

```
import org.webmacro.se  
import org.webmacro.se  
import org.webmacro.util.  
import java.io.*;  
import java.sql.*;  
import java.util.*;  
javax.servlet
```

# Projet Java

Jeu des 6 couleurs

TANTAOUI EL ARAKI Selim

THIBON Manon

# Table des matières

1. Rappel du sujet .....	3
2. Modélisation (description des classes) .....	3
3. Choix et description des algorithmes utilisés.....	4
4. Conclusion.....	5

## 1. Rappel du sujet

Le jeu des six couleurs est un jeu de stratégie se déroulant sur un plateau découpé en cases de 6 couleurs différentes (rouge, orange, jaune, vert, bleu, ou violet). Le but du jeu est de contrôler plus de cases que les adversaires à la fin de la partie.

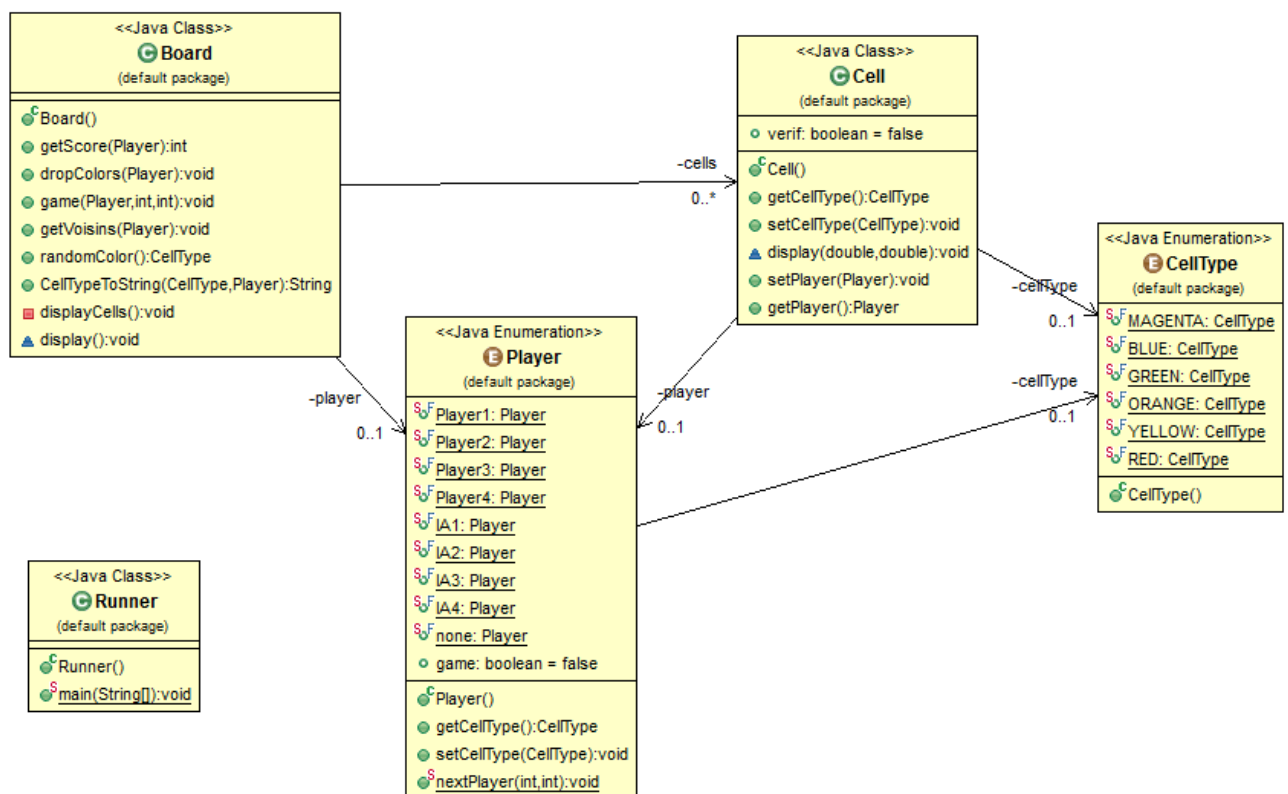
Les joueurs (de 2 à 4) commencent la partie en contrôlant chacun une case de la grille, ces cases doivent être de couleurs différentes. Les joueurs jouent chacun leur tour. À son tour, un joueur choisit une couleur différente de celle qu'il a actuellement, et non utilisée par ses adversaires.

- . Toutes les cases contrôlées par le joueur deviennent alors de la couleur choisie.
- . Toutes les cases de la couleur choisie et qui touchent une case contrôlée par le joueur passent sous son contrôle.

La partie s'arrête lorsqu'il n'y a plus de case non contrôlée, ou lorsqu'un joueur contrôle plus de la moitié des cases. Le joueur gagnant est celui qui contrôle le plus de cases.

## 2. Modélisation (description des classes)

Nous avons dans notre jeu cinq classes que nous vous présentons ci-dessous :



**RUNNER** : C'est la classe qui permet de lancer le jeu.

**BOARD** : C'est la classe principale du jeu. Elle contient une fonction d'initialisation du jeu depuis la console (choix du nombre de joueur, choix du nombre d'intelligence artificielle, définition du plateau de jeu...). Elle permet de créer le plateau du jeu, chaque cellule et de les colorer. Elle désigne aussi la position de chaque joueur. Nous y avons également inséré la méthode permettant le choix de la couleur par un joueur lors de la partie, humain ou IA. Elle permet aussi d'annoncer les scores des joueurs.

**PLAYER** : Enumération. Liste les joueurs et définit leur ordre de passage.

**CELL** : Cette classe permet de dessiner les cellules dans le mode graphique grâce à stdDraw.

**CELLTYPE** : Enumération. Liste les six couleurs disponibles pour les cellules du plateau de jeu.

### 3. Choix et description des algorithmes utilisés.

Commençons par la classe Board, la plus fournie :

**Board()** : Crée le plateau de jeu et initialise les joueurs

**getScore(Player)** : Obtiens le score du joueur en cours

**dropColors(Player)** : permet de jouer au jeu en sélectionnant une case

**game(Player)** : détermine le tour des joueurs et annonce la fin si la moitié du plateau est contrôlée

**getVoisins(Player)** : Obtiens tous les voisins du joueur de la cellule sélectionnée et en change la couleur

**randomColor()** : Crée une mise en forme random des couleurs de la grille

**CellTypeToString(CellType, Player)** : affiche le plateau de jeu sur la console

**displayCells()** : permet de créer les entités cells[ ][ ] et de les colorer

**display()** : crée la grille

Viens ensuite la classe Cell :

**getCellType()** : donne la couleur de la cellule

**setCellType(CellType)** : modifie la couleur de la cellule

**display(double, double)** : dessine grâce à l'interface graphique une cellule et la colorie

**setPlayer(Player)** : donne le joueur actuel

**getPlayer()** : change le joueur actuel

Dans la classe Player :

**getCellType()** : donne la couleur du joueur

**setCellType(CellType)** : modifie la couleur du joueur

**nextPlayer(int,int)** : permet le changement de joueur, dans le bon ordre

Dans la classe Runner :

**main(String[] )** : Permet que le tout démarre

## 4. Conclusion

Nous avons pu constater que l'organisation du projet Java doit être plus réfléchie que l'APP informatique en termes de programmation. En effet, il était ici beaucoup plus compliqué de répartir le développement en page parfaitement indépendantes, contrairement à un site internet. N'ayant pas eu beaucoup d'entraînement sur l'outil graphique, nous ne sommes pas vraiment satisfait du design plus que simpliste. C'est un point que nous aurions aimé approfondir si les projets annexe n'avaient pas perturbé notre organisation. Nous sommes tout de même satisfaits des progrès que nous avons fait dans nos compétences en Java. Certes notre jeu n'est pas des plus élégants, mais les compétences acquises sont solides.