

Programmation Impérative Projet S2 : Color Flood, lot B

Guillaume Aichhorn, Quentin Durand, Rémi Faure-Gignoux et Romanus
Rosari

Un Projet Ludique, et Sérieux

Table des matières

| | |
|--------------------------------------|---|
| Programmation Impérative Projet S2 : | |
| Color Flood, lot B | 1 |
| Introduction..... | 2 |
| 1 Objectifs de cette session | 3 |
| 2 Le code..... | 3 |
| Boucle de jeu | 3 |
| 3 Répartition des tâches | 5 |

Présentation du sujet

Après avoir mis en place les outils de base du jeu, telles que la vérification des conditions de victoire et la gestion des modifications de la tache. Nous nous intéressons ici à la mise en place d'une boucle de jeu. Notre objectif pour cette étape est en effet le développement d'une application jouable. On a aussi découvert des erreurs dans les fichiers envoyés dans le lot A, celles-ci ont été corrigées afin de permettre le fonctionnement de ce qui sera proposé ici.

1 Objectifs de cette session

- Lecture au clavier : Le joueur doit pouvoir décider de la taille de la grille en début de partie, puis sélectionner les couleurs.
- Affichage dans le terminal : Dans un premier temps on affiche le jeu dans le terminal...
- ...Pour ensuite proposer une interface graphique, via la bibliothèque SDL.

2 Le code

Affichage dans le terminal

- On s'assure d'abord que le flux soit vide, avant d'y afficher quoique ce soit, et ce grâce à la fonction *purger* du fichier *debut.c*.
- Dans le même fichier on trouve ensuite la fonction *demande_int*, celle-ci permet l'entrée au clavier de la taille du plateau, sa validation, et la purge du flux d'écriture. On l'utilisera aussi, tant qu'on a pas de solver, pour demander un nombre de coups maximum.
- On représente les couleurs par leurs initiales, et on les affiche avec retour à la ligne en fin de ligne, dans le terminal grâce à la fonction *affiche_grille* du fichier *LotB_affichage*. Elle permet aussi l'affichage du nombre de coups restants, passé en argument, on sépare les lignes du tableau par des lignes de tirets et les termes d'une ligne par des pipes.
- Le fichier *LotB_affichage* présente enfin une fonction *sel_coul* de sélection de la couleur. Elle permet une lecture au clavier d'un entier entre 0 et 5, associés chacun à une des couleurs (on précise au joueur lesdites associations) en vérifiant que l'entrée proposée par le joueur est valide. Le cas échéant on demande au joueur de rentrer une valeur correcte.

Boucle de jeu On dispose enfin d'un fichier *main* qui permet l'exécution de la boucle de jeu, pour ce faire, on demande la taille de la grille, grâce à la fonction *demande_int*, ainsi que le nombre de coups, ce qui reste accessoire, et permet ici uniquement de tester une condition de défaite.

On génère ensuite une grille de la taille voulue grâce à la fonction *creer_grille* du lot précédent. Dès lors on introduit une variable *victoire* qui sera assimilée, à

chaque tour au résultat de *cond_gagner*, on entre ensuite dans une boucle conditionnelle. Tant qu'il reste des coups, et tant qu'on a pas gagné, on sélectionne une couleur, on met à jour la tache, et on décrémente le nombre de coups de 1. On met aussi à jour *victoire*. Une fois sorti, on affiche un message annonçant la victoire ou la défaite du joueur.

SDL

On fait ici appel à la bibliothèque SDL qui permet de gérer une interface graphique. Une fois la grille affichée on dispose d'icônes de couleurs sur lesquels cliquer pour sélectionner les couleurs. On fait attention à distinguer le noir (couleur de fond) des autres couleurs, sinon, on ajoute une couleur en cas de clic sur le fond... On fait, comme pour la version sur terminal, appel aux fonctions du lot A pour mettre à jour la tache.

Bugs non corrigés

On a remarqué des erreurs de segmentation, lorsque la tache atteint la dernière colonne. Cela semble du à une erreur d'indexation, mais on n'a pas réussi à la repérer. De plus, plus surprenant, on aussi constaté la même erreur, lors de la sélection de la couleur rouge, dans certains cas, qui semblent aléatoires, et n'avons pas réussi à trouver de raisons à ce bug.

3 Répartition des tâches

Romanus Rosari

- Application des transformations de la grille sur le terminal (nettoyage)
- Test de la fin de partie sur terminal
- Sélection d'une couleur sous SDL
- Affichage sous SDL
- Test des conditions de fin, sous SDL

Rémi Faure-Gignoux

- Application des transformations de la grille sous SDL
- Sélection d'une couleur sous SDL
- Affichage sous SDL
- Test des conditions de fin, sous SDL
- ReadMe

Quentin Durand

- Lectures au clavier
- Affichage sur terminal
- Selection d'une couleur via terminal

Guillaume Aichhorn

- Test des conditions de fin de partie
- Rapport