**SPLENDOR PROJECT**

**Written by Julian Tjiong**

**Splendor’s official PDF:**

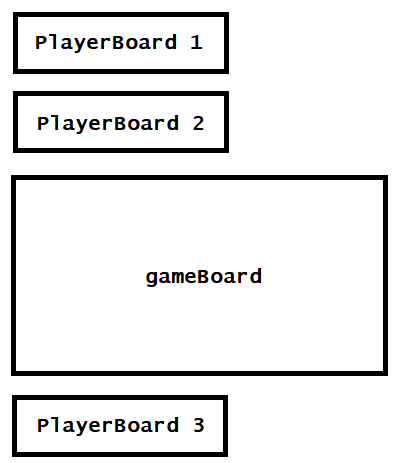
<https://cdn.1j1ju.com/medias/7f/91/ba-splendor-rulebook.pdf>

**Overview:**

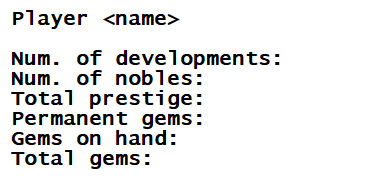
The goal is to first create a console-based Splendor and then convert it into a java GUI game.

**Display ideas:**

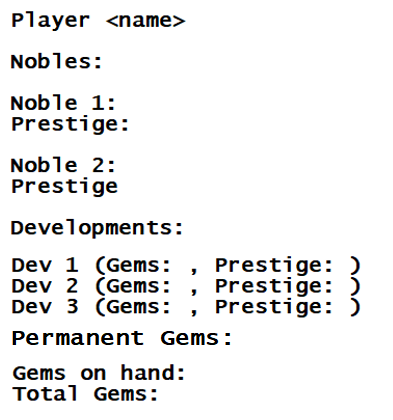
1. **Main**

****

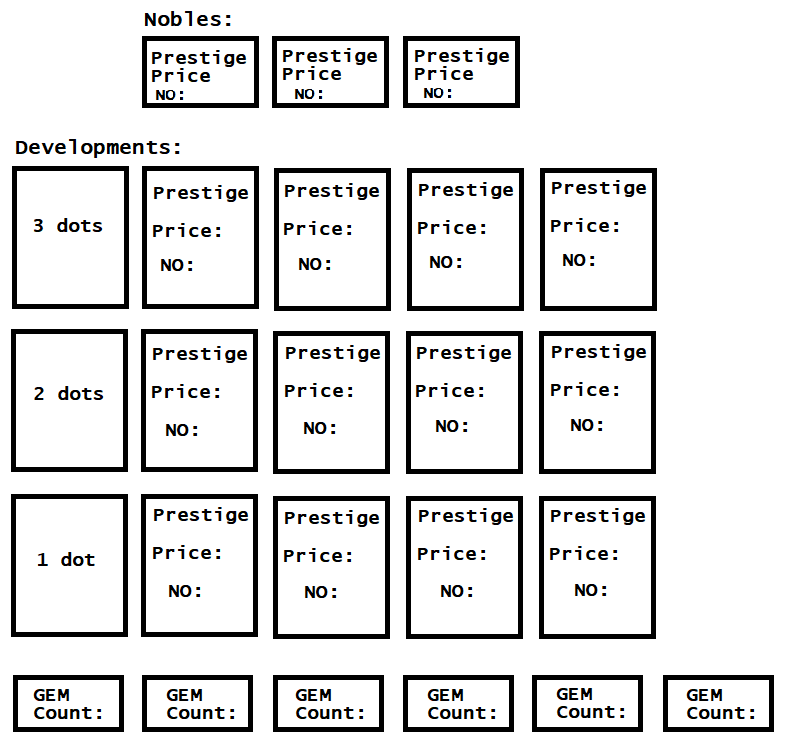
1. **PlayerDeck (public)**

****

1. **PlayerDeck (personal)**

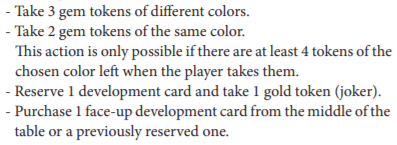
****

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)
* Check noble (whether eligible for nobles’ visits)
* Check development (whether gems are sufficient)
* Check gems (can’t have more than 10 hand gems)

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, number of players+1)

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)
* Print board

1. **Card**

The Card class will provide the following:

* Card number
* Prestige
* Price
* Gem type (permanent)
* Print Card

1. **Noble**

The Noble class will provide the following:

* No
* Prestige
* Price
* Print Noble

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.

After loading up the **Main** containing GameBoard and PlayerBoards as shown on the Display Ideas section, we need to **run checks on user input:**

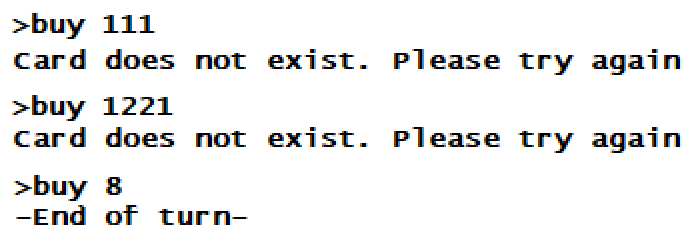
1. **Buying development:**



Above is the GameBoard

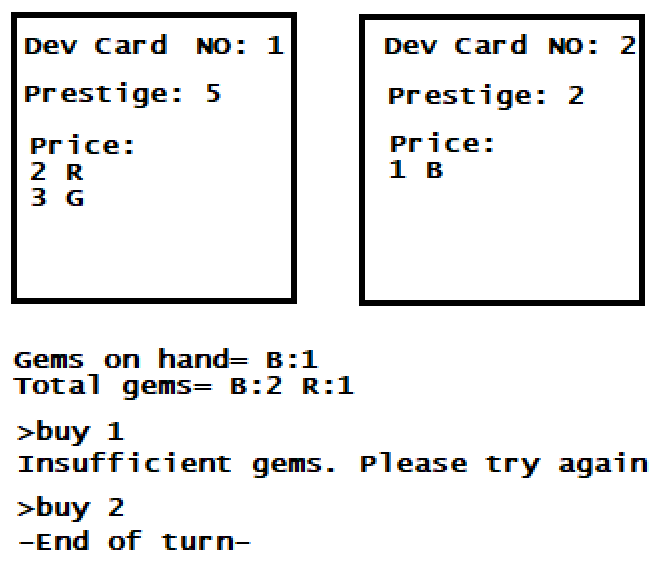
Now if the user’s to enter >take 11, we need to run check on whether or not the card number exists and if player has sufficient gems to pay the price.

Below is the possible scenarios for inexistent cards:

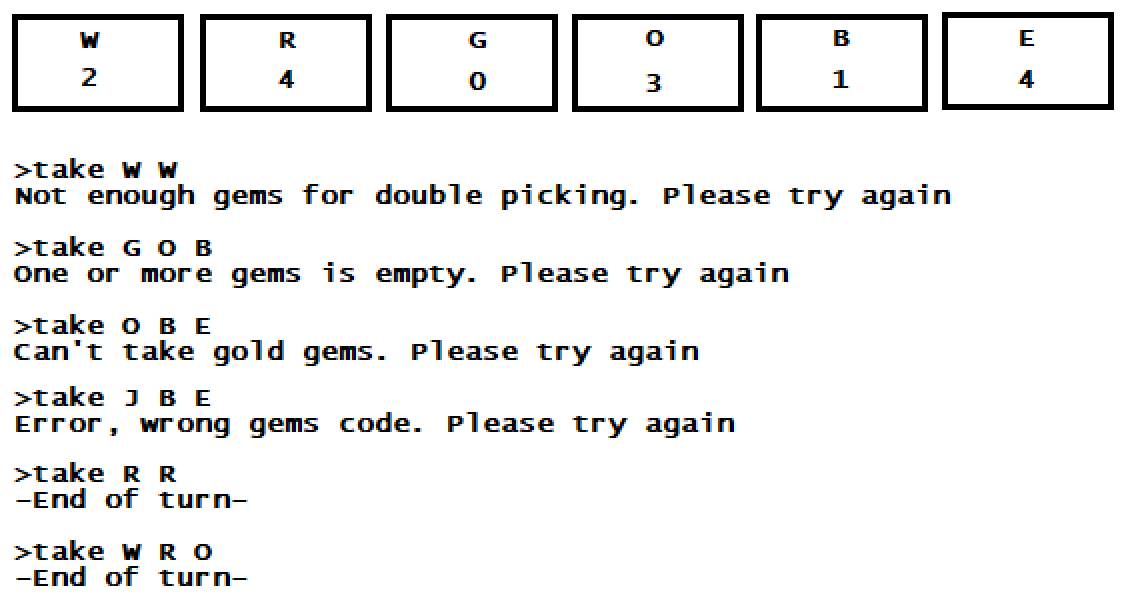


Card 8 does exist so taking it will end a player’s turn.

Now to check whether or not player has sufficient gems:



1. **Taking gems:**



Required checks:

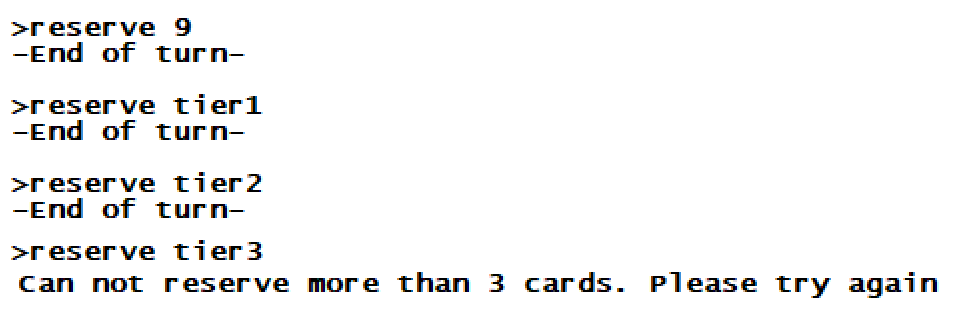
* If gems are available for double picking
* If one or more gems stack is empty
* If player picks gold gems
* If player enters wrong gem code (J does not exist)

As seen above there are 4 R gems so double picking is doable.

W, R, and O are all available as well.

1. **Reserving Development:**



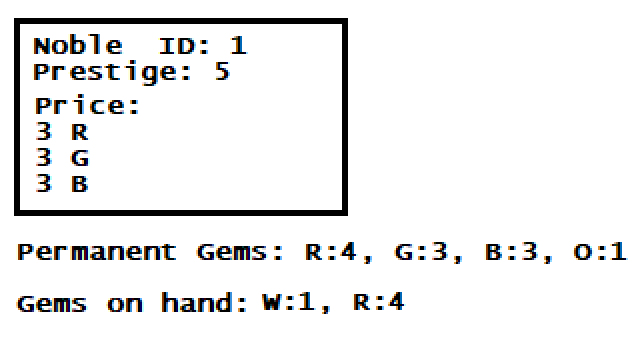


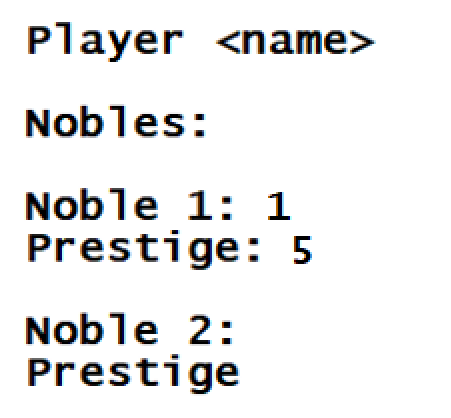
Reserving an existing card will put the card into the player’s reserved card stack. Player will automatically receive a Gold (E) gem if there are any.

Reserving tier1/tier2/tier3 will pick a random card from the selected tier card and store it in a player’s reserved stack (automatically revealing the card’s information).

A player can not have more than 3 cards reserved.

**Receiving Noble Visits:**



****

Player will automatically receive a noble’s visit, without having to purchase them.

**End Game:**

On the end of each round (after last player’s move), the game should check each player’s prestige. If one or more player has more >14 prestige, then the game ends.