École Internationale des Sciences du Traitement de l'Information

Rapport d'Informatique

PROJET QWIRKLE

Charles Duc
Luca Ordronneau

Tuteur : Peïo Loubière

Mai 2018 — Juin 2018

Table des matières

1 Introduction

Dans le cadre du projet du semestre 2, nous avons du coder le Qwirkle en Pascal. Le Qwirkle est un jeu qui est un mixte entre les dominos et le Scrabble. Afin de réussir à le coder nous avons du comprendre son fonctionnement et ces règles de base, tel que comment peut-on placer les pièces, comment marche le comptage de points. Pour réussir à appliquer ces règles nous avons créé des fonctions que nous allons vous expliquer ci-dessous leurs fonctionnements.
Pour un meilleur rendu visuel et pour des raisons de confort nous avons décidé de créer plusieurs Unit où nous répartissons les fonctions dans différents domaines.

2 Unit Type

Dans cette Unit on y a placer toutes les structures et types que nous utilisons dans notre projet.

Les principales étant le *ptr-piece* qui est l'adresse qui renvoi à une pièce, la *piece* étant une structure composée de couleurs(entier)/formes(entier)/suivant(adresse), qui nous permet de créer la pioche ainsi que les mains en liste de pièces.
Le *TypeGrille* est un tableau de N sur N, N étant le produit du nombre de tuiles du nombre de formes et du nombre de couleurs (on expliquera plus tard leurs utilités), le tableau est un tableau de *ptr-piece*.
On a aussi la structure *TypeMain* composée de deux listes et d'une pièce, qui joue un rôle primordiale dans nos fonctions de jeu.
De la même façon on a la structure *ToutEnUn* composé d'une main, d'une pioche, d'une grille et de points, elle nous permet de manipuler la pioche et la grille de façon beaucoup plus facile quand on fait appel aux deux à chaque tours.
On a aussi le *typeJoueurs* qui est une structure représentant un joueur, composé d'un âge, d'un nom, du nombre de points ainsi que de la main du joueurs.
Sans oublier la structure *GrillePoints* composé d'une grille et du nombre de points, elle nous permet de compter les points après que le joueur ai placé toute ces pièce pendant son tour.
Pour finir on a la structure *TypeParam* qui est la pour nous permettre de manipuler les paramètre facilement.

3 Unit afficheProjet

Dans cette Unit nous avons placé toutes les fonctions d'affichage et d'initialisation.

3.1 La grille

Tout d'abord pour l'initialisation de la grille nous avons décidé de faire de une grille de taille N sur N on a initialisé toutes les cases de la grille par une pièce de couleurs et formes 0 et de suivant NIL. Pour l'affiche on a du faire face a notre premier problème qui était de bien indiquer la bonne ligne et la bonne colonne afin de placer la pièce au bonne endroit. On a alors décidé à la place d'afficher la colonne et la ligne 0 d'afficher le chiffres correspondant au ligne et colonne, afin que le joueur puisse beaucoup plus facilement vire a quel colonne et ligne correspond une position.

3.2 La pioche

Pour le pioche, on l'initialise de façon aléatoire a l'aide du nombre de formes, de couleurs et de tuiles, ce qui nous permet d'avoir le bon nombre de pièces dans notre jeu. On se sert de trois boucle qui correspondent chacunes à la couleur, la forme ou la tuile et pour chaque pièces créée on la place à une position aléatoire dans la pioche.

3.3 La Main

On créée la Main d'un joueur en partant de la pioche de base. On rentre la pioche et ensuite on la sépare en fonction du nombre de pièces que l'on veut dans la main et on renvoi un *TypeMain* qui nous permet alors de renvoyer la main du joueur et la nouvelle pioche afin de créer les mains des joueurs suivants.

4 Unit Projet

Dans cette Unit, on y a mis le gros de nos fonctions celle qui nous permettent de créer l'ordre dans lequel les joueurs jouent ou encore remplacer une pièce de sa main par une de la pioche.

4.1 Créer l'ordre

Pour commencer il faut savoir que la liste comportant tout les joueurs de la partie est un tableau de *TypeJoueurs* nous créons donc tout les joueurs avec leur âges ,noms et mains. Ce qui nous permet ensuite de compter qui peut en jouer le plus et alors c'est ce joueur là qui commence la partie. Pour cela on se sert de la fonction *nbElements* qui compte combien il y a de pièce au maximum il peut jouer et ensuite on fait juste une comparaison très simple qui permet de savoir qui a la plus grand nombre de pièces pouvant être jouées.

4.2 La fonction remplace

Cette fonction nous permet de remplacer une pièce de la main par une pièce de la main et remettre la pièce de la main du joueurs dans la pioche a une position aléatoire.

4.3 La fonction jouer

Cette fonction nous permet de jouer une pièce de la main et de la poser sur la grille et piocher une pièce pour la remettre dans la main du joueur.

4.4 Les fonctions de placements

On a deux fonction de placement, la première va placer une pièce au milieu de la grille juste au premier tour ce qui permet de ne pas sortir de la grille quand on rajoute les pièces par la suite. La deuxième va permettre au joueur de placer ces pièces soit en colonnes soit en lignes, et alors choisir la position a laquelle il va pouvoir poser sa pièce.

4.5 Les vérifications

Pour vérifier si le joueur ne triche pas on a créé une fonction qui va permettre de vérifier si le joueur a le droit de placer la pièce en cette position, on vérifie alors si il y a au moins une pièce a coter de cette position et ensuite si d'après les règles il peut poser la pièce ici. Pour cela on a 4 fonctions différentes qui vérifie en haut, en bas, a gauche et a droite et on les appelle récursivement ce qui nous permet d'éviter toutes triches.

4.6 Le comptage des points

On se sert du même principe que pour les vérifications mais la on compte les points que le joueur a marquer après qu'il est placer toutes les pièces qu'il voulait placer durant son tour.

4.7 La fonctions de fin de jeu

Pour que le jeu se termine on doit respecter deux conditions, que la pioche sois vide et que un des joueurs ai fini sa main. Nous avons donc créée une fonction qui vérifie ces deux conditions.

5 Le programme Qwirkle

Dans cette partie du programme on a notre programme principale et toute les fonctions qui nous permette de faire tourner le jeu, c'est là que l'on manipule toute nos autres fonctions.

5.1 La fonction *remplacertt*

On appelle cette fonction dans notre fonction *testrempljoue*, et elle va nous permettre de remplacer un nombre de pièce que l'on choisi de sa main et de piocher pour les remplacer.

5.2 La fonction *jouertt*

On l'appelle aussi dans *testrempljoue* cette fois pour jouer un nombre de pièce choisi dans la grille, on ne peut alors que jouer sois sur une ligne sois sur un colonne.

5.3 La fonction *jouerttpremier*

C'est la même fonction que *jouertt* mais on l'appelle que le premiers tours et elle place directement dans la grille sans demander les positions.

5.4 La fonction *testrempljoue*

Dans cette fonction on fait choisir au joueur si il veut remplacer ou de jouer en appelant les fonctions *jouerttpremier*, *jouertt* et *remplecertt*.

5.5 La fonction *run*

Dans cette fonction on y appelle la fonction *testrempljoue* et on modifie donc la main du joueur dans la tableau de joueur ainsi que la grille et la pioche, on y modifie aussi le nombre de points pour le joueur qui joue.

5.6 La procedure *appelrun*

Dans cette fonction on appelle la fonction *run* et on vérifie si la jeu est fini ou non et le jeu est fini on affiche le gagnant.

5.7 La fonction *Param*

Elle nous permet de gérer les paramètres quand on veut lancer le jeu. On peut alors définire le nombre de formes, de couleurs ou encore de tuiles en lançant le jeu, même si il y a des paramètres de base qui sont entrés dans la fonction.

5.8 La procedure *lecturejeu*

Dans cette procédure on y appelle la fonction *Param* afin de définire les paramètres, on peut alors grâce a ces paramètres initialiser la grille et la pioche et ainsi commencer a jouer.

6 Conclusion

Pour l'IA, elle est opérationnel mais il manque le comptage de point.
On ne peut jouer que entre humains ou entre IA nous n'avons pas coder quand l'on peut jouer avec les deux.
Nous pensons que il aurait fallu un peu plus de temps afin que nous puissions finir le projet en entier.
Nous espérons que vous pourrez jouer sans problèmes au Qwirkle.