

Rapport Color Flood

Lot-B



The A-Team

Loubna Anthea
Abdellah Bader

Projet IPI2

ENSIIE
Telecom Physique

Sommaire

I. Répartition du travail

- 1) L'équipe du travail.....p3**
- 2) Le rôle de chacun des membres.....p4**
- 3) Les rencontres.....p5**
- 4) La communication entre les membres.....p6**

II. Conception algorithmique

- 1) Jeu en mode texte.....p6**
- 2) Introduction à SDL.....p7**
- 3) Jeu en mode graphique.....p8**

II. Documentation et tests mémoires

- 1) Vérification de l'absence de fuites.....p10**
- 2) Documentation du code.....p10**

I- Répartition du travail

1-L'équipe du travail



L'équipe du projet est constituée de 4 membres :

- ◆ Loubna Hennach (INOC)
- ◆ Anthea Mérida Montes de Oca (ENSIIE)
- ◆ Abdellah Elazzam (INOC)

- ◆ Bader Khatofi (INOC)

2-Le rôle de chacun des membres

Anthea Mérida :

- référente pour le lot B du projet
- gère la communication entre les membres de l'équipe
- organise les rencontres dans différents lieux (BNU, Bibliothèque TPS)
- Chargée de la programmation de la boucle de jeu en mode texte

Loubna Hennach :

- Chargée de la rédaction du rapport
- Chargée de la rédaction du README
- Chargée de la génération du rapport Doxygen.
- Chargée de la programmation de la boucle de jeu en mode texte

Abdellah Elazzam:

- Chargé de la mise en place de l'interface graphique avec SDL.
- Chargé de l'adaptation de la boucle de jeu en mode graphique.
- Chargé des commentaires Doxygen et du rapport pour la partie graphique.

3-Les rencontres

Tout au long de la période consacrée à la réalisation du Lot B du projet, nous avons essayé de nous réunir pour faire le point sur l'avancement du travail. Une communication presque permanente nous était garantie grâce à l'outil Scoledge (partie communication).

Le planning des rencontres et le compte rendu de chacune est le suivant :

➤ **1ère rencontre:**

Date et Lieu : Samedi 4 Mars à la BNU

Détails de la rencontre :

- ◆ Programmation de la boucle de jeu en mode texte.
- ◆ Mise en place de l'outil graphique

Point d'avancement :

- ◆ Jeu en mode texte marche.
- ◆ Nouveau Makefile

➤ **2ème rencontre:**

Date et Lieu : Vendredi 10 Février à L'ENSIIE

Détails de la rencontre :

- ◆ Amélioration de la boucle de jeu en mode texte
- ◆ Programmation de la boucle de jeu en mode graphique
- ◆ Tests de fuite mémoires avec Valgrind
- ◆ Rédaction du README

Point d'avancement :

- ◆ Jeu en mode texte et en mode graphique.
- ◆ Rédaction du rapport
- ◆ Vérification du sprint.

4)La communication entre les membres

Nous nous sommes servis des outils qui avaient été mis en place pour le premier sprint pour assurer la communication entre les membres de l'équipe. Notamment Github nous a permis de mieux gérer les changements apportés par chacun aux différents éléments du lot.

Cependant, nous avons eu quelques difficultés concernant l'organisation du travail. En effet, on a remarqué que quand on travaille ensemble dans le même lieu on arrive à mieux nous coordonner, alors que quand il s'agit de travailler individuellement, on a plutôt tendance à oublier de mettre à jour le dépôt Github ou de communiquer avec les autres sur ce qu'on a fait.

Pour remédier à cette situation, on essayera de travailler ensemble plus souvent, ce qui fera que la communication deviendra plus fluide.

II – Conception algorithmique

1)Jeu en mode texte:

Les fonctions de base pour la boucle de jeu en mode texte sont implémentées dans `loop_game.c`, leur prototypes sont implémentés dans `loop_game.h`

La fonction `main` qui déroule la boucle de jeu est implémentée dans `main_jeu.c` qui a pour exécutable `jeu_texte`.

1.1 Fonctions de base de la boucle de jeu:

- get_color: permet à l'utilisateur de saisir une couleur
- get_size: permet à l'utilisateur de saisir la taille de la grille désirée
- get_nombre_coups: permet à l'utilisateur de saisir le nombre de coups autorisés.
- Turn: permet d'effectuer les changements majeurs à la grille à chaque tour de jeu.

1.2 Boucle de jeu:

game : dans une boucle while, on fait appel aux fonctions de base pour dérouler le jeu.

1.3 Test de fin de jeu:

check: si toutes les cases de la grille sont de la même couleur et que l'utilisateur ne rentre pas « Q » on affiche un message de victoire.
Si l'utilisateur dépasse le nombre de coups autorisés on sort de la boucle de jeu avec un message « Vous avez perdu »
Si l'utilisateur rentre « Q » on sort du jeu et donc partie non terminée

2)Introduction à SDL

La SDL (*Simple DirectMedia Layer*) est une bibliothèque logicielle permettant de développer des programmes gérant le clavier, la souris pour une meilleure visibilité du jeu.

3) Jeu en mode graphique

Les fonctions de base pour la boucle de jeu en mode graphique sont implémentées dans SDL.c, leur prototypes sont implémentés dans SDL.h

La fonction main qui déroule la boucle de jeu est implémentée dans Jeu.c qui a pour exécutable Jeu.

3.1 Fonction d'affichage du menu :

L'affichage menu permet un déplacement par une flèche rectangle entre trois options, la première pour le choix de la taille, la deuxième pour le choix du nombre de coups autorisés et finalement un choix pour quitter le jeu.

L'outil flèche est basé sur le dessin de trois carrés : un qui est le choix de l'option en question en noir et les deux autres en blanc en fonction de la position de la flèche. Cette solution est due au fait que je n'ai pas trouvé une façon pour libérer les flèches (Surface_Rectangle) à chaque fois qu'on se déplace (une alternative était de tracer la couleur blanche au-dessus). Comme conséquence, cette méthode permet d'allouer une infinité de flèches sans qu'elles soient libérées (Fuites mémoires énormes surtout si on reste longtemps dans le menu).



3.2 Boucle de jeu :

La boucle de jeu commence par l’affichage de la grille en allouant sa taille suivant le choix fait en menu, ainsi que le compteur de mouvements par rapport au nombre de mouvements autorisés, et enfin une interface fixe pour les choix de couleurs.

La partie finit par montrer un message de victoire et, le cas échéant, le nombre de coups faits pour gagner, ou bien de défaite.

Contrairement à l’affichage menu, la boucle de jeu n’est pas gourmande en termes de mémoire, vu qu’il y a libération de surface à chaque coup ainsi que des appels à des fonctions qui permettent de libérer la mémoire comme

TTF_CloseFont(police),

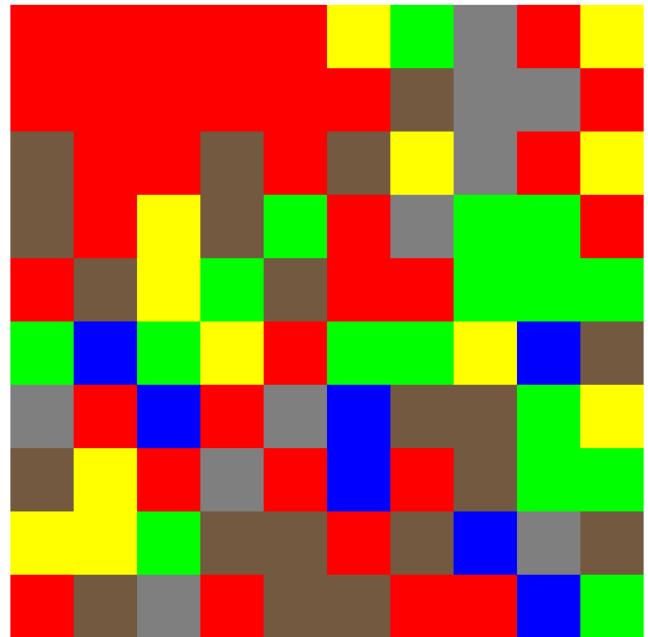
SDL_FreeSurface(Surface), TTF_Quit(),

SDL_Quit() ainsi la fonction de libération

de la grille free_grid(&grille). Un autre problème de fuites se voit dans le fait que la boucle de jeu est infinie et contient le menu plus la boucle de jeu, pour permettre une infinité de parties qui débutent avec les options de menu.



5/12



II. Documentation et tests mémoires

1)Vérification de l'absence de fuite mémoire:

On teste l'absence de fuite mémoire avec Valgrind pour l'exécutable jeu_texte de la boucle de jeu en mode texte, on a autant d'allocations que de free et donc pas de fuite mémoire.

2)Documentation du code:

Les commentaires en format doxygen sont inclus dans loop_game.h pour la boucle de jeu en mode texte, d'autres commentaires complémentaires sont inclus dans loop_game.c et main_jeu.c, ceci nous permet de ne pas avoir de commentaires dupliqués quand on génère le rapport Doxygen.

Comme pour le lot A ,le README contient toutes les instructions d'installation des outils mis en œuvre dans le projet; notamment l'installation de SDL, ainsi que les instructions de leurs utilisation.