




Projet - IN204

21/03/2024

Pin JIN et Shikun WEI

Analyse d'usage du logiciel

- Objectif : Effacer le plus de lignes possible pour gagner des points et monter en niveau. Plus le niveau est élevé, plus les pièces tombent rapidement.

- Démarrage de la partie : Mode Solo ou Multijoueur
- Contrôles du jeu :
 - Mouvement horizontal
 - Rotation : Faire tourner la pièce.
 - Descente : À chaque tick, la pièce descend d'un cran.
- Collisions
- Lignes complètes : Les lignes sans espace vide sont supprimées
- Niveaux et scores

Analyse fonctionnelle

- Division en deux parties: **Logique du jeu** et **Affichage**




Logique du jeu:

- Exprimer des pièces
- Générer aléatoirement
- Contrôler le mouvement
- Supprimer la ligne
- Calculer le point

Affichage:

- Colorer des pièces
- Générer les pages
- Interagir avec des joueurs
- Contrôler le changement de page
- Montrer les résultats

Logique du jeu

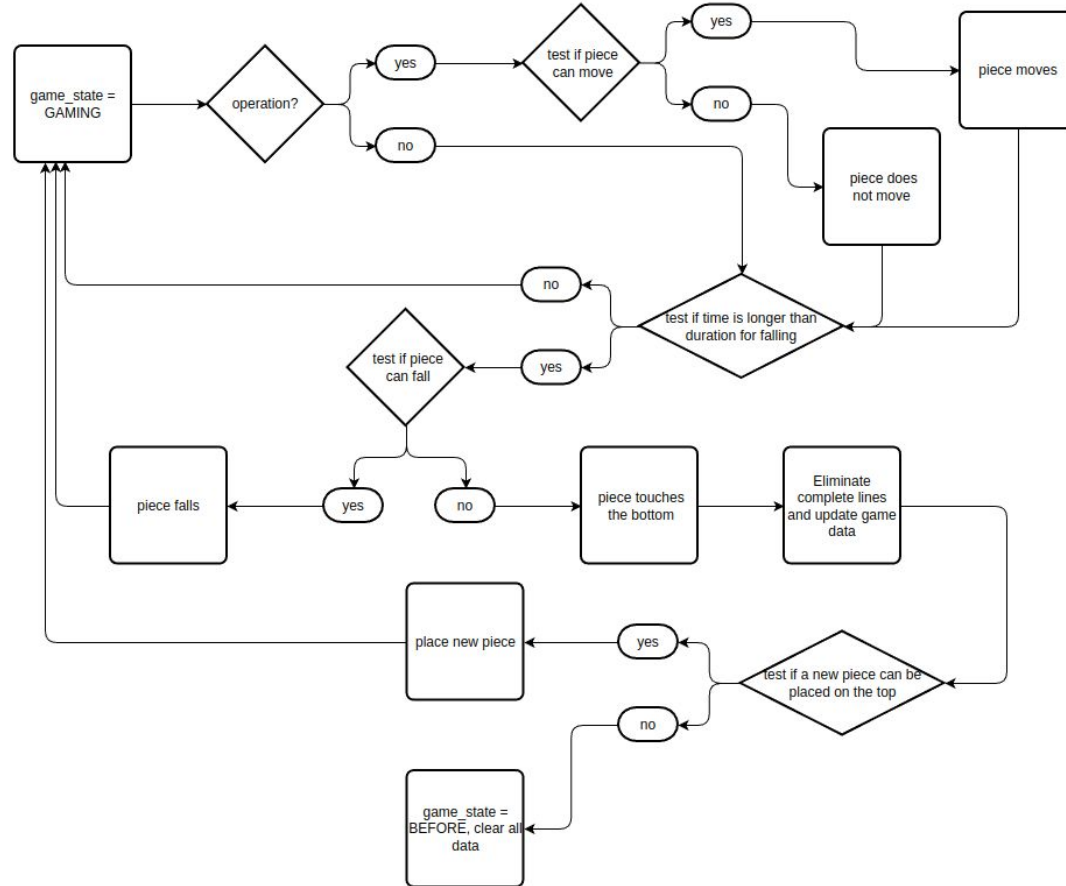
- 
1. **matrix.h**: Création d'une classe pour une matrice qui peut être tournée
 2. **piece.h**: Création d'une classe contenant l'orientation, la forme et la position de chaque pièce
 3. **game_logic.h**: Création d'une classe qui implémente la logique de base du jeu

```
C game_logic.h  
G main.cpp  
C matrix.h  
C piece.h
```


Logique du jeu



game_logic.h



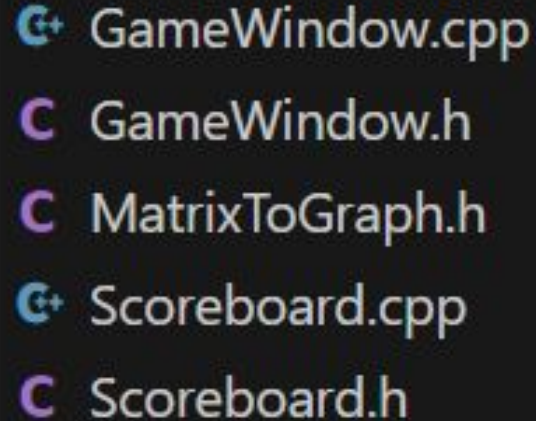
Architecture de l' affichage



MatrixToGraph.h: colorer les pièces selon leur type

Scoreboard.h: créer une classe pour la gestion des points dans le mode multijoueur

GameWindow.h: gérer la fenêtre et l'interaction avec les joueurs

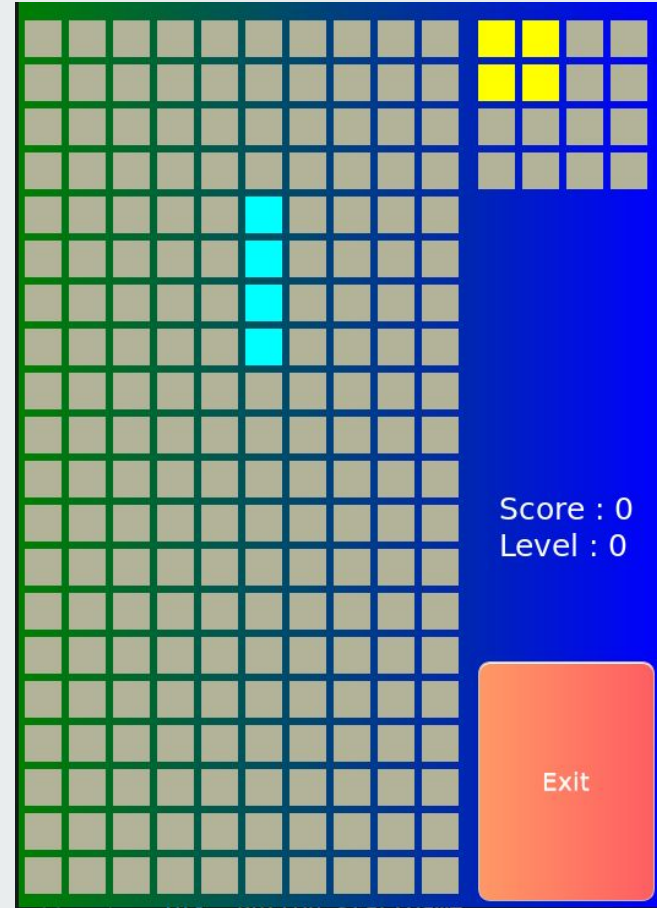


```
GameWindow.cpp
GameWindow.h
MatrixToGraph.h
Scoreboard.cpp
Scoreboard.h
```

Bibliothèque Utilisée: GTK

Outils:

- DrawingArea
- ColumnRecord
- Window, Button, Grid, Label
- Css pour l'embellissement



Description de l' affichage

Principe: changement de Page selon les états différents



Deux paramètres des états: `game_state` et `page_state`

	HOME	GAME	AFTER_GAME	MULTI	AFTER_MULTIT
BEFORE	Home Page			Update listes des joueurs	
GAMING		Update pièces, scores		Update pièces, scorebords	
GAME_OVER			Fin Page avec scores		Fin Page avec scoreboard

Mode Multi-joueurs

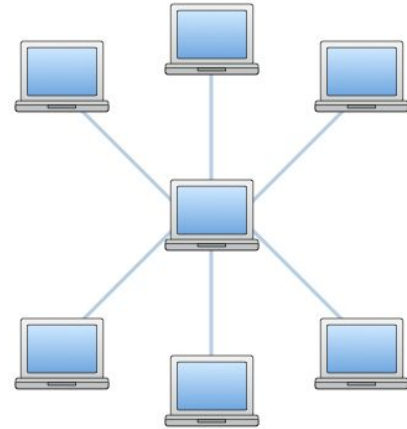
Protocole: TCP/IP

Librairie: `sys/socket.h`

Topologie: étoile

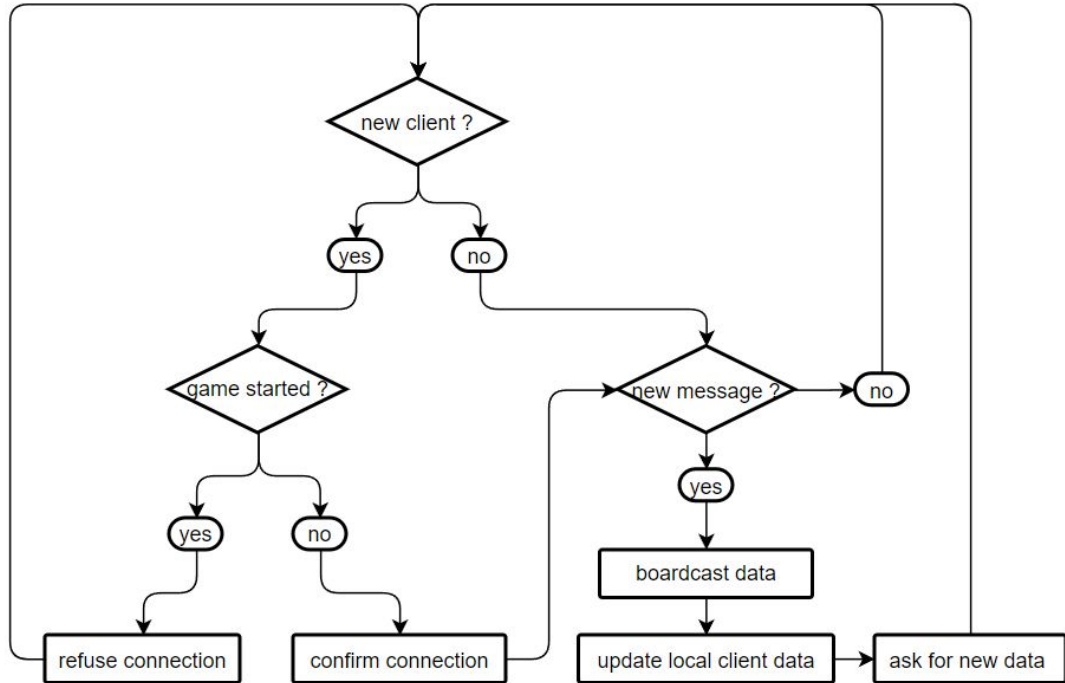
- Serveur: joueur qui crée le jeu
- Clients: les autres joueurs

Clients envoient leur data (score, level...) au serveur et serveur diffuse les informations



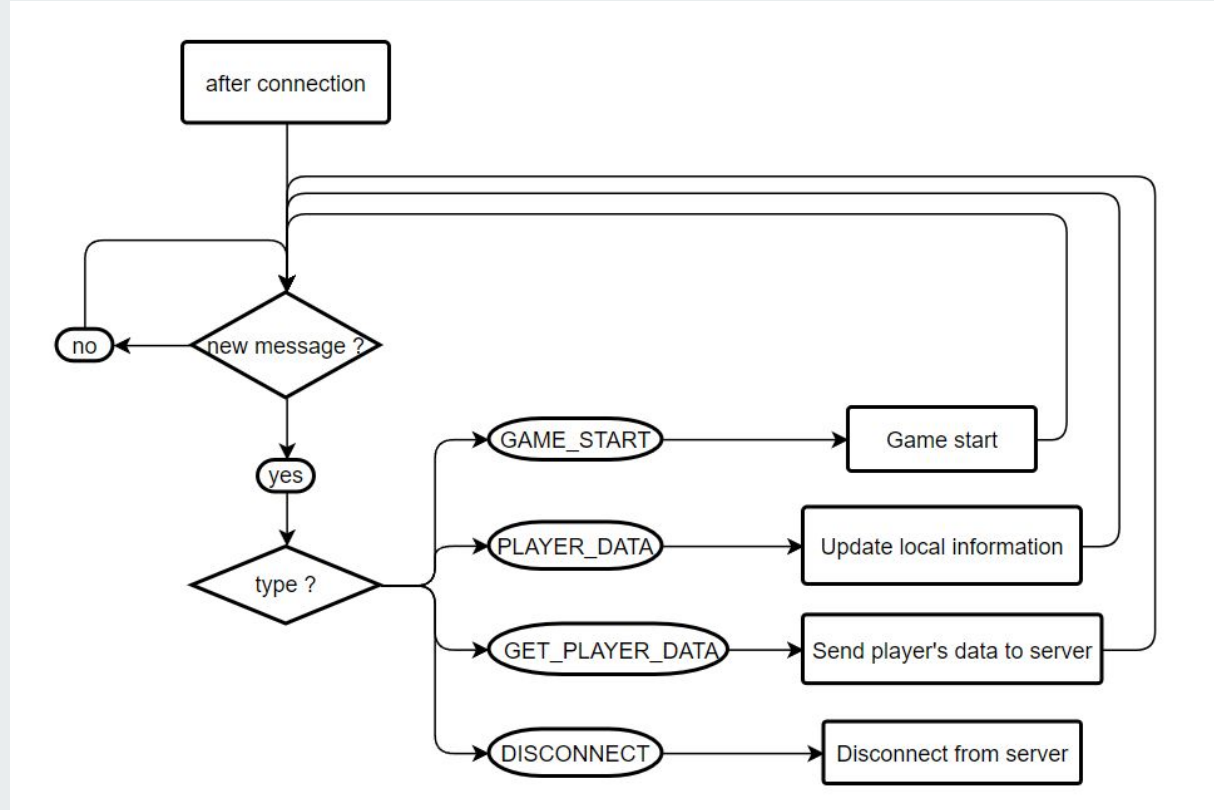
Communication entre joueurs

logique de server

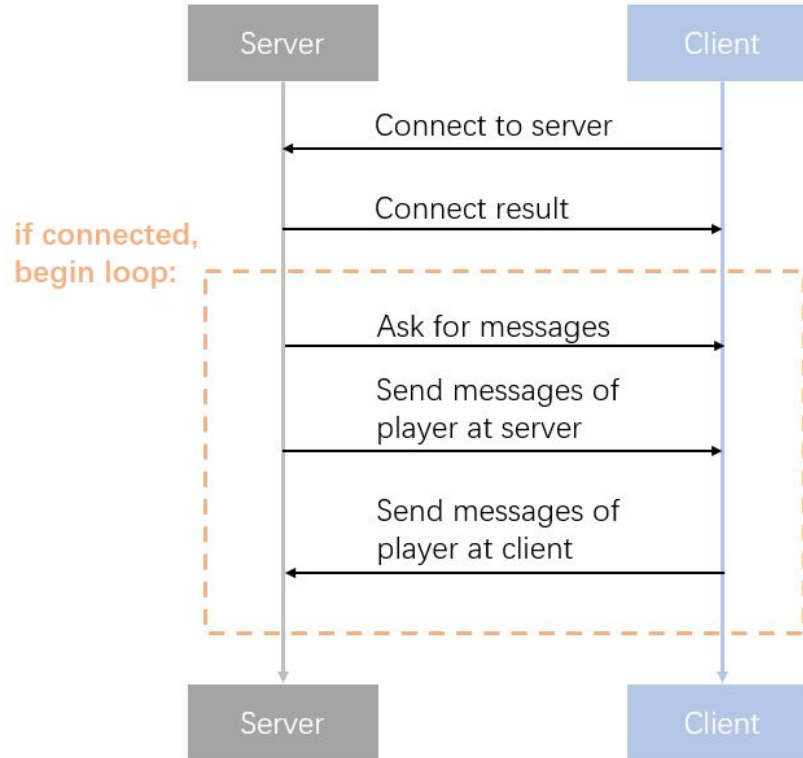


Communication entre joueurs


logique de client



Communication entre joueurs



Amélioration pour l'avenir

- 
1. Les joueurs doivent être dans le même sous-réseau LAN.
 2. Lorsqu'une partie multijoueur se termine et est réglée, il y aura un court délai dans la mise à jour des données des joueurs locaux sur la page de règlement.
 3. Une fois que le client s'est déconnecté du serveur, bien que le jeu puisse continuer, il n'y aura pas de page de règlement normale à la fin du jeu, elle clignotera et reviendra à l'interface HOME.