INFO0030: Five or More

Groupe 24: Timothy Smeers, Martin NoirFalise

Table des matières

1 GUI

La GUI est gérée en plusieurs parties, ces parties sont gérées indépendemment les unes des autres. Chaque zone est gérée logiquement dans le modèle et gérée visuellement dans la vue. Les deux étant controlées par le controleur. (cfr. Figure 1).

— La zone rouge contient les menus du programme. Les sous-menu sont liés à des fonctions de callback propres à la vue.

— La zone bleue contient les pions de couleurs du prochain tour. Un vecteur fournit une référence vers chacun des boutons permettant de mettre à jour leurs image à chaque tour. Cette zone contient également l'affichage du score actualisé à chaques tours.

— La zone verte contient le grille de jeu. Une matrice des boutons conserve une référence vers chacun des boutons du tableau de jeu, cela permet de mettre à jour leur image lors d'un déplacement des pions par le joueur.

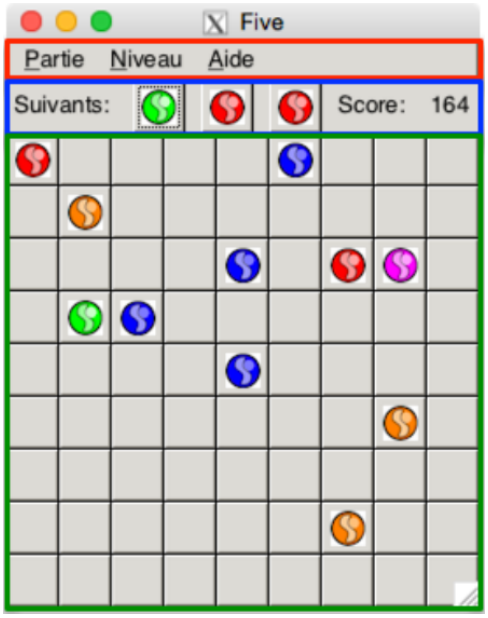

FIGURE 1 – Groupements dans la fenêtre

2 Structures de Données

2.1 modele.c

2.1.1 Struct Modele_t

Cette structure contient toute les données utiles à la gestion globale du jeu.

2.2 vue.c

2.2.1 Struct Vue_t

Cette structure contient en même temps la structure du modèle pour pouvoir avoir accès à toutes les données de jeu. Et contient également la matrice des boutons présents dans la grille, mais aussi les boutons suivants et le score à afficher sur la GUI

2.3 controleur.c

2.3.1 Struct Controler_t

Cette structure contient les deux structures, Modele_t et Vue_t.

2.3.2 Struct Data_tab_t

Cette structure contient les informations de la grille permettant de crer des g_signal_connect() ;

2.4 utils.c

Ce fichier contient tout les algorithmes simples et de conversion utiles dans plusieurs fichiers sources.

3 Algorithmes

Lors de l'élaboration de ce projet, nous avons développé plusieurs algorithmes complexes.
— L'algorithme de recherche A*.
— Algorithme de répartition aléatoire.
— Algorithme de suppression des pions gagnants.

3.1 Algorithme de recherche A*

L'algorithme de recherche A* est construit sur base de la récursivité et va prendre une à une les cases disponibles aux alentours jusqu'à atteindre (ou pas) la case d'arrivée en contournant les obstacles

3.2 Algorithme de répartitions aléatoires

Cet algorithme permet de tirer au hasard un nombre entre 1 et le nombre de cases vides restantes dans la grille de jeu pour ensuite les répartir dans cette même grille.

Le nombre de pions à placer dépend du niveau de jeu choisi :

3 pour le niveau "facile".
3 pour le niveaux "moyen".
7 pour le niveaux "difficile".

Une fois ces pions placés dans la grille, nous nous devons de réduire la taille du vecteur représentant les cases libres à sa nouvelle valeur.

3.3 Algorithme de suppression des pions gagnants

Le but de cet algorithme est de supprimer les pions qui ont permis l'incrémentation du score.

Une fois les pions supprimés, nous nous devons d'actualiser la taille du vecteur représentant les cases libres à sa nouvelle valeur.

4 Profiling du Code et Analyse de Performance

Après un profiling, à l'aide de GProf, de notre exécutable,
— Les deux fonctions les plus appelées sont (int /char*) et leur complexité est très légère.
— Elles sont appelées énormément de fois.

5 Interface Graphique GUI

L'interface est illustrée dans les figures :
2, 3, 4, 5, 6, 7, 8, 9, 10

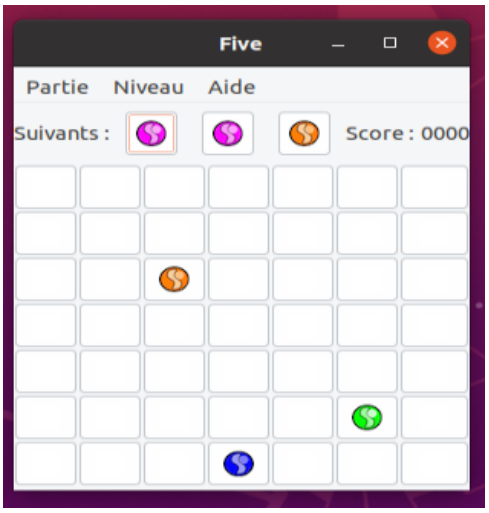

FIGURE 2 – Interface de départ pour un niveau facile.

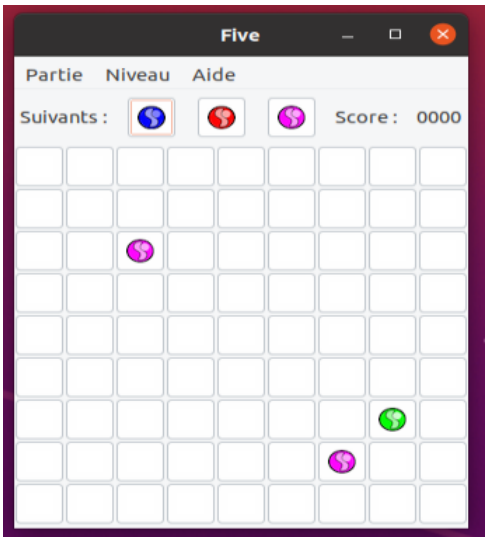

FIGURE 3 – Interface de départ pour un niveau moyen.

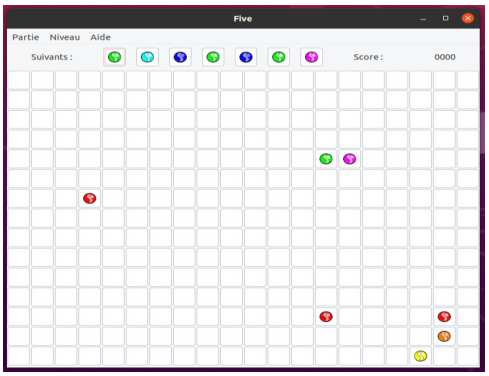

FIGURE 4 – Interface de départ pour un niveau difficile.

FIGURE 5 – Interface de fin lorsque le jeu est terminé, nous devons renseigner notre nom.

FIGURE 6 – Interface du menu de partie. L'utilisateur peut recommencer une nouvelle partie ,quitter le jeu ou consulter les scores.

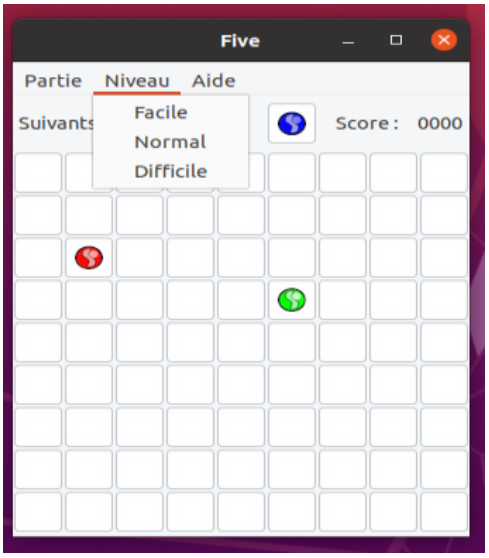

FIGURE 7 – Interface du menu niveau. L'utilisateur peut sélectionner un niveau.

FIGURE 8 – Interface du menu d'aide. L'utilisateur peut sélectionner les crédits.

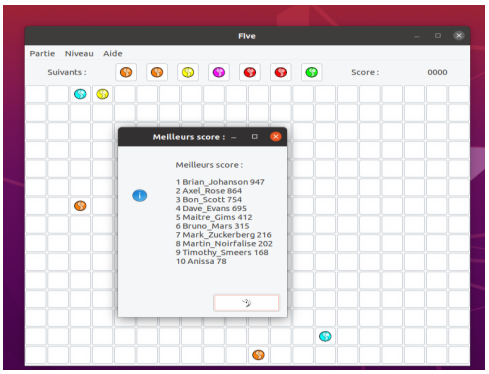

FIGURE 9 – Interface du pop-up de score. L'utilisateur peut visualiser les meilleurs scores.

FIGURE 10 – Interface du pop-up "à propos". L'utilisateur peut consulter les crédits.

6 Gestion du code

Lors de la réalisation de notre projet, nous avons directement utilisé Un SCM (Git) pour pouvoir travailler en même temps sur des fichiers en parallèle.

Le concept du SCM nous étant familier, nous n'avons pas eu de mal à nous en servir.

La gestion de tickets et de branches étant très utiles, nous avons jugé bon de nous en servir au maximum avant de commencer à faire des merge-request.

7 Coopération au sein du groupe

La coopération au sein de notre groupe à été très fructueuse et partagée. Nous ne pouvons pas préciser exactement les taches que nous avons chacun réalisées séparément. Mais nous avons travaillé de différentes façons.

7.1 Martin

Création de la VBox principale avec tous les menus et la grille. Élaboration des algorithmes.

7.2 Timothy

Déboguage et refactoring du code. Création des pop-up et algorithmes liés à la barre de menu.

7.3 Commun

Les tâches communes ont été très nombreuses avec quelques "Brain-Stormings" avant de commencer un nouveaux ticket pour mettre au point notre approche pour l'élaboration du programme.

8 Améliorations du programme

Si nous avions disposé d'un peu plus de temps nous aurions :

— Ajouté un meilleur algorithme de traitement des meilleurs scores.
— Ajout de différents pop-up's permettant de personnaliser la fenêtre de jeu.
— Algorithme A* beaucoup plus élaborer.
— Création de pions bonus permettant de supprimer plusieurs pions autours (tel une bombe).

9 Apprentissage

Ce projet nous a permis à tout les deux de pouvoir évoluer en groupe pour faire évoluer notre capacité à nous adapter à l'autre. Ce projet étant assez volumineux nous avons compris l'importance du déboguage et de la programmation défensive. Bien entendu ceci nous à permis d'apprendre à nous documenter d'une manière plus rigoureuse sur les librairies utilisées utiles à l'élaboration d'algorithmes.

10 Documentation

Pour plus d'informations sur le code vous pouvez consulter le site internet contenant la documentation doxygen.