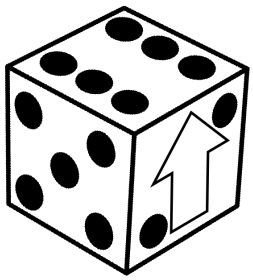
Tuesday, 10 September 2019

DiceUp Documentation & Plans



Application

Graphical User Interface

The GUI is based on **JavaFX** and will follow the model-view-controller scheme. The model will be made within **FXML**.

Board Layout

The *backgammon board* was introduced in a root container with **top-left** align (objects are printed horizontally beginning from right).

Following items were used inside the board:

- **VBox** - *LeftView*, *the left window of the board*
  - **HBox** - *TopLeftCols*, the *top left* window that displays 6 columns (*Col*##)
    - **VBox** - *Col*##, a column that displays the chips vertically
  - **VBox** - *GapLeft*, a space that has VGrow ALWAYS that fills space when resized
  - **HBox** - *RightLeftCols*, the *bottom left* window that displays 6 columns (*Col*##)
    - **VBox** - *Col*##, a column that displays the chips vertically
- **VBox** - *MiddleSector*, a middle divider that displays hit chips
  - **VBox** - *MiddleCol*, a column that displays the chips vertically
  - **VBox** - *MiddleCol*, a column that displays the chips vertically
- **VBox** - *RightView*, the right window of the board
  - **Same content as *LeftView***

Logic

The logic package of the game will include the barebones of the game play. The *GUI* package will communicate with this *GamePlay* package to retrieve status of the game.

***TODO*: DESIGN THE UML DIAGRAM!**

Class Structure

This package is designed to work with and without the GUI for future plans on *machine learning.*

Following classes were created in the package:

Tuesday, 10 September 2019

- **Game** - wrapper that holds the status of the current round
- **Player** - holds players information
  - **AI** - extends Player that behaves as an *interface* for **AI** methods
- **Dice** - generates a number in 1:6 and stores it
- **Board** - stores 4x6 **Column**s and retrieves current positions of individual chips
- **Column** - stores **Chip**s
- **Chip** - stores an identifier for ownership