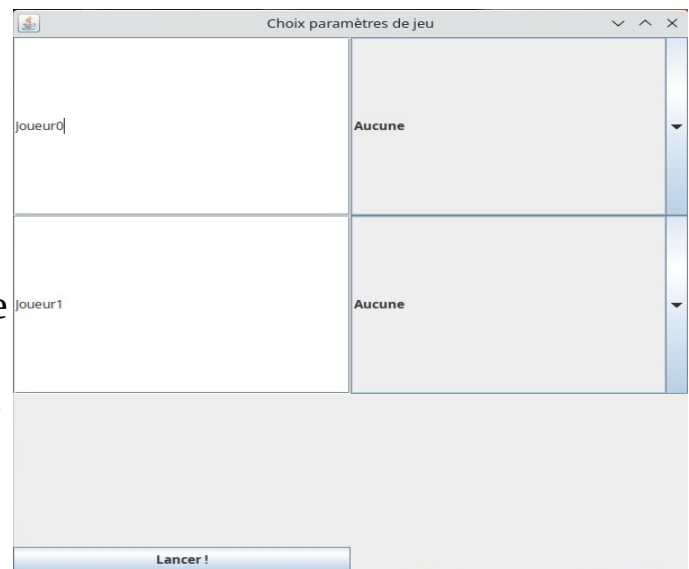
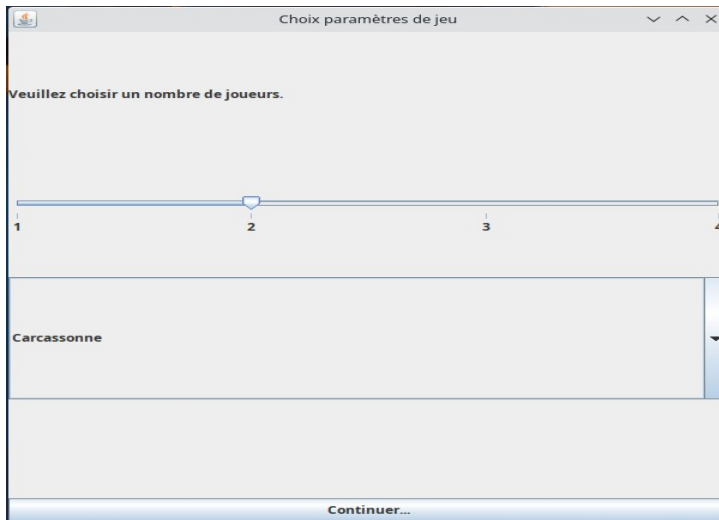


Les parties du cahier des charges qui ont été traitées :

Lorsque l'on démarre le fichier VueLanceur.java une nouvelle fenêtre s'ouvre pour nous demander le nombre de joueurs ainsi que le mode de jeu que nous voulons lancer. Il est possible de choisir Carcassonne, Dominos ou bien Dominos avec l'affichage dans le terminal.

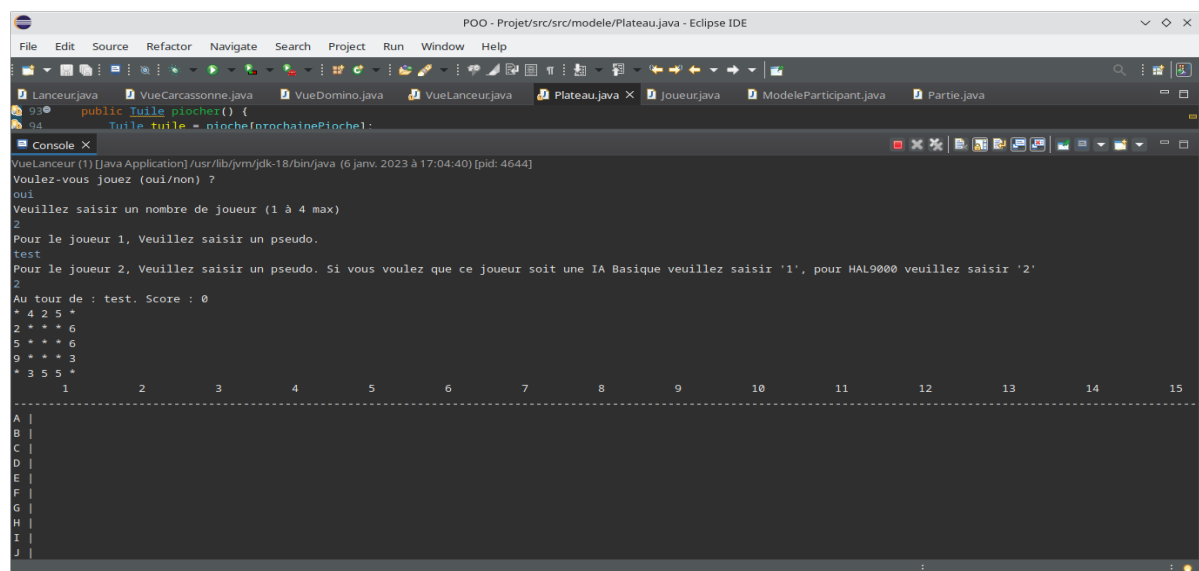
Après avoir choisis nos paramètres, nous allons pouvoir choisir le pseudo de

chaque joueur. Par défaut, le pseudo est « JoueurI » avec I un chiffre entre 0 et 3. Pour chaque joueur, sauf le premier, il est possible de choisir si le joueur est une IA.



Une fois tous les paramètres remplis le mode de jeu se lance (sauf pour les dominos avec l'affichage terminal qui se lance directement après la première fenêtre).

Pour les dominos dans le terminal, nous avons dans un premier temps les questions pour le paramétrage du jeu. Ensuite, la partie commence. La partie s'achève lorsqu'il n'y a plus de tuiles dans la pioche.



Pour chaque mode de jeu :

Il est possible d'abandonner en cours de partie. Si un joueur abandonne alors nous le retirons de la liste des participants et nous vérifions qu'il reste encore au moins un humains parmi les participants, sinon la partie s'arrête.

Il est aussi possible de tourner une tuile autant de fois qu'on le souhaite ou encore de la defausser. Si cette tuile est defaussee elle est « jetée » c'est-à-dire qu'elle n'est pas réintégrée dans la pioche.

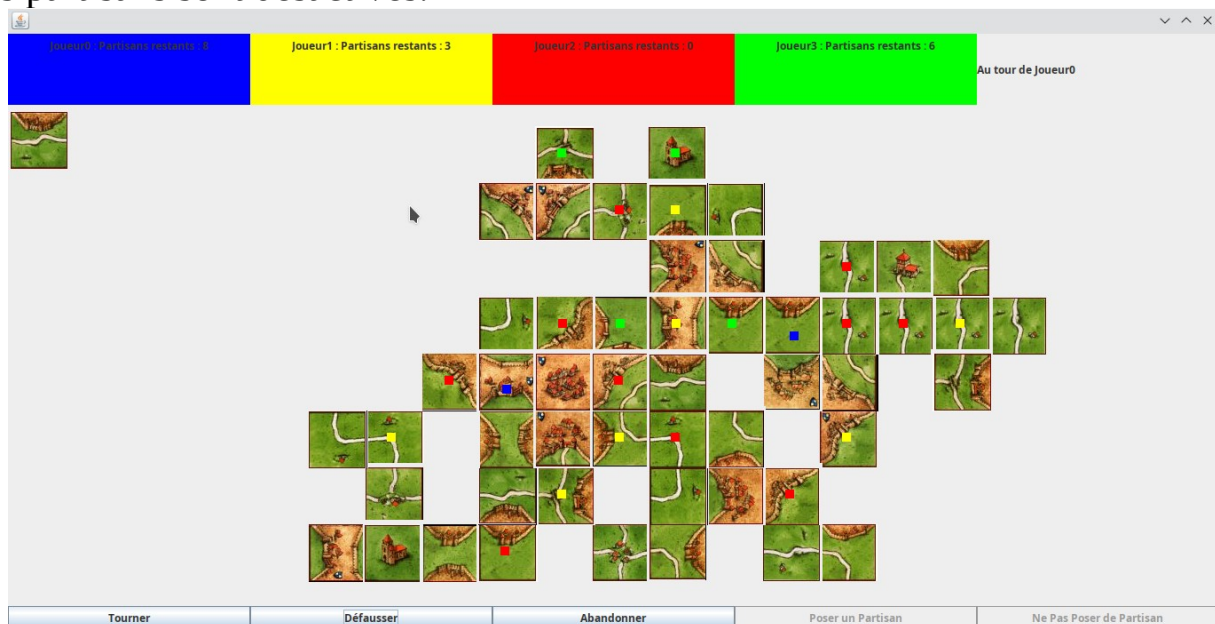
Il est toujours notifié à qui est le tour ainsi que la carte piochée. Il est possible de jouer contre IA.

Pour Carcassonne :

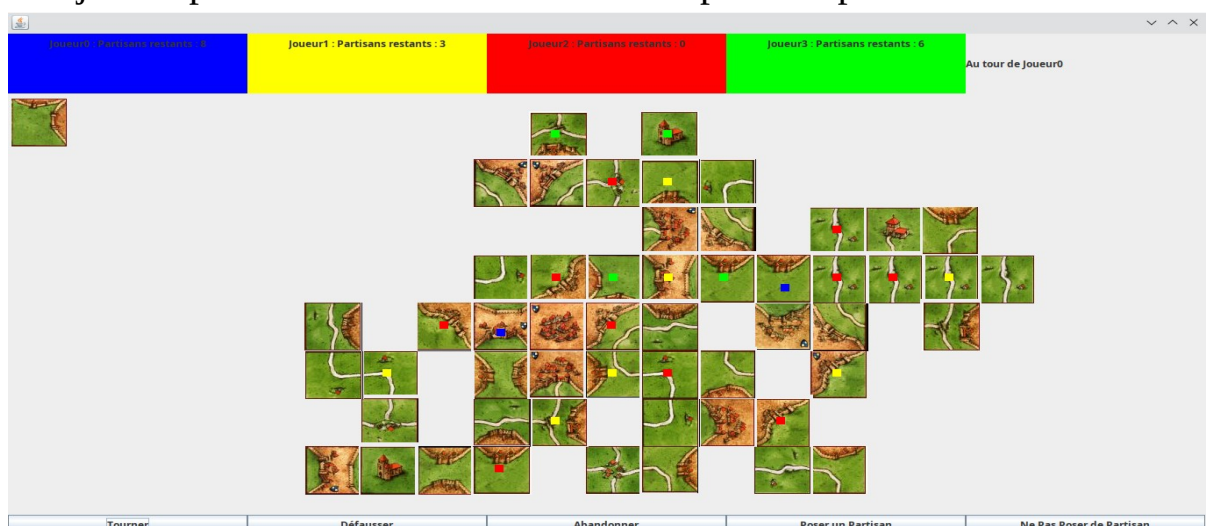
Chaque joueur a une couleur qui lui assignée. Cette couleur correspond à la couleur des partisans du joueur en question. Par exemple :

le premier joueur a comme couleur bleu alors ses partisans sont bleus. Il y a en plus deux boutons : « Poser un Partisan » et « Ne Pas Poser de Partisan ».

Chaque joueur à 10 partisans. Avant de poser une tuile les boutons concernant les partisans sont désactivés.



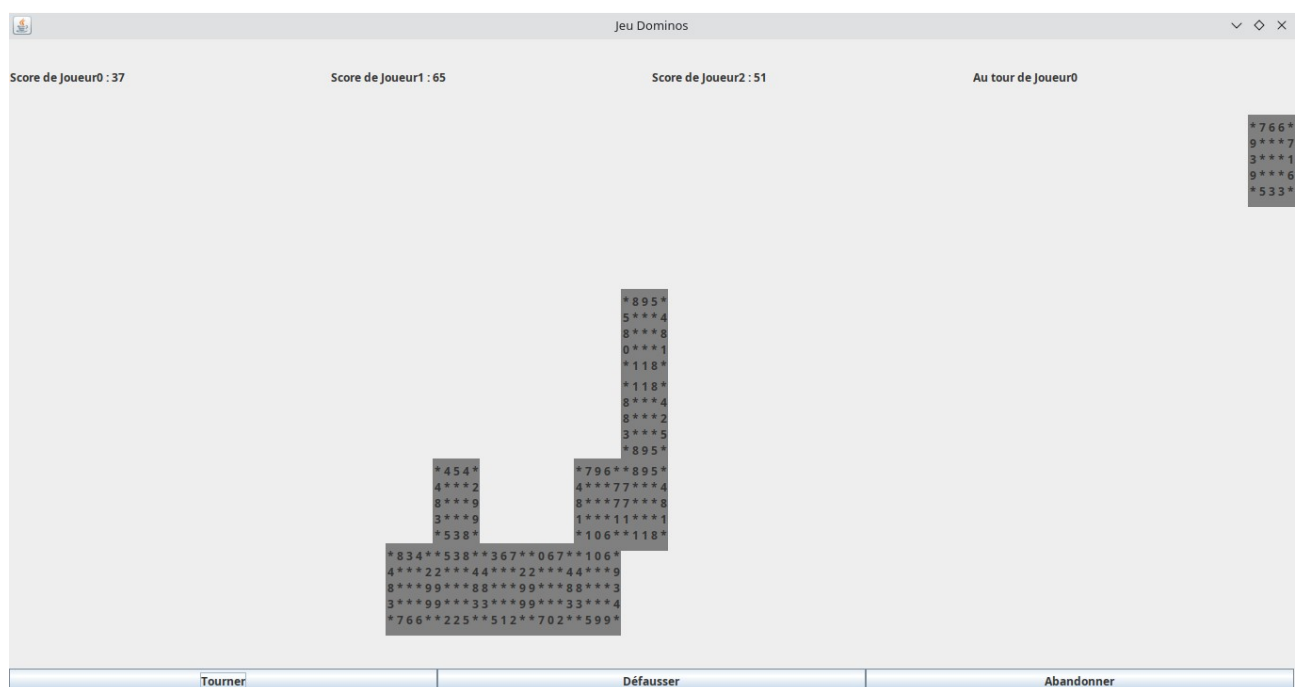
Lorsqu'un joueur pose une tuile les deux boutons pour les partisans s'activent.



Si le joueur choisie de poser un partisan alors il doit, sur la tuile qu'il vient de poser, cliquer à l'endroit où il veut poser un partisan, son nombre de partisans diminue alors de un et ensuite le tour passe au prochain joueur. Un partisan est représenté par un petit carré de la même couleur que celle du joueur. Si il choisie de ne pas en poser le tour passe directement au prochain joueur. De plus, il est possible de jouer contre IA basique. Cette IA parcours aléatoirement le plateau contenant les tuiles et si elle trouve une place alors elle pose la tuile sinon elle la défaisse. L'IA peut aussi poser un partisan de façon aléatoire.

Pour Dominos :

En haut de l'écran est notifié le score de chaque joueur qui change à chaque fois qu'une tuile est posée.



Il est possible de jouer contre une IA Basique (la même que pour Carcassonne) mais aussi contre une IA améliorée (HAL9000). HAL9000 va parcourir tout le plateau et tester chaque possibilité pour ne retenir que celle qui rapporte le plus de points. A la fin de la partie un message affiche le pseudo du gagnant ainsi que son score.

En résumé, l'ensemble du cahier des charges minimal a été rempli ainsi que la fonctionnalité supplémentaire HAL9000 pour le jeu domino.

Les problemes connus :

Il y a eu plusieurs difficultés concernant l'interface graphique.

Tout d'abord il était difficile de mettre en place la methode `mouseClicked` de l'interface `MouseListener`. En effet, trouver comment lors d'un clique, détecter la ou les tuiles qui se trouvent à proximité pour vérifier si la tuile piochée est posable à l'endroit du clique était compliqué.

Ensuite, il y a eu des problèmes pour l'IA. En effet, une fois que l'IA a trouvé dans le plateau une place pour la tuile, comment la poser et la faire apparaître dans l'interface graphique au bon endroit et dans le bon sens ? Par exemple : pour trouver un endroit, l'IA va parcourir le tableau d'une certaine façon mais va aussi tourner plusieurs fois la tuile. Si l'IA trouve une place il se pouvait que la tuile ne soit pas dans le bon sens dans l'interface graphique. Pour régler ce bug il a fallu gérer le nombre de fois qu'une tuile est tourner par l'IA.

Par ailleurs, il y a eu des problèmes pour la pose des partisans. La premiere était pour ne permettre la pose d'un seul partisan seulement sur la dernière tuile posée. La deuxième était pour rafraichir l'interface graphique. Par exemple : si nous posons un partisan alors celui ci ne s'affichait pas tant que la taille de la fenêtre n'avait pas été modifiée.

Pistes d'extensions :

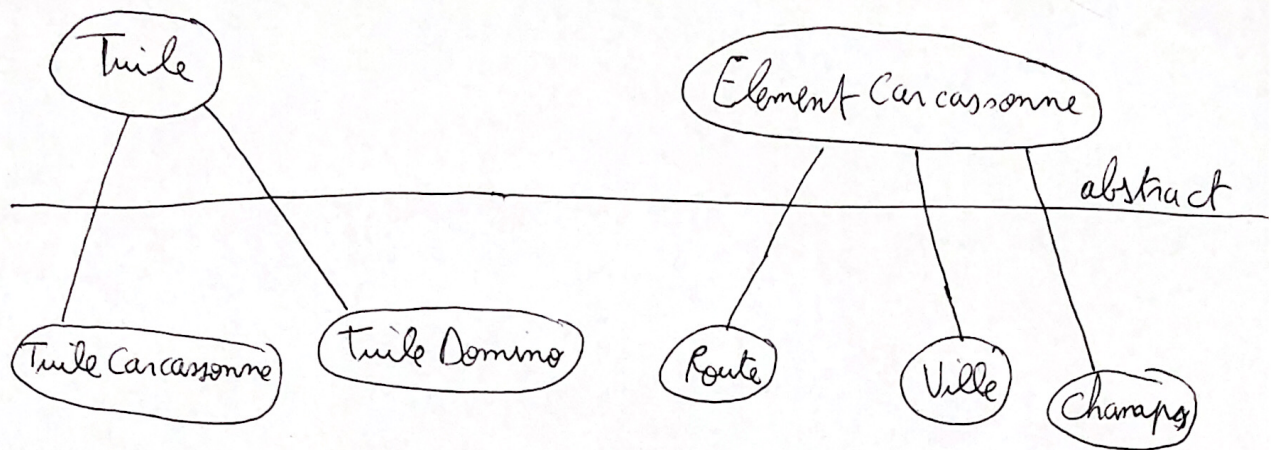
Carcassonne 100 % :

L'idée est de localiser précisément le clique pour savoir à quelle case du tableau de la tuile il correspond. On rajoute un boolean `partisanPresent` en attribut pour chaque objet `ElementCarcassonne`. Pour pouvoir poser un partisan il faut que `partisanPresent` soit à `false` sinon la pose est impossible. Lorsque l'on pose une tuile, on vérifie si les objets `ElementCarcassonne` des deux tuiles en contact on `partisanPresent` à `true`. Si oui alors le champ `partisanPresent` des objets en questions qui viennent d'être posés passe lui aussi à `true`.

Pour le score on explore récursivement en utilisant `partisanPresent` et on compte les points selon les règles du jeu.

HAL9000 pour Carcassonne :

Une fois le score de carcassonne implémenté, on utilise la même méthode HAL9000 que pour les dominos.



Joueur

Modele Participant

Orientation Tuile

Plateau

Lanceur

Partie

Vue Carcassonne

Vue Domino

Vue Lanceur

Package src/

modele/

Champ.java
Route.java
Ville.java
ElementCarcassonne.java
Joueur.java
ModeleParticipant.java
OrientationTuile.java
Plateau.java
Tuile.java
TuileDomino.java
TuileCarcassonne.java

vue/

Lanceur.java
Partie.java
VueLanceur.java
VueCarcassonne.java
VueDomino.java