Technical documentation

Global structure

The code is organized using the **MVC** (Model - View - Controller) model. It means there are 3 main packages :

- **model** : Contains all objects and data
- **view** : Allow to display the game
- **control** : Allow the user to interact with the game using inputs

And there is a Main.java containing the main loop of the game.

Model structure

The model structure is also splitted in different classes. The more important ones are listed here.

- **Game.java** : Singleton allowing access to all the data of the game;
- **TimeBoard.java** : The board where to move the pawn;
- **Patch.java** : Immutable record representing a patch;
- **Player.java** : Contains all data concerning a single player;
- **QuiltBoard.java** : Allow manipulation of patches with a player quilt board.

View structure

The view has two important classes.

- **ViewConsole.java** : Draw the game on the console for version 1 and 2;
- **ViewWindow.java** : Draw the game on a graphical window for version 3 and 4.

The two classes are independent and has no implementation in common because of their behavior which are too different.

Control structure

The control has two important classes.

- **ControlConsole.java** : Manage input on console version of the game;
- **ControlWindow.java** : Manage input on windowed version of the game.

Those classes have no common implementation because of their different behavior, like views classes.

Update since version 2

- Created a complete graphical interface;
- The player can now flip, rotate and place a patch at the position of his choice on his quilt board;
- The user can choose a file containing a list of patches to play a personalized version of the game.