

DARMAISIN Clément
DUFFAUT Julien

Rapport de projet Java :

Jeu de Pente

[Introduction](#)

[Présentation du projet](#)

[Analyse](#)

[Classes utilisées](#)

[Relations entre les classes](#)

[Fonctionnement global](#)

[Réalisation](#)

[Choix techniques](#)

[Fonction CheckCase\(\)](#)

[Format de fichiers](#)

[Utilisation](#)

[Mode d'emploi](#)

[Configuration requise](#)

[Conclusion](#)

[Bilan](#)

[Optimisations possibles](#)

[Extensions possibles](#)

Introduction

Présentation du projet

L'objectif de ce projet était de recréer le jeu de Pente d'Hasbro. Dans ce jeu, chaque joueur place des pions à tour de rôle sur un plateau de jeu de 19*19 cases. Si un joueur parvient à aligner cinq pions, il gagne la partie. Si deux pions se font entourer de pions de la couleur opposés, ils changent eux-mêmes de couleur. Si un joueur arrive à convertir cinq paires de pions adverses, il gagne également la partie.

Analyse

Classes utilisées

Pour ce projet, nous avons utilisé une classe "Principal" contenant la fonction main, une classe "FenetrePrincipale" créant la fenêtre, une classe "Quadrillage" dessinant le plateau de jeu et une classe "Pion" permettant de créer et de placer des pions.

Relations entre les classes

La fonction main contenue dans "Principal" crée une instance de "FenetrePrincipale" et en appelle par conséquent le constructeur. Celui-ci crée la fenêtre en lui attribuant différentes propriétés (comme la taille par exemple) et crée une instance de "Quadrillage" qu'il place sur la fenêtre. La fonction MouseClicked de la classe "FenetrePrincipale" (qui est appelée à chaque fois qu'un utilisateur clique avec la souris) crée une instance de la classe Pion qu'elle place à l'endroit où l'utilisateur a cliqué.

Fonctionnement global

-Classe Principal

-Fonction main : Cette fonction est la première fonction exécutée lorsqu'on lance le programme. Elle crée une instance de la classe FenetrePrincipale appelant ainsi son constructeur.

-Classe FenetrePrincipale :

-Fonction FenetrePrincipale : Le constructeur de la classe FenetrePrincipale définit les différentes propriétés de la fenêtre (la taille, la possibilité de fermer la fenêtre en cliquant sur la croix en haut à droite, le positionnement, l'impossibilité de redimensionner la fenêtre), puis crée une instance de la classe Quadrillage qu'il ajoute à la fenêtre. Il initialise ensuite le tableau qui stockera l'état de chaque case du plateau (les cases sont initialisées à 0 qui signifie "case vide").

-Fonction MouseClicked : Cette fonction est appelée à chaque fois qu'un utilisateur clique à l'aide de la souris. Elle commence par vérifier si la localisation du clique se situe bien sur le plateau de jeu et si la case est bien vide. Si c'est le cas, elle commence par répertorier l'état de la case dans le tableau prévu à cet effet puis crée une instance de la classe Pion qu'elle place aux bonnes coordonnées. On appelle ensuite la fonction CheckCase

-Fonction CheckCase : Cette fonction a pour rôle de vérifier les pions se trouvant dans chacune des 8 directions autour d'un pion pour calculer dans un premier temps la longueur des 4 lignes possibles autour du pion (Horizontale, verticale, et les 2 diagonales), et ensuite combien de pions ennemis se trouvent dans chaque direction du pion pour déterminer si il les a convertis à sa couleur. Cette fonction vérifie enfin si le joueur a gagné.

-Classe Quadrillage :

-Fonction Quadrillage : Ce constructeur se contente de charger l'image de croix dans la variable img.

-Fonction Paint : Cette fonction est appelée à la création d'une instance de la classe. Elle est constituée de deux boucles qui placent les différentes images de croix afin de constituer le plateau de jeu. On place ensuite les lettres au début de chaque colonne puis les chiffres au début de chaque ligne.

-Classe Pion :

-Fonction Pion : Ce constructeur charge l'image du pion en fonction du tour actuel (qui a été passé en paramètre du constructeur) puis stocke les coordonnées du positionnement du pion en tant que données membre.

-Fonction Paint : Cette fonction est appelée à la création d'une instance de la classe. Elle place l'image aux bonnes coordonnées.

Réalisation

Choix techniques

Nous avons choisi de séparer l'affichage graphique des calculs.

Ainsi, on dispose d'un tableau d'entiers de 21*21 (Les lignes et colonnes supplémentaires servent de zones de sécurité pour ne pas sortir du tableau).

Fonction CheckCase()

Nous avons choisi d'implanter cette fonction de la façon suivante :

On commence par vérifier si nous sommes au tour du joueur 1 ou 2 et on donne ainsi la valeur qui représente un pion allié ou un pion ennemi.

On initialise ensuite les compteurs qui définiront le nombre de pions alliés alignés pour chaque ligne possible (Horizontale, verticale, et les 2 diagonales).

On démarre une boucle for qui effectuera 8 itérations, représentant chacune des directions autour du pion venant d'être placé.

Un switch détermine les coordonnées qui correspondent à la direction en cours.

Si il y a un pion après le nouveau dans la direction actuelle, on recherche les pions qui suivent. D'abord, les pions alliés dont on stocke le nombre dans une variable puis de même pour les pions ennemis. Si il y a 2 ennemis suivis d'un pion allié, on modifie le tableau d'entiers pour convertir les pions ennemis, on rajoute graphiquement un pion allié à leur place et on incrémente le compteur de conversions.

On rajoute le nombre d'alliés trouvés à la ligne correspondante avec un switch.

Une fois la boucle for terminée, on vérifie si le joueur a atteint une des conditions de victoire (Une ligne supérieure à 5, ou plus de 5 paires ennemies converties).

Si oui, on déclenche la fonction qui appelle le pop up de victoire.

Format de fichiers

Les images créées avec Gimp sont au format PNG, pour la transparence qu'il soit adapté au projet.

Utilisation

Mode d'emploi

Lors du lancement du programme, on arrive directement dans le jeu.

Le joueur 1 (pions rouges) peut directement placer son premier pion.

La partie se finit lorsque l'un des joueurs à atteint l'une des conditions requises par l'apparition d'un pop up désignant le vainqueur.

Le jeu ne dispose malheureusement pas de menu.

Configuration requise

Le jeu fonctionne sous Windows et Linux cependant il y aurait un bug graphique sous Windows lors de l'apparition ou de la fermeture du pop up annonçant la victoire d'un joueur.

On ne sait pas a quoi celui ci est dû.

Conclusion

Bilan

Au final, le projet est abouti et respecte les contraintes énoncées dans le sujet. Toutefois, il aurait été possible, avec un peu plus de temps, d'y apporter quelques améliorations.

Le développement du projet peut être décomposé en plusieurs étapes correspondant aux principales difficultés rencontrées :

1. Détection des clics de la souris
2. Placement des pions sur le quadrillage
3. La fonction de vérification des fonctions autour du pion
4. Le changement graphiques des pions convertis

Optimisations possibles

Tracer des lignes avec les méthodes de Swing aurait peut être été plus élégant que d'utiliser une image que l'on répète 19*19 fois.

Peut être qu'au lieu des variables OffsetX et OffsetY nous aurions du utiliser un tableau stockant les offset en x et en y pour chaque direction.

Extensions possibles

Evidemment, un menu est la chose la plus flagrante qui manque actuellement au projet. Il aurait été bien de pouvoir réinitialiser la partie en cours ou d'en commencer une autre une fois la partie terminée.

Il y a aussi des points en plus que nous aurions aimés apporter comme l'ajout d'un mode à 4 joueurs dans les options par exemple.