Document de conception : Jeu des Six Couleurs

- Rédaction du manuel : Alexis Dahan et Nicolas Deltheil
- Date de rédaction : Mai 2016
- Numéro de version : 2.1

isep
École d'ingénieurs du numérique

Tuteur : Mr Rousset

Table des matières

Table des figures

Introduction

Ce rapport est le document de conception technique du Jeu des Six Couleurs.

Ce document a pour but de rappeler le sujet du projet, de modéliser les codes où il est demandé de décrire les classes utilisées ainsi que d'expliquer le choix et de décrire les algorithmes utilisés à la réalisation du jeu.

Vous trouverez l'intégralité du code de ce projet sur cette adresse Github :

https://github.com/Jimmy-Jones/JavaProject

Rappel du projet

Le but de ce projet est de programmer, en langage Java, un Jeu des Six Couleurs.

Rappel des règles du jeu :

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.

Modélisation

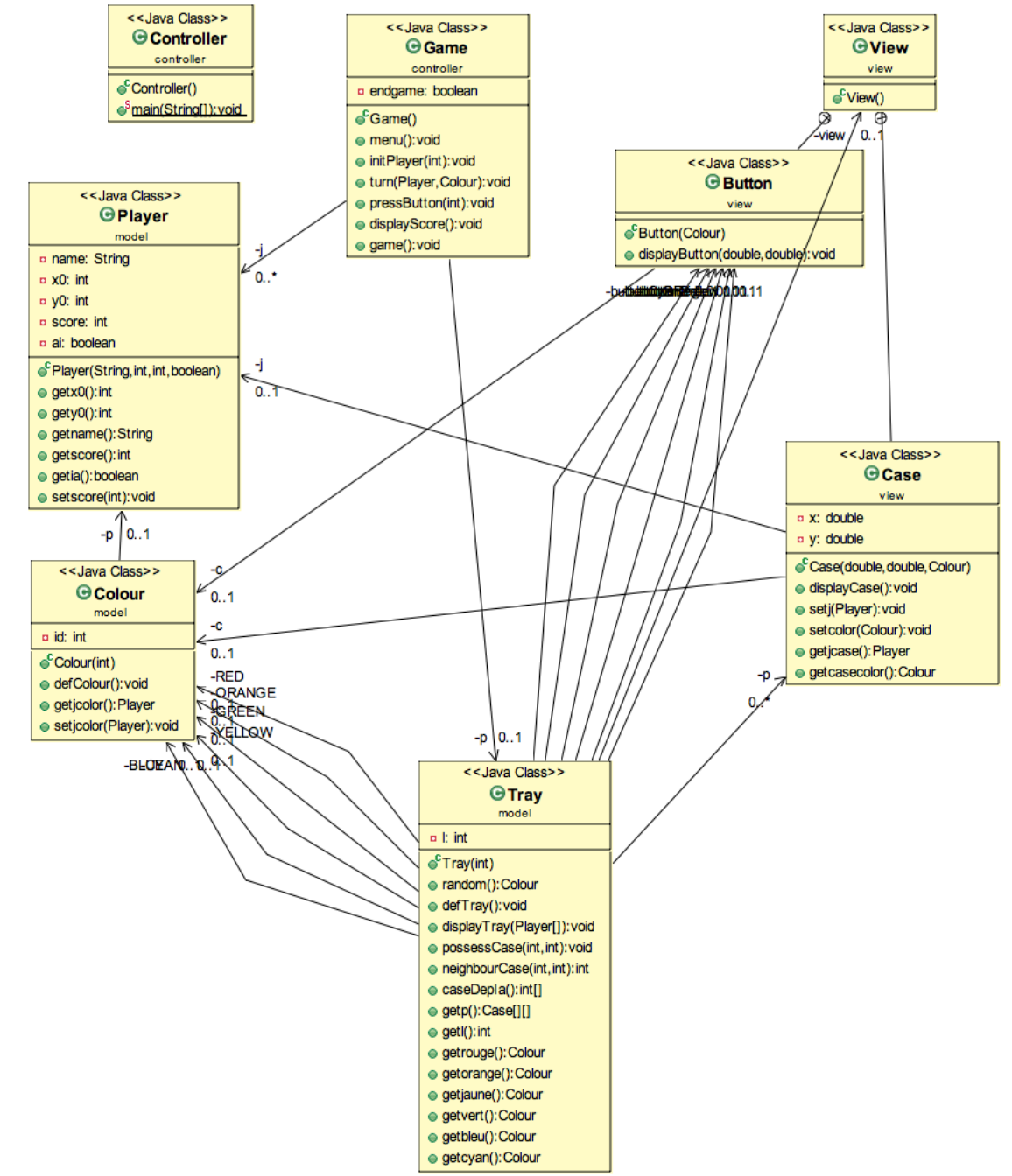

Figure 1: Diagramme UML du jeu

Explication du MVC :

Vue : elle contient la description de l'interface graphique avec ses différents composants,
Modèle : elle contient tous les attributs et méthodes qui doivent participer à la tâche principale à accomplir.
Contrôleur : une classe de contrôle qui contient l'ensemble des '*listeners*' et qui, lorsque des événements parviennent via la vue, prévient le modèle en conséquence.

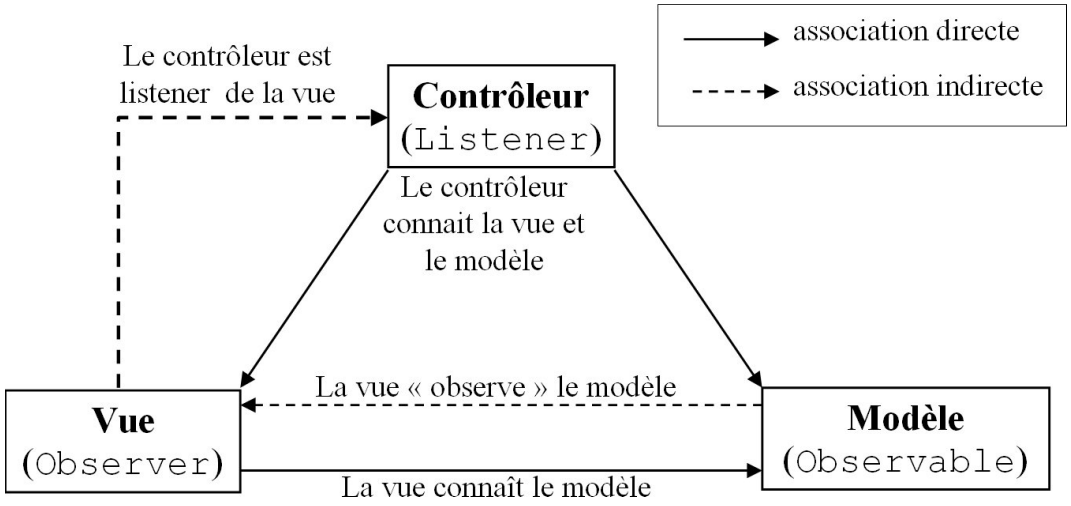

Figure 2: Schéma explicatif du MVC

L'intérêt de la méthode MVC lors de la programmation d'un jeu est une conception claire et efficace grâce à la séparation des données de la vue et du contrôleur.
Un gain de temps de maintenance et d'évolution du site afin d'y apporter des modifications et une plus grande souplesse pour organiser le développement du site entre différents développeurs (indépendance des données, de l'affichage (webdesign) et des actions).
On note que certaines fonctions qui devraient se trouver dans la « view » se trouvent dans le « model » ou dans le contrôleur, à savoir tout ce qui concerne le menu de départ et le menu de fin qui sont dans leur fonction propre pour éviter trop d'aller - retour avec la vue.
Après coup, nous trouvons que par soucis d'ergonomie du code, il aurait été plus judicieux de les placer dans le package « view ».

Description des classes :

- **Controller**
  - Controller

Il s'agit de la classe principale (*main*), elle sert à exécuter le jeu.

  - Game

Cette classe sert à structurer le jeu : à savoir l'affichage du menu, le choix du nombre de joueur.

- **Model**
  - Colour

Cette classe a pour but de définir les couleurs du jeu et affecter les couleurs à un joueur.

  - Player

Cette classe permet d'identifier le joueur : par ses coordonnées de la couleur choisie sur le plateau, son score et s'il s'agit d'une IA ou non.

- Tray

Cette classe permet le remplissage aléatoire du plateau (« Tray » en anglais) ainsi que savoir si le joueur peut prendre possession de ces cases voisines (diagonales exclues)

- **Vue**

- View

Dans cette classe principale se trouve deux classes :

- Button

Cette classe permet l'affichage des boutons de couleurs proposées pour jouer.

- Case

Cette classe permet de décrire les cases en fonction de leur coordonnée sur le plateau, leur couleur et leur joueur associé.

Choix et description des algorithmes utilisés

Nous avons programmé ce jeu uniquement en mode graphique afin de présenter pour la soutenance un travail aboutie avec une bonne visuelle et finir par programmer le mode console. Nous avons étudié l'implémentation de l'affichage en mode console dans le package « View » mais par une mauvaise estimation du temps nécessaire pour la programmation de cette « extension », nous n'avons pas pu, dans cette première version, y intégrer le mode console.

Extension : Graphique

Pour la partie graphique du jeu, nous avons opté par la librairie *StdDraw*.
Se trouvant dans « Game » dans le package « Controller ».

Figure 3: Menu d'accueil

La fonction utilisée pour coder ce menu est « `public void menu()` » :

```
public void menu(){
    StdDraw.setCanvasSize(800,600);
    StdDraw.setXscale(-0.5,p.getl()+3.5);
    StdDraw.setYscale(-2.5,p.getl()-0.5);
    StdDraw.picture(7.5, 5, "imgmenu.png");
        StdDraw.setPenColor(StdDraw.BLACK);
        StdDraw.setFont(new Font("arial", Font.BOLD,56));
        StdDraw.text(p.getl()/2+2, p.getl()-3, "Six colours game");
        StdDraw.setFont(new Font("arial", Font.BOLD,35));
        StdDraw.text(p.getl()/2+2, 8, "Number of players");
        StdDraw.setFont(new Font("arial", Font.BOLD,20));
    StdDraw.circle(p.getl()/2+2, 6,p.getl()/8);
    StdDraw.circle(p.getl()/2+2, 3,p.getl()/8);
    StdDraw.circle(p.getl()/2+2, 0,p.getl()/8);
    StdDraw.setPenColor(StdDraw.ORANGE);
    StdDraw.filledCircle(p.getl()/2+2, 6, p.getl()/8);
    StdDraw.setPenColor(StdDraw.GREEN);
    StdDraw.filledCircle(p.getl()/2+2, 3,p.getl()/8);
    StdDraw.setPenColor(StdDraw.YELLOW);
    StdDraw.filledCircle(p.getl()/2+2, 0,p.getl()/8);
    StdDraw.setPenColor(StdDraw.BLACK);
    StdDraw.text(p.getl()/2+2, 6, "2 players");
    StdDraw.text(p.getl()/2+2, 3, "3 players");
    StdDraw.text(p.getl()/2+2, 0, "4 players");
    int choice =0;
```

Figure 4: Codes Menu d'accueil

Il a fallu paramétrer l'ensemble de ce qui s'affiche sur le menu d'accueil : l'image en arrière-plan, la police et la taille des écritures et la couleur pour l'emplacement des boutons pour le choix du nombre de joueurs.

Se trouvant dans « Game » dans le package « Controller »,

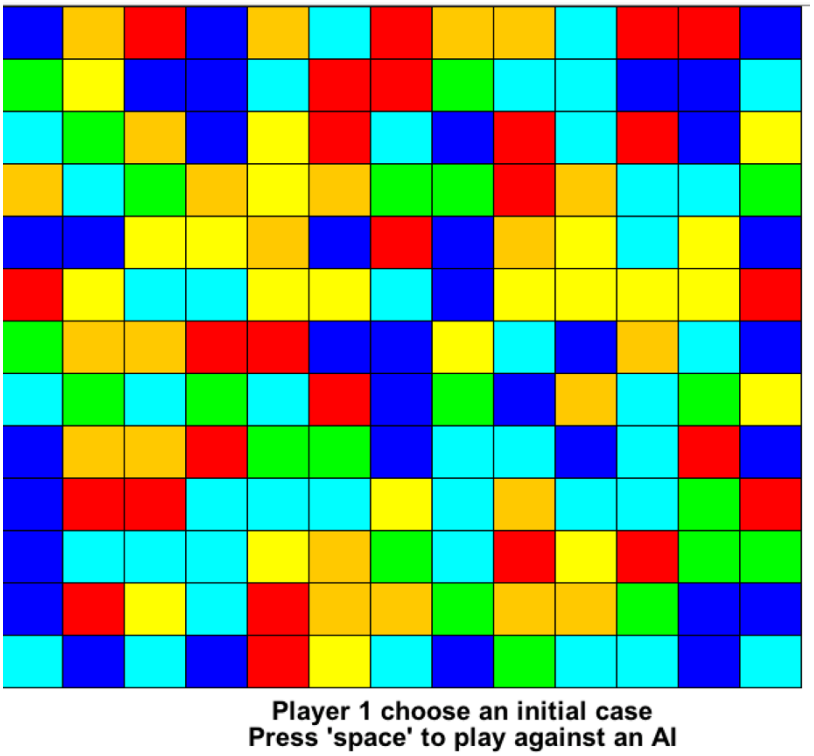

Figure 5: Initialisation du début de la partie

La fonction utilisée est « `public void initPlayer(int nb)` ».

Se trouvant dans « Tray » dans le package « Model » :

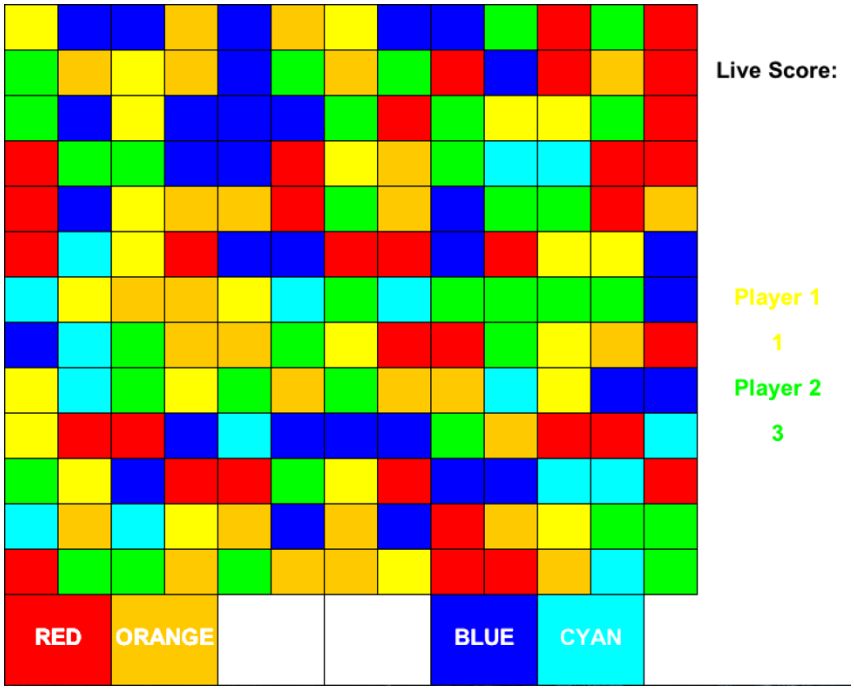

Figure 6: Interface du jeu pendant une partie

On constate que les couleurs non disponibles pour un joueur ne s'affichent pas et que le nom du joueur est de la couleur de sa dernière couleur choisie.

Se trouvant dans « Game » dans le package « Controller » :

Figure 7: Interface à la fin du jeu annonçant le gagnant

```
public void displayScore(){
    StdDraw.setPenColor(StdDraw.BLACK);
    Player count=j[0];
    int best=j[0].getscore();
    for(int i=1;i<this.j.length;++i){
        int score=j[i].getscore();
        if(score>best){
            best=score;
            count=j[i];
        }
    }
    StdDraw.clear();
    StdDraw.setPenColor(StdDraw.BLACK);
    StdDraw.setPenRadius(1000);
    StdDraw.picture(7.5, 5, "imgwinner.png");
    StdDraw.setFont(new Font("arial", Font.BOLD,35));
    StdDraw.text(p.getl()/2,p.getl()/2,"And the winner is: "+ count.getname());
    StdDraw.setFont(new Font("arial", Font.BOLD,25));
    StdDraw.text(p.getl()/2,p.getl()/2-1,"with "+Integer.toString(count.getscore())+" points");

}
```

Figure 8: Code de l'Interface à la fin du jeu annonçant le gagnant

Extension : Choix du nombre de joueurs

Sur la capture du menu d'accueil du jeu (voir figure 3), il est possible de choisir le nombre de joueurs pour une partie. Il est possible de jouer à 2, 3 et 4 personnes (IA comprise).
Se trouvant dans « Game » dans le package « Controller », le code associé est le suivant :

```
//Different choices of players
while(true){
    if(StdDraw.mousePressed()){

        int x=(int)Math.round(StdDraw.mouseX());
        int y=(int)Math.round(StdDraw.mouseY());
    if(x<=8 && x>=8 && y<=7 && y>=5 ){
        choice=2;
        break;
        }
    if(x<=8 && x>=8 && y<=4 && y>=2 ){
        choice=3;
        break;
        }
    if(x<=8 && x>=8 && y<=1 && y>=-1 ){
        choice=4;
        break;
        }
    }}
    p.displayTray(this.j);
        this.initPlayer(choice);
```

Figure 9: Code du choix du nombre de joueur pour une partie

```
public void displayTray(Player[] play){
    StdDraw.clear();
    for(int i=0;i<l;i++){
        for(int j=0;j<l;j++){
            p[i][j].displayCase();
        }
    }
    if(play!=null){
        StdDraw.setFont(new Font("arial", Font.BOLD,20));
        StdDraw.text(this.l+1,this.l-2,  "Live Score:");
    for(int t=0;t<play.length;++t){
        if(play[t]!=null){
            this.p[play[t].getx0()][play[t].gety0()].getcasecolor().defColour();

        StdDraw.text(this.l+1,this.l-2-2*t-5,play[t].getname());
        StdDraw.text(this.l+1,this.l-3-2*t-5,Integer.toString(play[t].getscore()));

        StdDraw.setPenColor(StdDraw.BLACK);
        StdDraw.square(l, t, l+1);
    }}}
```

Figure 10: Code de l'apparence graphique en fonction du nombre de joueur

Cette algorithme renvoi pour un certain joueur son nom et sa couleur afin de modifier l'apparence graphique.

Extension : Intelligence Artificielle

Se trouvant dans « Tray » dans le package « Model »,

```
public int[] caseDepIa(){
    int score=0;
    int x=0;
    int y=0;
    for(int i=0;i<this.l;++i){
        for(int j=0;j<this.l;++j){
            int provi=Math.max(score, neighbourCase(i,j));
            if(provi>score){
                x=i;
                y=j;
                score=provi;
            }
        }
    }
    int[] tab={x,y};
    return tab;
}
```

Figure 11: Code de l'Intelligence Artificielle

Cette algorithme décrit l'intelligence artificielle du jeu, cette IA choisit la couleur lui permettant de capturer le plus de cases, en comptant le maximum de cases disponibles de la même couleur autour de sa zone propriétaire.

Dans la version multijoueur, il est indiqué quels sont les joueurs humains et quels sont les joueurs IA :

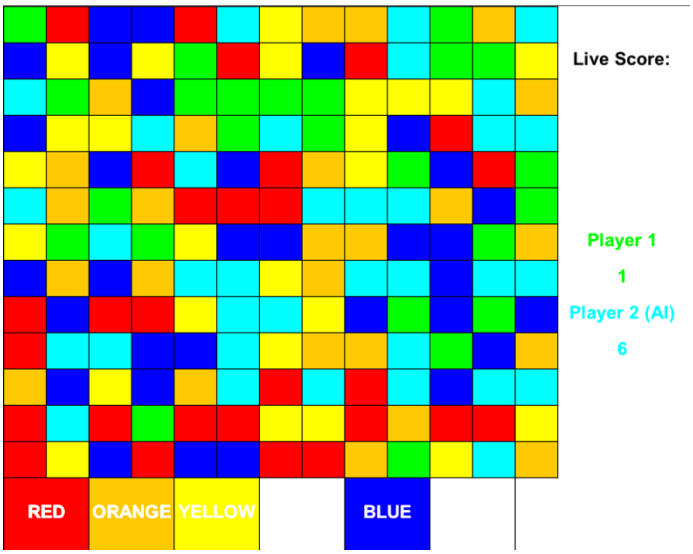

Figure 12: Distinction graphique entre le joueur et une IA

Conclusion

Pour la matière « Algorithmique et programmation » enseignée à l'ISEP, nous devions réaliser comme projet de fin de semestre un Jeu des Six Couleurs.
Réalisé uniquement en mode graphique et non en mode console, ce jeu nous a permis d'utiliser la rigueur et les méthodes vues pendant les cours et pendant les TP avec aussi d'autres supports tels qu'Internet.
Les techniques vues justement lors des TP nous ont été fondamental à la réalisation de ce jeu qui reprend plusieurs méthodologies de réflexions rencontrées pendant ces séances de travaux pratiques.

Nous apporterons l'attention sur le nombre de vus de notre projet sur **GitHub**, ayant un dépôt public, ce projet peut être visible et copié par des personnes mal intentionnées.

Sources :

http://perso.telecom-paristech.fr/~hudry/coursJava/interSwing/boutons5.html
https://openclassrooms.com/courses/l-algorithme-min-max