**SPLENDOR PROJECT**

**Splendor’s official PDF:**

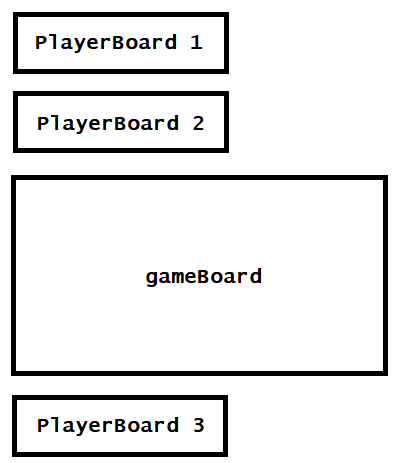
<https://cdn.1j1ju.com/medias/7f/91/ba-splendor-rulebook.pdf>

**Overview:**

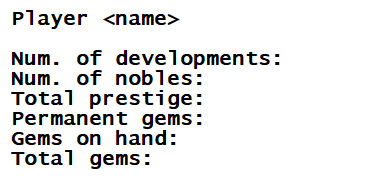
The goal is to first create a console based Splendor and then convert it into a java GUI game.

**Display ideas:**

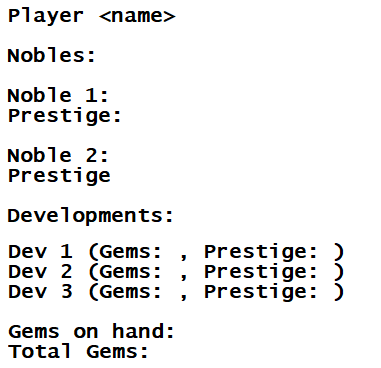
1. **Main**

****

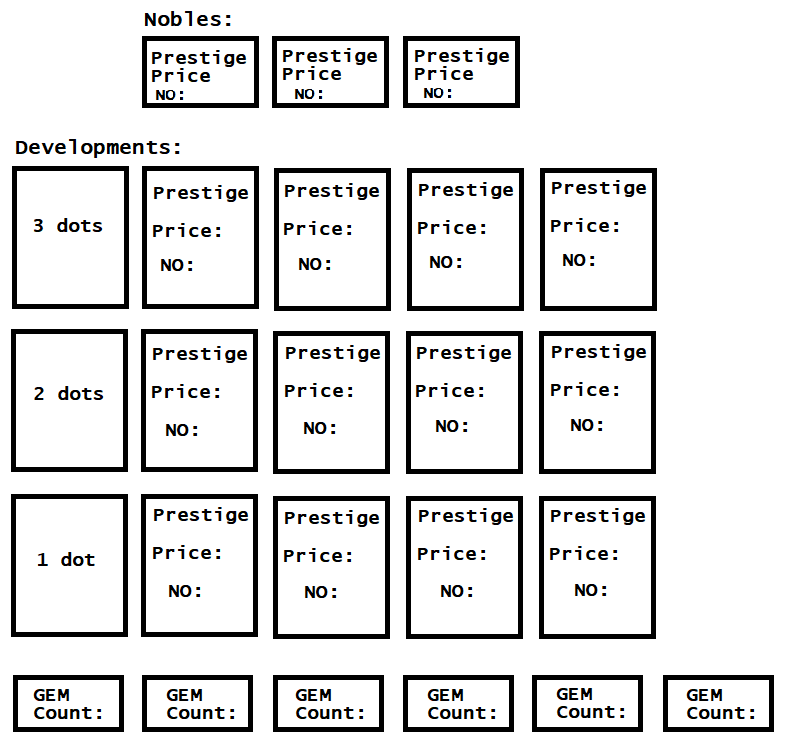
1. **PlayerBoard (opponent)**

****

1. **PlayerBoard (personal)**

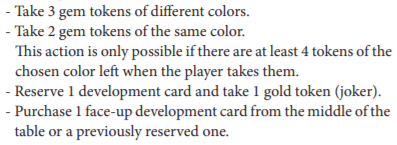
****

1. **Game Board**

****

**GAMEPLAY IDEA:**

As stated on the gameplay’s PDF, on each move, a player can:



The initial part of the project will be on making a command prompt gameplay with 2 different move options, >take, >purchase, and >reserve. The gems will be indicated with the chars: W (white), R (red), G (green), O (black), B (blue), and E (gold).

If a player is to take 3 different gems:

>take R G O

Take 2 same coloured gems:

>take W W

Reserve a development card (take 1 gold token):

>reserve <development number>

Purchase a face-up-development card from the table or previously reserved one:

>buy <development number>

Input validation checks will be provided accordingly.

**OBJECTS**

1. **Player**

The Player class will provide the following:

* Name
* Player no (1/2/3)
* Deck
* Prestige (provided in deck as well)

1. **PlayerDeck**

The PlayerDeck class will provide the following:

* Developments (collection of developments)
* Nobles (collection of nobles)
* Permanent gems
* Gems at hand
* Total gems (permanent gems + gems at hand)
* Prestige

Functionalities:

* Add development (automatically adds prestige & permanent gems)
* Add nobles (automatically adds prestige)
* Add gems (gems on hand)
* Print deck (print for personal deck & public deck)

1. **GameBoard**

The GameBoard class will provide the following:

* Three dot cards (collection of hard coded cards)
* Two dot cards (collection of hard coded cards)
* One dot cards (collection of hard coded cards)
* Gems:
* Gold gems 4 (always)
* 5 other gems each (3 players)
* 4 other gems each (2 players)
* Nobles (collection of hard coded nobles)

Functionalities:

* Take a development
* Take gems (3 different gems or 2 same gems but only if there are 4 of the same colours left in the stack)
* Reserve a development (automatically gives the player a gold gem)
* Receive gem payment (if a player purchases a development then refill the gem stacks accordingly)
* Refill development (every time a player purchases a development, refill it with cards from the developments stack according to the number of dots)

1. **Cards**

The Cards class will provide the following:

* Prestige
* Price
* Gem type

1. **GameEngine**

The GameEngine class will be providing the functionalities of the game. There will be 2 GameEngine class implementing the same interface for 2 and 3 players (or possibly a basic AI as well).