Pour rappel, il faut créer une version complète et distribuée du jeu Démineur, par groupe de deux étudiants. Notez que bien que ces travaux soient en groupe, rien ne garantit que les étudiants au sein du groupe auront la même note.

Plusieurs remises intermédiaires sont demandées afin de vous suivre dans votre travail, respectivement le 19 février pour la partie métier de votre travail, le 26 mars pour l'implémentation de l'interface console et le 7 mai pour l'implémentation de l'interface graphique.

Notez également que les exigences exposées ici sont celles communes aux maîtres-assistants en charge de l'unité d'enseignement DEV4. Il convient donc de vous renseigner des contraintes supplémentaires spécifiques de votre maître-assistant.