

Yu Yang Huang Herve

Kante Cherif

Compte Rendu

Sommaire :

-Schématisation / Communication

-Le Front End

-Les difficultés

Schématisation / Communication

Au tout début du projet on a chacun de notre côté analysé l'énoncé, puis imaginé et écrit pour règle ou contrainte une classe, interface ou classe abstraite ainsi que ses fonctions ou plutôt un nom de fonction qui lui serait associé. Cela nous a permis d'avoir un petit schéma des étapes nécessaire à la programmation de ces quatre jeux.

```
MethodeRegle.txt - Bloc-notes
Fichier Edition Format Affichage ?

Règle des jeux sous forme de méthode (à modeliser avec...)
Catégorie: || Nom du jeu + methode associer || Domaine d'action de la methode agit

Règle du domino :
Initialement : //Je sais pas si on met cette interface dans commencement bref ;
-protected int Limite(); //Sur le nbr de domino

Joueur : // presque sur qu'il y aura une interface joueur;
-protected void main(); //Joueur afin de pouvoir l'essayer dans un tes
-protected String toString() //Meme chose
-

Pour commencer : // Interface commencement
Plusieurs possibilités :
-protected Joueur plusJeune(); // sur la liste des joueur, ou autre on verra
-protected Joueur doubleLePlusGrand(); // Pareil sur la liste de joueur;
-protected Joueur PiocheEleve(); // Meme chose ;

Domino ://Ce sera a priori une class
Argument(s): (Private ou public on s'en fou pour l'instant)
-int partieGauche;
-int partieDroite;
```

```
Projet.txt - Bloc-notes
Fichier Edition Format Affichage ?

//////////////////////////Pour Domino Général//////////////////////////

Il y à 2 cas:

Soit on utilise les variable avec le type var pour le haut et le bas du domino com
ça domino n'aura pas besoin d'être abstraite et tout pourra se faire dans la cla
Domino. Pour afficher les domino par exemple que cela soit un dessin, un chiffre
une couleur on pourra tout mettre dans les attribut var( comme en python).

Soit on la met abstraite mais il faudra donc des classe en plus pour les 2 type d
jeu de domino.
Utiliser une fonction qui renverra un string pour le haut et bas du domino.

//////////////////////////Pour la classe main//////////////////////////

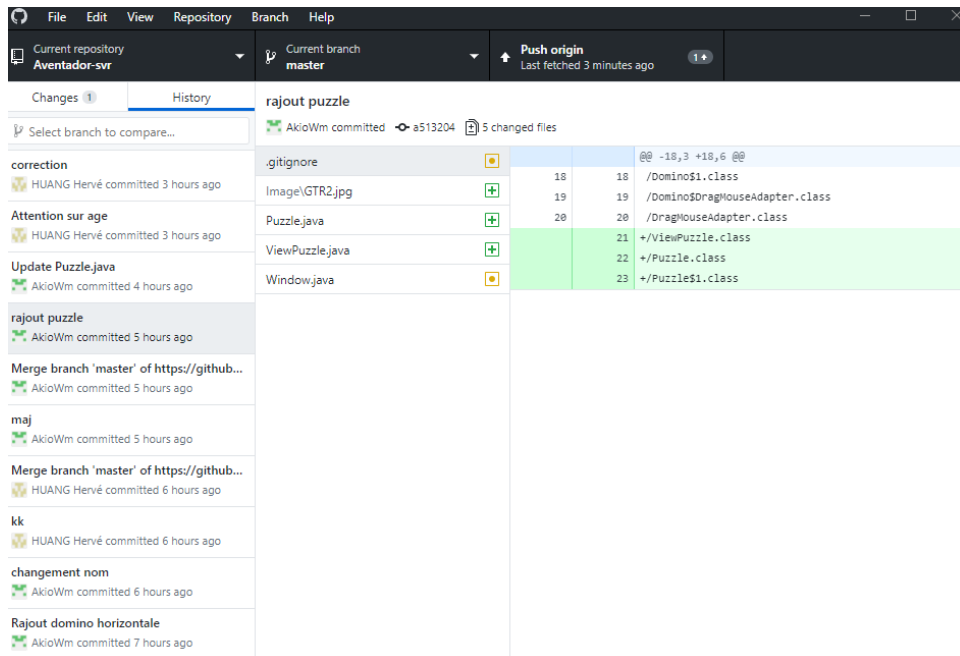
on va remplir la main avec une autre classe(la classe pioche).

il y aura une fonction qui va afficher les dominos que le joueur à en main.

//////////////////////////Pour la classe Pioche//////////////////////////
```

En effet, lorsque nous avons commencé la partie programmation, nous avons tout d'abord réécrit chaque classe et fonction du schéma. De plus, créant un schéma et en regroupant les classe et fonction auparavant nous avons remarqué que certaines fonctions ou classes étaient redondantes donc nous avons pu les identifier et les placer dans des interface ou classe abstraite lorsque nous le remarquions.

Afin de permettre au projet de devenir un vrai projet de groupe et non un travail où chaque binôme fait sa partie dans son coin. Nous avons donc sollicité l'utilisation de GitHub et de GitHub Desktop.



Nous avons, tout d'abord créé chaque classe et fonction que nous avons schématisée, en laissant vide le corps des fonctions. Par la suite, nous les avons remplies.

La partie Graphique :

Les fenêtres :

Pour la cette partie il fallait tout d'abord créer une fenêtre de base. Comme les jeux sont tous différent chaque jeu devait avoir sa propre fenêtre pour ou au moins son propre panel de jeu. Nous avons par conséquent créer une première fenêtre pour domino.

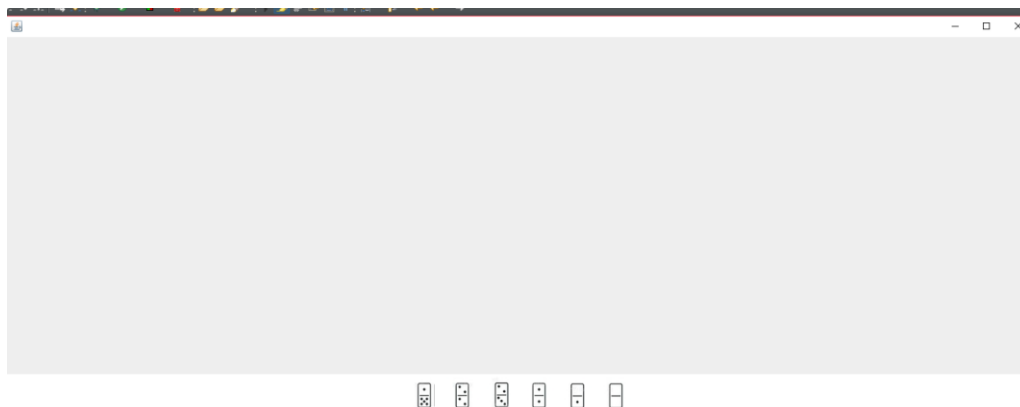
Le Panel du Joueur :

Le panel du joueur était constitué de pièces (domino,puzzle etc.). Au début nous avons créé le panel avec des dimension égal à la fenêtre pour sa longueur et pour sa hauteur nous avons pris la hauteur de la fenêtre divisée par quatre.

Les Dominos :

Pour les dominos nous avons pris une image sur google image puis nous avons rogné les différents dominos. La difficulté rencontrée était la taille des images, il fallait toutes les redimensionnée car elles étaient trop grandes pour la taille du panel des mains.

Pour le moment la fenêtre ressemblait à l'image ci-dessous :

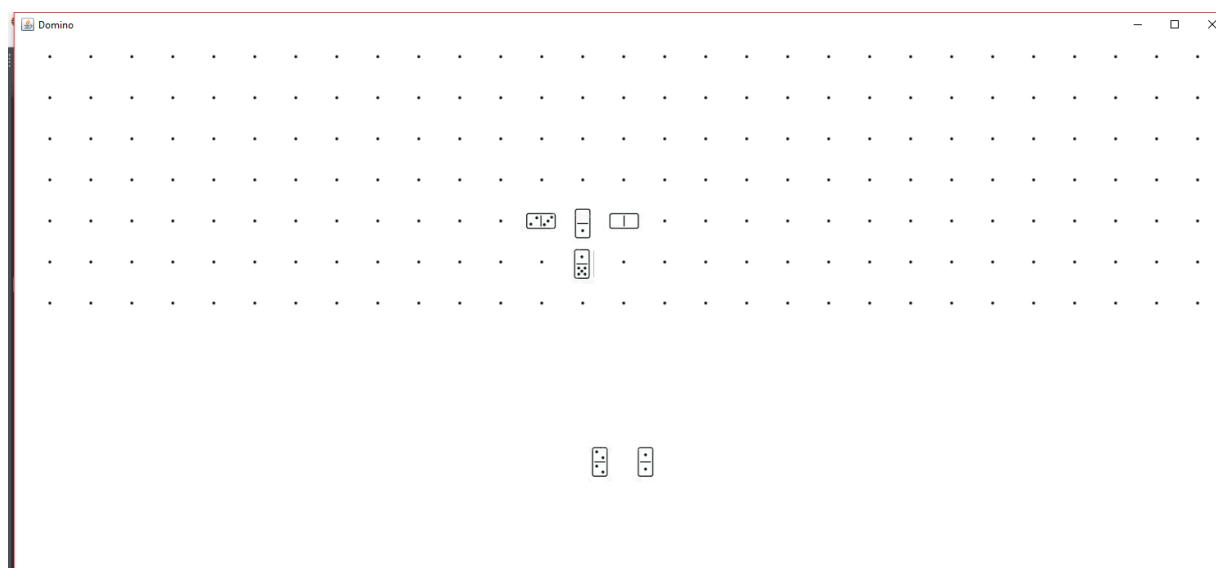


Les Panels du jeu :

Le panel de jeu est propre à tous les jeux du projet. Pour le créer il y a eu pas mal de difficulté :

- La zone de jeu : Il fallait créer tout d'abord la zone de jeu. Pour le quadrillage nous avons pris des JLabel avec une image de point plutôt que de faire un quadrillage pour gérer les déplacements plus facilement.
- Le déplacement : Il fallait gérer le déplacement des pièces sur la zone de jeu. La plus grande difficulté était donc de savoir comment mettre en place ce déplacement. Il nous a fallu faire pas mal de recherche pour trouver la solution adéquate et peut-être compliquer pour avoir un déplacement simple pour notre jeu. Pour cela nous avons utilisé la classe TransferHandler. L'idée de base était de déplacer entièrement l'objet Domino qui était un JPanel mais cela fut trop compliquée. En effet pour déplacer des Container il fallait créer notre propre classe TranferHandler héritant de la classe mère du même nom. La solution la plus simple était de tout simplement déplacer des ImageIcon plutôt que des JPanel ou JLabel. En effet cela était beaucoup plus simple à mettre en place et comme la zone de jeu est composée de JLabel avec des ImageIcon représentant un point nous avons juste à "activer" le déplacement des images des dominos. Cela est aussi le cas pour les images du puzzle. Pour pouvoir utiliser la souris nous avons utilisé un dragMouseListener. Il nous suffisait juste de cliquer sur l'image d'un domino pour le placer sur la zone de jeu. En plaçant cette image sur la zone de jeu cette dernière remplace l'image qui se trouvait sur le JLabel cible à savoir un point. Cela nous a permis donc de mettre en place un système de "Drag'n Drop".

Voici l'image de la fenêtre finale ci-dessous :



Les Difficultés

Les plus grandes difficultés rencontrées ont été la partie graphique ainsi que la connexion des fonctions entre le front end et le back end. En effet, il nous manquait encore les liaisons entre les différentes fonctions des jeux avec nos JPanels afin de les rendre fonctionnels. Nous avons mis beaucoup de temps à faire la partie graphique ce qui nous a donné pas mal de retard sur pour faire la suite des jeux.

En effet j'ai (Cherif) pris pas mal de temps pour mettre en œuvre le "Drag'n Drop" des objets car je ne comprenais pas comment cela fonctionnait. Pendant ce temps mon binôme (Herve) a continué sur la partie back end du projet. Au final lors de la mise en commun de nos codes via notamment la plateforme GitHub nous eûmes pas mal de conflit à gérer et cela nous a pris également beaucoup de temps.

Il y a eu également une contrainte de temps en vue des nombreux cours et TP à faire dans les différentes matières.