Livrable du projet Java :

Les 6 couleurs

Groupe 4 :

Simao Pedro

Conte Julien

Sommaire :

Table des matières

I. Description du jeu

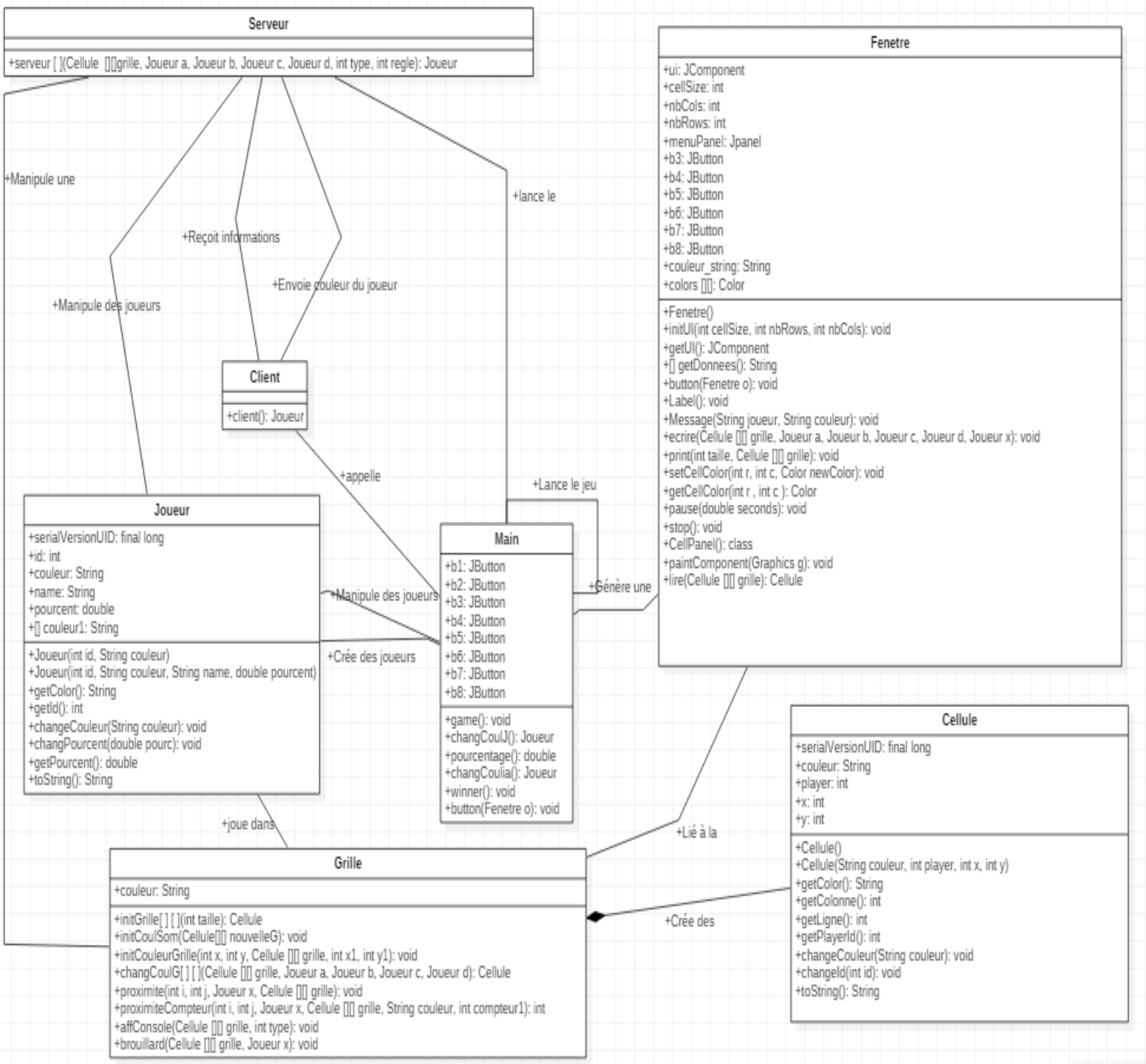
Le jeu consiste à occuper le plus de place possible sur le plateau. Le premier joueur à atteindre une occupation de 50% pour deux joueurs ou 25% pour quatre joueurs. Chaque tour, un joueur choisit une couleur. Toutes les cellules appartenant au joueur et celles à proximités sont changées dans la couleur choisie. Le joueur suivant ne pourra pas choisir la même couleur que l'autre joueur ou celle qui l'avait déjà.

En début de partie, chaque joueur se voit attribuer une couleur et une cellule dans un sommet de la grille sans qu'elle soit la même que celle de l'adversaire.

II. Fonction réalisée

- Version minimale
- Jeu en réseau
    - Deux joueurs sur deux ordinateurs différents (Jouer en réseau)
- Intelligence artificielle
    - IA qui choisit les couleurs aléatoirement (Difficulté 0)
    - IA qui choisit une couleur en fonction du nombre autour d'elle. (Difficulté 1, marche avec 4 joueurs et plusieurs IA du même type)
    - IA qui choisit une couleur pour ennuyer le joueur (Difficulté 2, ne fonctionne qu'avec 2 joueurs)
- Choix du nombre de joueur
    - Possibilité de jouer à 2, 3, 4 joueurs.
- Brouillard de guerre
    - Le joueur qui est en train de jouer ne voit que les cases qui lui appartiennent (Valable qu'en jouant en réseau)
- Enregistrement d'une partie
    - Deux boutons, pour sauvegarder et récupérer la dernière partie sauvegardée si on quitte le jeu.

III. Diagramme UML



IV. Description des algorithmes et des classes

Classe utilisée :

- Main (rediger fonctionnement)
- Grille : Cette classe sert à faire des opérations sur la grille comme changer les couleurs des cellules de celle-ci.
- Cellule : Cette classe comporte toutes les méthodes ce qui concerne l'objet cellule
- Joueur : Cette classe comporte toutes les méthodes qui concerne l'objet Joueur
- Serveur : Cette classe permet d'host le serveur si on est en mode multijoueur en ligne
- Client : Cette classe permet de se connecter au serveur et de jouer en ligne
- Fenêtre : Cette classe gère toute l'affichage graphique du jeu.

Pseudo code :

Pseudo-code changCoulIa :

On donne à la fonction le joueur a,b,c,d,x, le rang de l'ia (rang) et la grille (grille)

Initialisation des variables compteurs à 0 pour celles qui le nécessitent

Création de trois tableaux dont l'un vide (couleur1 de taille 6), un autre avec toutes les couleurs du jeu (couleur) et un tableau (comptCouleur de taille 6) qui comportera le nombre d'apparition de chaque couleur à côté des cellules du joueur donné à l'ia (donc soit l'ia soit le joueur à ennuyer)

Initialisation de tout le tableau couleur1 en null

Initialisation de tout le tableau comptCouleur à 0

Pour coulCompteur = 0, tant que coulCompteur < taille du tableau de couleur, coulCompteur ++

Si couleur[coulCompteur] != couleur du joueur a et de celle de b et celle de c et celle de d

couleur1[coulCompteur1] = couleur[coulCompteur]

coulCompteur1++

Si rang=0

compteur= nombre aléatoire compris en 0 et le nombre de joueur dans la partie

Sinon

Si rang=1

Initialisation de x1 et y

Pour i =0, i< taille de couleur1, i++

Pour x1=0, x1< taille de la grille, x1++

Pour y=0, y< taille de la grille, y++

comptCouleur [i]=comptCouleur [i]+

Grille.proximiteCouleur(x1, y, x, grille, couleur1 [i] ,0) (note1)

Sinon

Initialisation de x1 et y

Pour i =0, i< taille de couleur1, i++

Pour x1=0, x1< taille de la grille, x1++

Pour y=0, y< taille de la grille, y++

comptCouleur [i]=comptCouleur [i] +

Grille.proximiteCouleur(x1, y, a, grille, couleur1 [i] ,0) (Note1)

Note1 : La fonction proximiteCouleur compte le nombre d'apparition d'une couleur donnée autour d'une case. C'est une fonction récurrente qui continue tant qu'elle trouve des couleurs similaires à celle donnée.

Pseudo-code serveur :

On donne à la fonction grille, joueur a, b, c, d, regle

On définit un tableau qui nous permettra de stocker les joueurs (stockJoueur de taille 4)

un serveurSocket (ss) sur le port 9531

On affiche en attente de connexion

On définit le pourcentage pour gagner à 50 (on ne gère que 2 joueurs)

On crée un socket qui va écouter le port dans l'attente d'une connexion (s)

On affiche « le serveur a accepté la connexion de s »

On crée des variables qui vont écrire des objets (out) dans le buffer et lire (in) le buffer

On vide le buffer

On envoie la variable regle sur le buffer

On vide le buffer

Tant que les pourcentages du joueur a < pourcent et les pourcentages du joueurs b < pourcent

On vide le buffer

grille= Grille.changCoulG(grille, a ,b ,c d) (Note2)

Si regle=0

On fait l'affichage normal

Sinon

On fait un affichage brouillard (Note3)

On demande au joueur a de changer de couleur

grille= Grille.changCoulG (grille, a ,b ,c d) (Note2)

On envoie la grille

On vide le buffer

On envoie le joueur a

On vide le buffer

On envoie le joueur b

On vide le buffer

On envoie le joueur c

On vide le buffer

On envoie le joueur d

On vide le buffer

On affiche « En attente du joueur » + id du joueur b

On récupère le joueur b que nous a envoyé le client

On calcule les pourcentages de a et de b qu'on leur associe

On reset le buffer

On ferme in, out et le serveurSocket

On stock le joueur a, b, c, d dans le tableau stockJoueur

On retourne le tableau stockJoueur

Note 2 : On appelle la fonction qui change les couleurs de la grille en fonction des changements des couleurs de chaque joueur.

Note3 : On joue suivant les règles du brouillard de guerre.

V. Conclusion

Ce projet nous a permis d'appliquer et d'approfondir nos compétences en java et ainsi pouvoir saisir avec succès le cours d'algorithmique et programmation java.

Concernant ce qu'il manquait en termes de fonctionnalités du jeu, 3 jours seraient suffisants pour le mettre en place.

Lien sur git : https://github.com/Tutileni/Jeu_java_p