

# $\pi$ Cross



QUILFEN Anthony et GROSLEVIN Rémi

*Rapport du jeu de Picross réalisé en JAVA dans le cadre du projet en Programmation Orientée Objet*

IUT de Dijon - Rapport rendu le 14 juin 2012

---

# Modélisation du Jeu

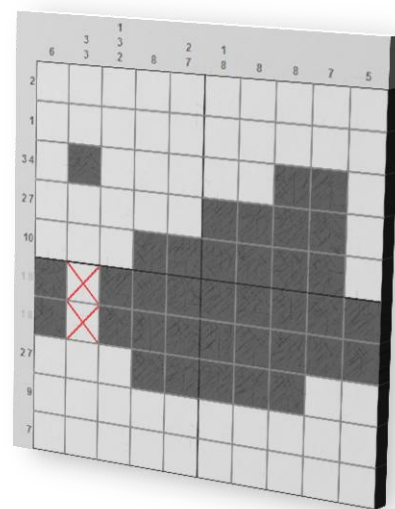
---

## Démonstration d'une partie de jeu

*Le jeu de Picross consiste à remplir une grille de jeu, à l'aide d'informations concernant le nombre de cases à cocher dans les lignes et les colonnes. Le joueur remplit chaque case de la grille une-à-une jusqu'à ce que celle-ci soit complétée. Au final, elle laisse apparaître un dessin.*

Une grille de Picross classique se compose :

- D'une grille à deux dimensions
- D'informations sur les lignes et les colonnes
- D'un quadrillage pour assister le joueur



Pendant la partie, le joueur aura la possibilité de voir le nombre d'erreurs commises, son score actuel ainsi qu'une vue d'ensemble du dessin qu'il est en train de découvrir. Une barre de menu accessible en haut de la fenêtre lui permet de faire une nouvelle grille, de quitter le jeu ou encore d'afficher le manuel d'aide. Si le joueur pense avoir découvert une erreur, il peut sécuriser la case, à l'aide du clic droit, avec une croix noire. Cette case est figée jusqu'à un second clic droit sur elle.

# Diagramme des Classes et Agrégations

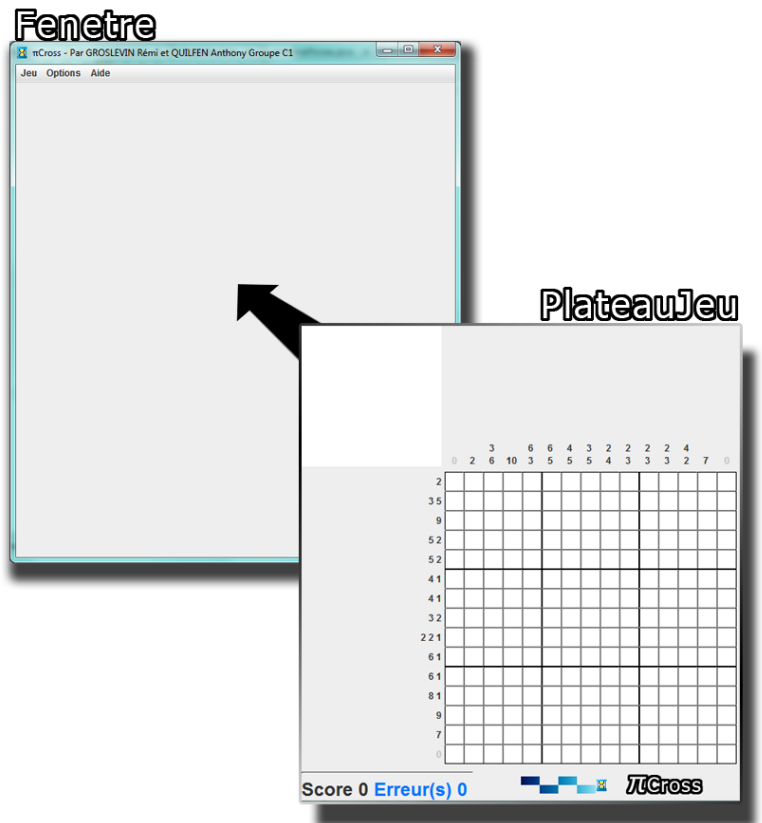
Chaque classe remplit un rôle, soit dans l'interface, soit dans le déroulement de la partie.

## La classe **Fenetre**

Cette classe, héritant de JFrame, est la fenêtre complète du logiciel. Elle affiche une barre de menu en haut permettant de contrôler la partie, et un plateau de jeu au centre.

## La classe **PlateauJeu**

Héritant de JPanel, le plateau de jeu est situé dans la fenêtre du logiciel (voir image ci-contre). Il contient la grille, les informations relatives à la partie en cours et la vignette. Ce plateau est référencé dans la fenêtre et affichée au centre de cette dernière.



## La classe **Manuel**

Héritant de JFrame, le Manuel est la fenêtre d'aide que l'on peut ouvrir via le menu d'Aide.

## La classe **CaseJeu**

Héritant de JPanel, cette classe représente une case de la grille de jeu. On change son état en cliquant dessus. Sa couleur peut changer et une croix peut être dessinée dedans. La croix est de la taille de la case même après redimensionnement :

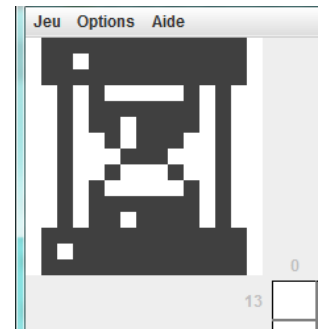


## La classe **Croix**

Cette classe est une croix pouvant être dessinée dans une **CaseJeu**. Elle possède une largeur, une hauteur et une couleur qui s'actualise à chaque rafraichissement de la fenêtre.

## La classe **PixelVignette**

Héritant de JPanel, cette classe représente un pixel de la vignette en haut à gauche de PlateauJeu. Il possède deux coordonnées en x et y du pixel ainsi que la couleur de ce dernier. Un PixelVignette aux coordonnées (x,y) est lié à la CaseJeu aux coordonnées (x,y).



## La classe **ImageJeu**

Cette classe est l'image à trouver lors d'une partie. Elle permet de récupérer de générer toute partie à partir d'une simple image bitmap de profondeur 24bits. L'atout de cette classe est de pouvoir récupérer la couleur du pixel aux coordonnées x et y.

## La classe **RegleJeu**

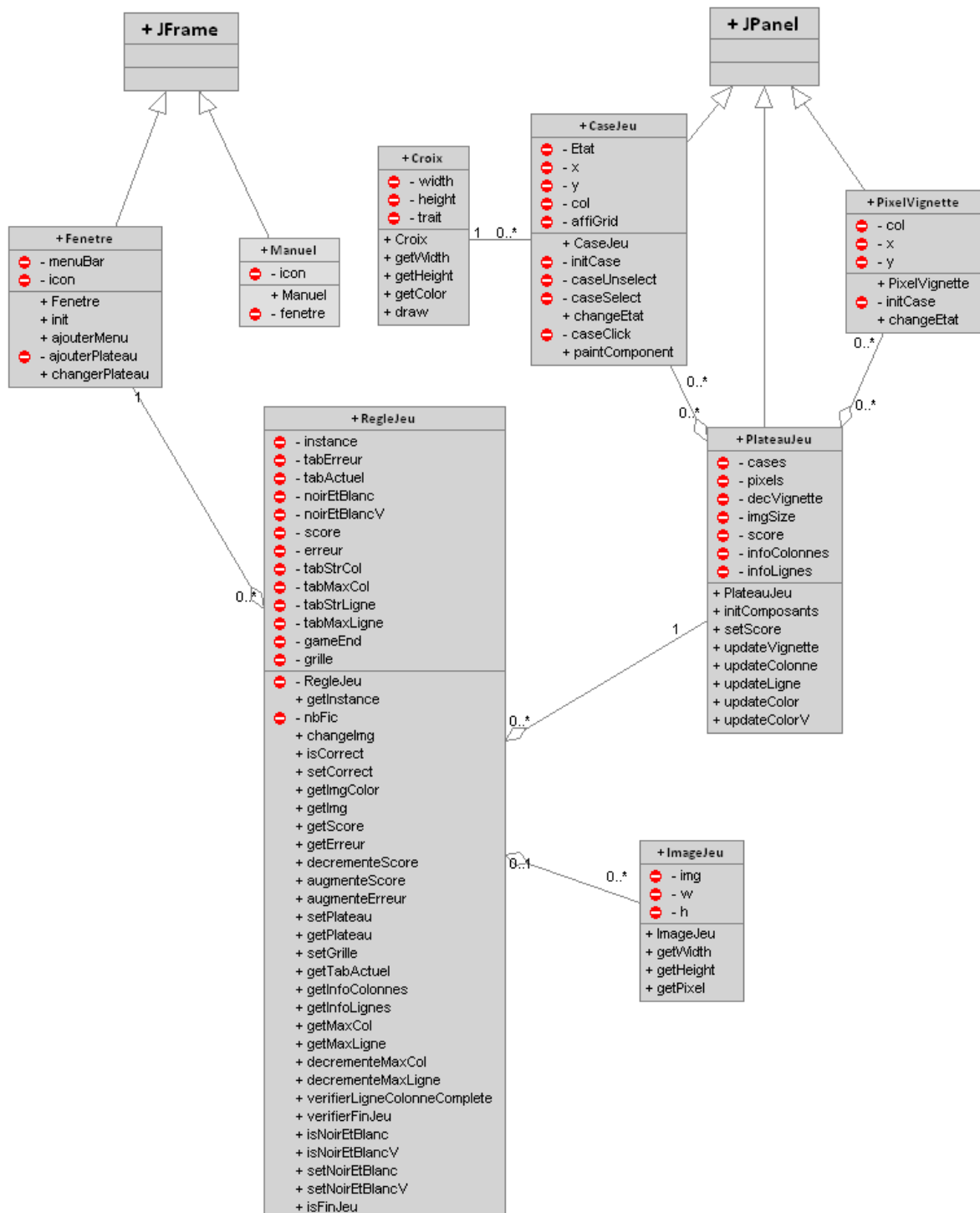
Cette classe est le centre du jeu. Toutes les règles et tous les traitements y sont définis. Lors d'une partie, il y a 1 plateau de jeu, 1 score, 1 image, 1 grille de jeu à gérer etc... RegleJeu est un objet défini de façon unique (singleton) contenant toutes les données du jeu en cours et interconnectant toutes les autres classes du jeu.

## Design du jeu

Le design est classique. Il offre un PiCross sobre et classe avec une couleur bleue dominante. Nous avons choisi de donner un sablier en guise de logo à notre logiciel puisqu'il est la première image que nous avons colorisée pour le mode jeu en couleurs.

Pour paraître plus attrayant, nous avons modélisé le sablier en 3 dimensions et le résultat est appréciable dans le logiciel ainsi que sur la page de garde de notre rapport.

# Diagramme UML

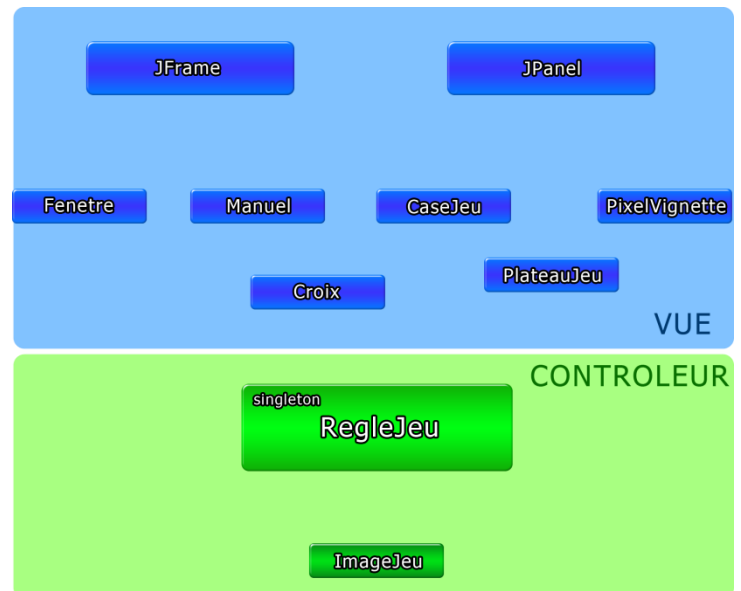


# Implémentation en Java

## Modèle et Contrôleur

La classe **RegleJeu** est la classe centrale du logiciel. Nous avons tenté de réaliser le design pattern **MVC**. Mais ayant rencontré de nombreuses difficultés dans l'implémentation en java, nous avons rassemblé **Modèle** et **Contrôleur** en une même classe. **RegleJeu** choisit une image au hasard parmi celles fournies dans le dossier accompagnant le jeu, et fabrique une partie à partir de cette dernière.

La classe **ImageJeu** peut faire partie du **modèle** puisqu'elle contient les données des pixels de l'image du jeu.



## Vue

La **Vue** est composée de plusieurs classes liées et gérée par **RegleJeu**. Nous n'avons pas implémenté l'interface **Observer** ni même hérité la classe **Observable**. Mais nous avons gardé à l'esprit que c'est au **Modèle** (**RegleJeu**) de signaler à la **Vue** qu'elle doit se mettre à jour. Ainsi **RegleJeu** se voit munir de fonctions appelant toutes les méthodes **update** de la **Vue**.

## Améliorations

Nous avons conçu notre jeu avec les améliorations que nous allons apporter pour faciliter l'implémentation des classes. Parmi ces améliorations :

- ▶ **Jeu en couleurs** : Possibilité de jouer en couleurs (Vignette et jeu séparément)
- ▶ **Lire les images bitmap** : Une partie de Picross se fait à l'aide des images bitmap (24bits) présentes dans le dossier img. Il suffit d'ajouter une image adéquate dans le dossier pour rajouter une grille dans le jeu. Les pixels blancs sont les erreurs, les autres couleurs sont à trouver.
- ▶ **Grille de jeu 5x5** : Toutes les 5 cases, les traits de la grille sont accentués pour une meilleure lisibilité du quadrillage.
- ▶ **Usage du code HTML** : Design de l'interface pensée et structurée en HTML pour la simplicité de mise en page.
- ▶ **Survol de la Case** : La case survolée est encadrée en bleu et ses coordonnées s'affichent dans une infobulle.

# Manuel d'instructions

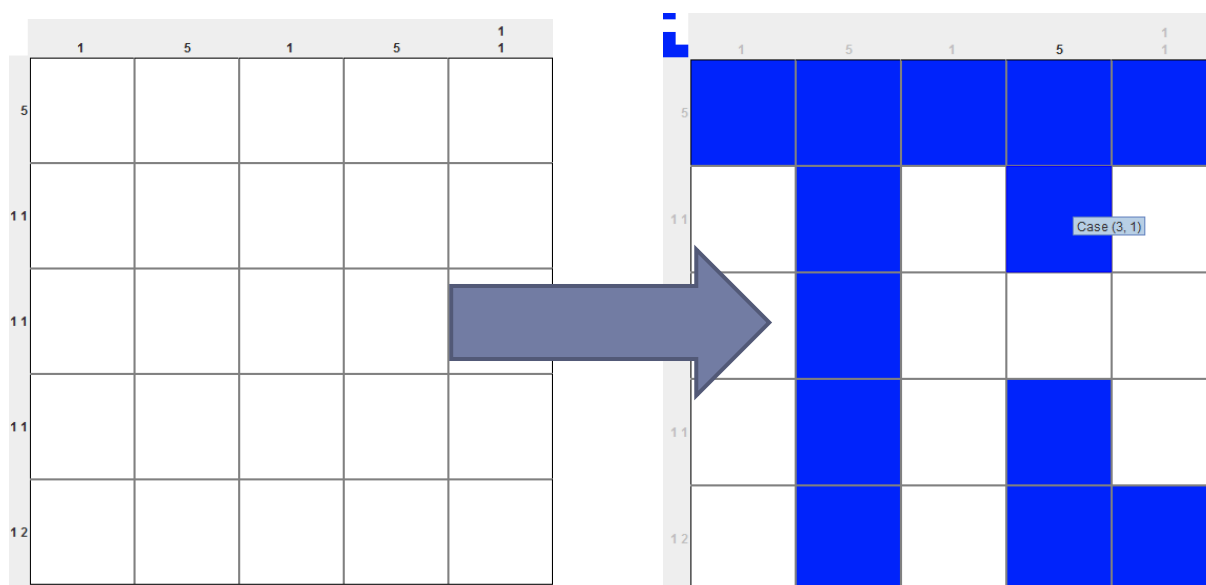
## Lancer une partie

1. Lancez le programme

Une grille 15x15 est générée par défaut. Vous pouvez lancer une nouvelle partie en allant dans le menu *Jeu -> Nouvelle Partie* en choisissant le niveau de difficulté.

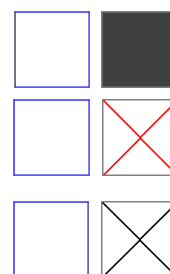
| Jeu                             | Options | Aide   |
|---------------------------------|---------|--------|
| 5x5 - <u>N</u> ouvelle Grille   |         | Ctrl-1 |
| 10x10 - <u>N</u> ouvelle Grille |         | Ctrl-2 |
| 15x15 - <u>N</u> ouvelle Grille |         | Ctrl-3 |
| 20x20 - <u>N</u> ouvelle Grille |         | Ctrl-4 |
| <u>Q</u> uitter                 |         | Ctrl-Q |

2. Remplissez la grille en s'aidant des informations placées horizontalement et verticalement. Elles correspondent au nombre de cases à remplir consécutivement



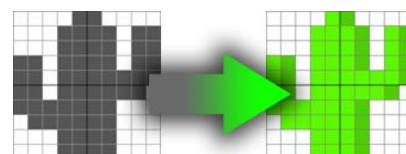
3. Si vous trouvez une case correcte, celle-ci se grise.

Si la case est incorrecte, c'est une erreur et une croix rouge se dessine.



4. Si vous pensez avoir trouvé une case incorrecte, cochez-la à l'aide du clic droit. La case est sécurisée.

5. Vous avez la possibilité de jouer en couleur en allant dans le menu *Options->Couleurs/Noir et Blanc*. Choisissez de mettre en couleur, la vignette ou le jeu.



6. Pour obtenir de l'aide, allez dans le menu *Aide -> Manuel* pour plus de détails.