Game Hanabi

dev manual

2019/2020

Project by :

MAKENDI Bryan

YAHYAOUI Mohamed Yassine

Class Hierarchy :

Classe Interface Abstract Classe Enum

Classe Hierarchy of the project :

java.lang.Object

- fr.umlv.board.Board
  - Board
  - Token
- fr.umlv.card.Card
  - Card
  - Deck
  - CardColor
- fr.umlv.clue.Clues
  - AskClues
  - Clues
  - ClueTypes
- fr.umlv.hanabi.Hanabi
  - AbstractGameController
  - Actions
  - Coordinates
  - Hanabi
  - InitGame
  - SimpleGameController
  - SimpleGameControllerGraphique
  - SimpleGameControllerConsole
  - SimpleGameView
  - SimpleGameViewConsole
  - SimpleGameViewGraphique
  - SimpleGameData
- fr.umlv.player.Player
  - Player
  - Hand

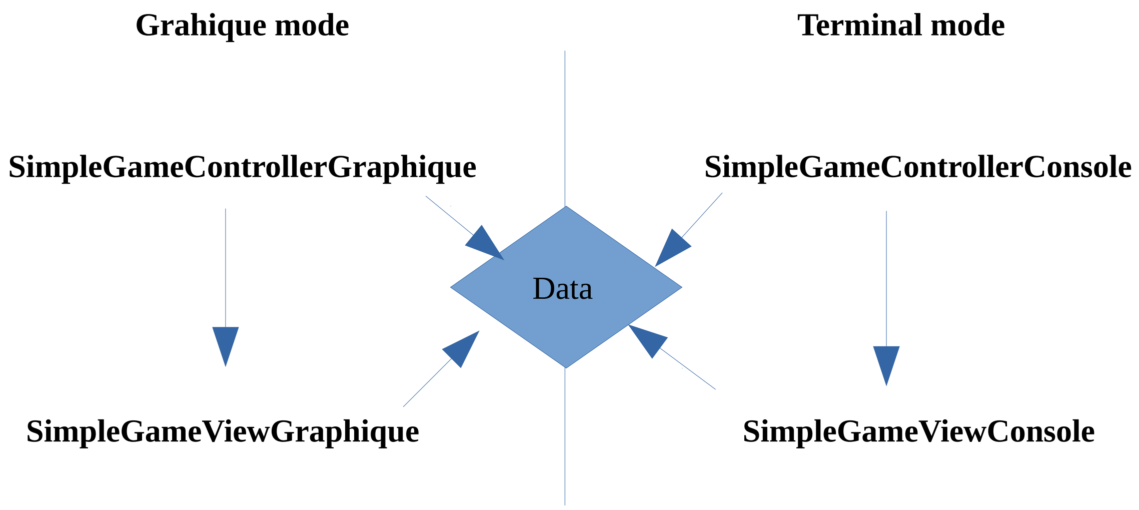
Enum Hierarchy

- java.lang.Object
  - java.lang.Enum<E> (implements java.lang.Comparable<T>, java.lang.constant.Constable, java.io.Serializable)
    - fr.umlv.card.CardColor
    - fr.umlv.clue.ClueTypes

Changes and corrections from last version

In the last version we didnt occupy of graphic interface, so the Hierarchy of the project was mainly for allowing and analysing the entry of the user, with no regards of the diffrences between the two games modes !

As result the controller for the First MVC occupied of only regestering the entry of the user and allowing the specific actions, to fix that , we saperated the MVC in two versions, for the Controller and the View,

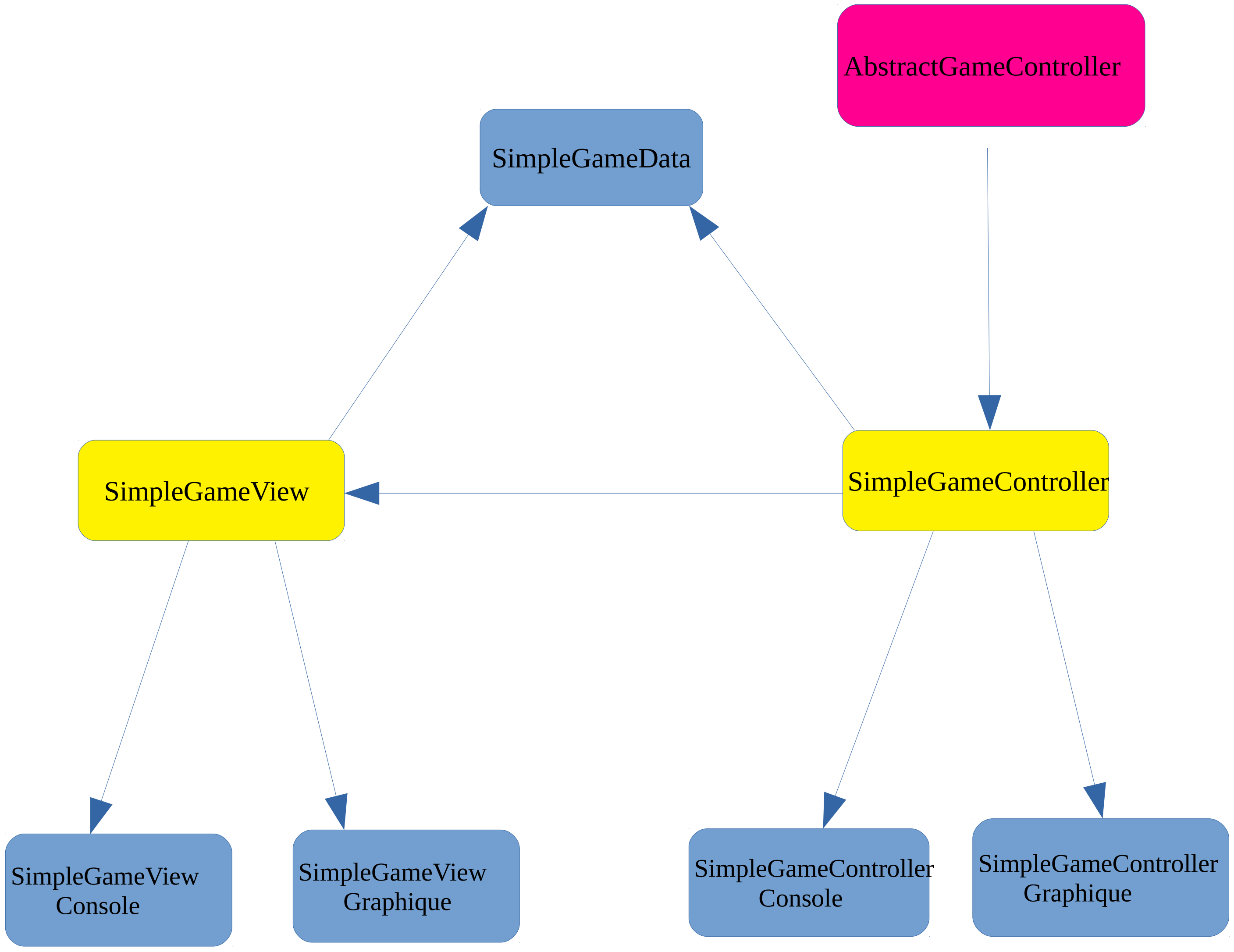


bare explication of the MVC model adapted

SimpleGameData by definition contains only the data of the game, which is the same on the two modes, so there is no reason to fully saperate the two MVCs,

certain methods is shared between the Controller graphique and the controller Console, also for the view graphique and view console, to assemble the two versions we implemented two interfaces :

representation of the final MVC used :



🟥 Abstract classe

🟦 classe

🟨 Interface