RAPPORT DU PROJET:Jeu d'assemblage de formes

réalisé par :

Sidiki KABA
Alassane BAMBA
Siega BISSOMBOLO
SAID AHMED Toihir Sitti Nourane

le 07 décembre 2020.

I-INTRODUCTION :

Après plusieurs séances de travaux pratiques avec notre professeur JOANNES Falade qui prêtait toute son attention en cas de besoin, il nous a été demandé de réaliser un Jeu d'assemblage de formes au cours des dernières séances de TP restantes.

Le but de ce devoir est de réaliser une application de jeu, dotée d'une interface graphique, qui consiste à assembler des formes de sorte qu'elles occupent le moins de place possible

Notre programme sera fonctionnel en ligne de commande mais aussi avec l'interface graphique.

Dans notre cas, L'idée s'inspire du fameux Tétris, mais les modalités sont différentes :
-toutes les pièces sont exposées sur l'aire de jeu dès le début de la partie,
-à tout moment,le joueur peut choisir une pièce en cliquant dessus,et alors la possibilité de la faire tourner ou de la déplacer,
-son but est de minimiser l'espace occupé par l'ensemble des pièces.Plus précisément,la fonction d'évaluation sera l'aire du plus petit rectangle(parallèle aux axes)recouvrant l'ensemble des pièces.

II-ORGANISATION DU PROJET

1) RÉPARTITIONS DES TÂCHES :

Le projet est essentiellement composé de deux (2) parties. D'une part le programme devrait être fonctionnel en ligne de commande et d'autre part avec le modèle MVC nous devons le faire marcher avec l'interface graphique. Bien avant de commencer, nous avons jugé bon d'élaborer un plan de travail afin de bien cerner les difficultés auxquelles nous serons confrontés tout au long du projet.

Afin de pouvoir finir dans le temps, chacun d'entre nous a pris une partie de notre projet, étant au nombre de quatre personnes, KABA Sidiki et BISSOMBOLO Siega se sont occupés essentiellement de la partie modèle c'est à dire les classes (Createur,CreateurPieces,InterfacePieces,Observable,Observateur,Pieces,PieceL,PieceT,PieceU,PlateauJeu,Position).

SAID AHMED Toihir Sitti Nourane et BAMBA Alassane de la vue et du contrôleur c'est à dire les classes(EspaceJeu,PanelMenu,VueJeu,TetrisJeu).

Il faut toute fois signaler que Toihir et Bamba nous ont aussi accordé un peu de leur temps dans la réalisation du modèle vue que c'était la partie la plus difficile à réaliser.

2) ARCHITECTURE DU PROGRAMME :

Concernant l'architecture du programme nous avons organisé le projet de la façon suivante:

src : est le répertoire contenant,le package tetris_mvc qui contient toutes les classes pour faire jouer le jeu comme il le faut.

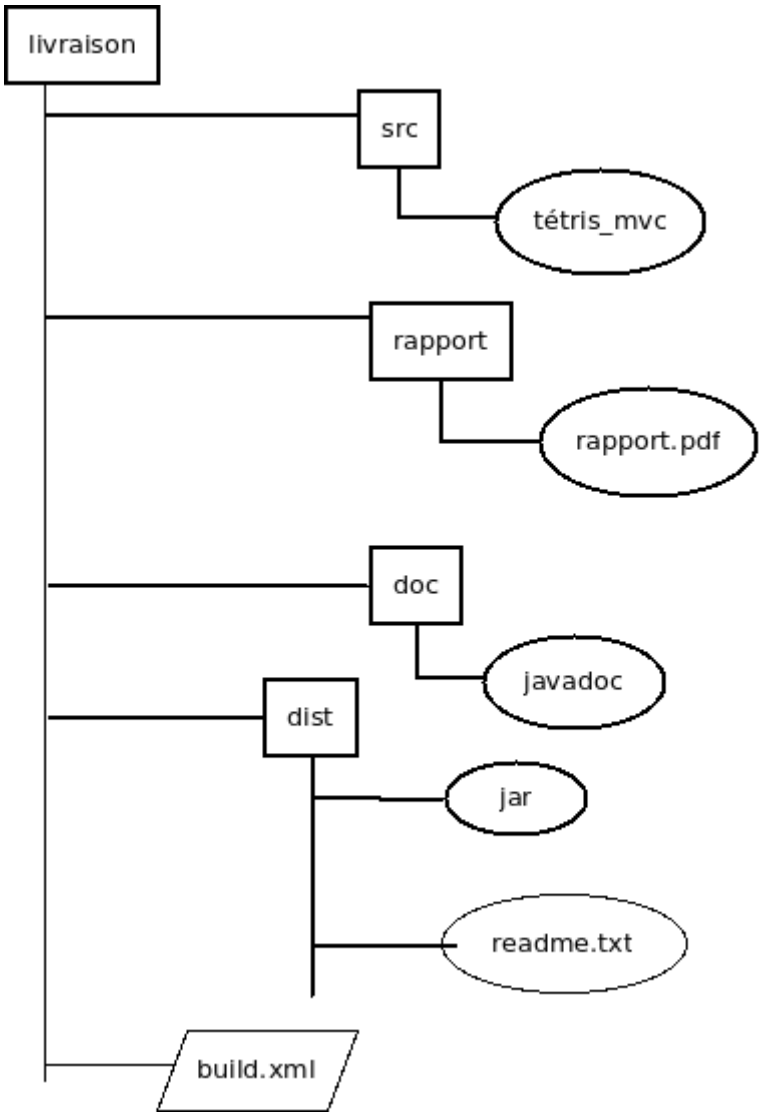
doc :est le répertoire contenant la Javadoc du programme.

dist : est le répertoire contenant le fichier jar du programme.et aussi le fichier readme.txt

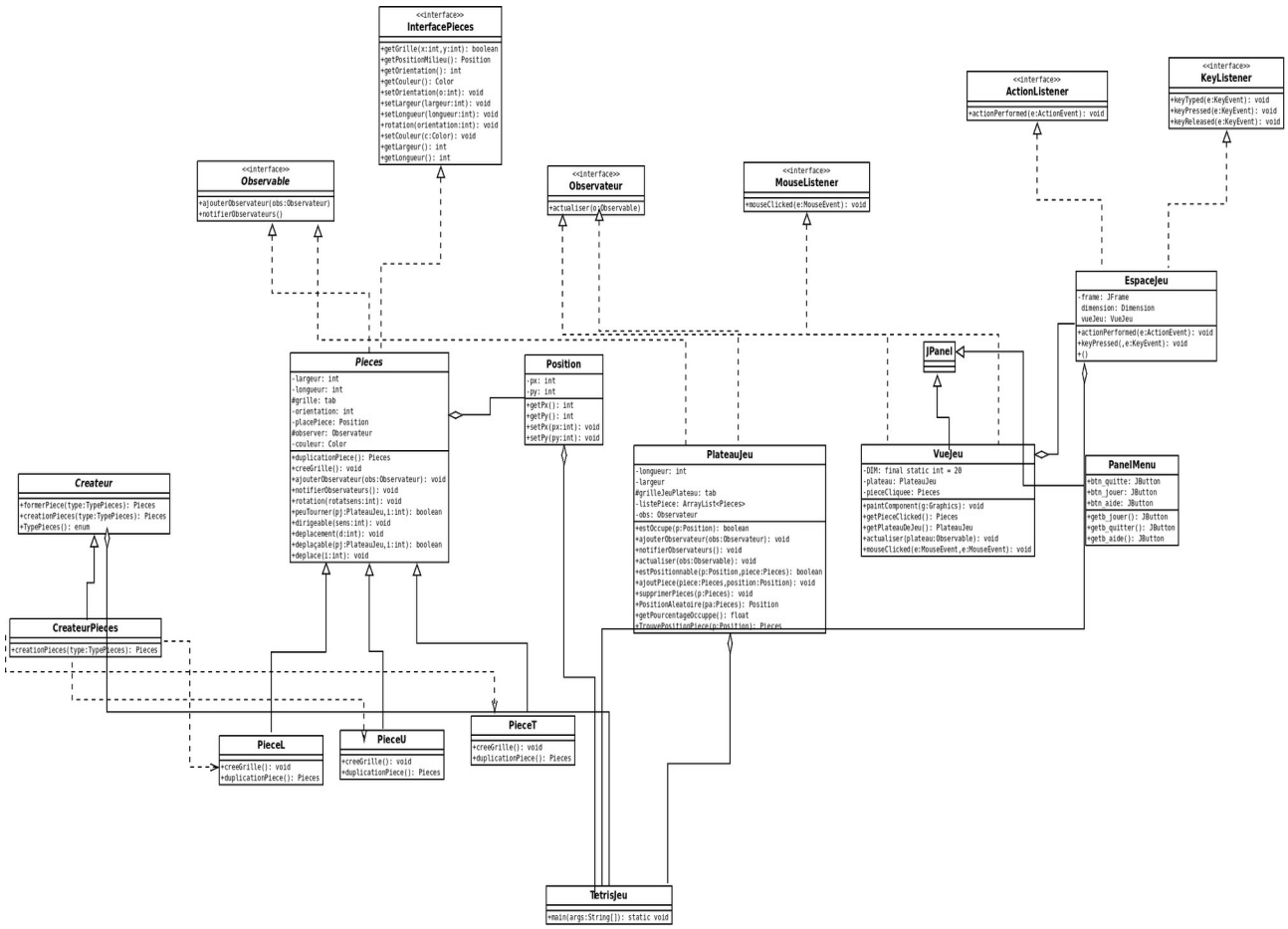
rapport :est le répertoire contenant le mini-rapport du projet.

build.xml :est le fichier permettant de ré-générer le contenu de dist via ANT.(n’a pas pu être générer)

Schéma de l’architecture du programme :



3)Diagramme de classe du projet



III-ÉLÉMENTS TECHNIQUES

Dans la partie modèle :

La méthode formerPiece(TypePiece type)de la classe **Createur**: Cette méthode intègre le design pattern "Fabrique" car la formation d'une pièce est déléguée a la sous-classe.
En effet, on remarque que la méthode "formerPiece" appel une autre méthode "creationPieces" qui retourne uniquement le type de pièce souhaité.
Ensuite , avec ce type, on fait appel **PieceSouhaite.creeGrille**() qui crée la grille de la pièce correspondante.

Grâce au design pattern Fabrique l'ajout d'une nouvelle Piece dans notre jeu ne nécessite pas de modifier le code existant (exceptée la classe principale Main). En effet, il suffit de créer les classes correspondantes aux Type de pièces souhaités (qui hériteront de la classe **Pieces**) puis Il suffira d'ajouter le nouveau type dans l'énumération de cette classe puis d'ajouter ce type dans la classe **CreateurPiece** avec la nouvelle classe correspondante.
Cela permet d'être robuste dans notre création de Pièce.

La méthode deplace(int i)de laclasse **Pieces**:Cette méthode permet de déplacer une pièce.En effet, elle utilise la méthode **déplaçable**(PlateauDeJeu p,int i) pour voir si la pièce peut être déplacer avant d'utiliser la méthode **deplacement**(int i) pour faire la translation via le changement de coordonnée et après elle notifie ses observateurs du changement.

La méthode dirigeable(int sens) de la classe **Pieces**:Cette méthode permet de faire la rotation d'une pièce.En effet, elle utilise la méthode **peuTourner**(PlateauDeJeu p, int sens) pour voir s'il est possible d'effectuer une rotation de la pièce avant d'utiliser la méthode **rotation**(int sens) pour effectuer la rotation et après elle notifie ses observateurs du changement.

La méthode estPositionnable(Position p, Piece piece) de la classe **PlateauJeu**: méthode confirmant s'il est possible de mettre une pièce a une position donné de la grille de jeu.En effet, on vérifie si la pièce ne déborde pas du cadre de jeu aussi s'il est possible de d'ajouter cette pièce a cette position.
La méthode getPourcentageOccuppe() de la classe **PlateauJeu**:cette méthode permet de calculer le pourcentage occupé par l'ensemble des pièces sur la grille.En effet,plus on occupe moins de place plus le pourcentage est petit.

IV-CONCLUSION

En conclusion, la réalisation du projet nous a demandé pas mal d'effort afin de pouvoir y parvenir. Certes, nous n'avons pas pu réaliser tout ce qui nous a été demandé mais, notre jeu est bel et bien fonctionnel en interface graphique.

Par ailleurs, nous ne sommes pas totalement satisfaire de notre projet vu qu'il est susceptible de beaucoup d'améliorations possibles. Malgré, notre insatisfaction nous n'avons pas pu faire mieux avant la date limite vu qu'il n'y avait pas mal de devoirs de l'école à rendre dans presque les mêmes délais Cependant, nous avons décidé ensemble de poursuivre la réalisation du projet même après l'avoir rendu.

Cela nous permettra d'améliorer le jeu et de découvrir en profondeur la programmation avec le langage Java.

V-REFERENCES

https://design-patterns.fr/fabrique-en-java
https://design-patterns.fr/observateur
http://zetcode.com/javagames/tetris/thebibliography