Concordia University
Department of Computer Science and Software Engineering
Advanced Programming Practices
SOEN 6441 – Winter 2019

Build -1 Presentation

Krunal Jagani	4005 6939
Siddhant Arora	4008 5538
Jatan Gohel	4007 8112
Harsh Vaghani	4008 4099

Guided By
Dr. Rodrigo Morales

Content :

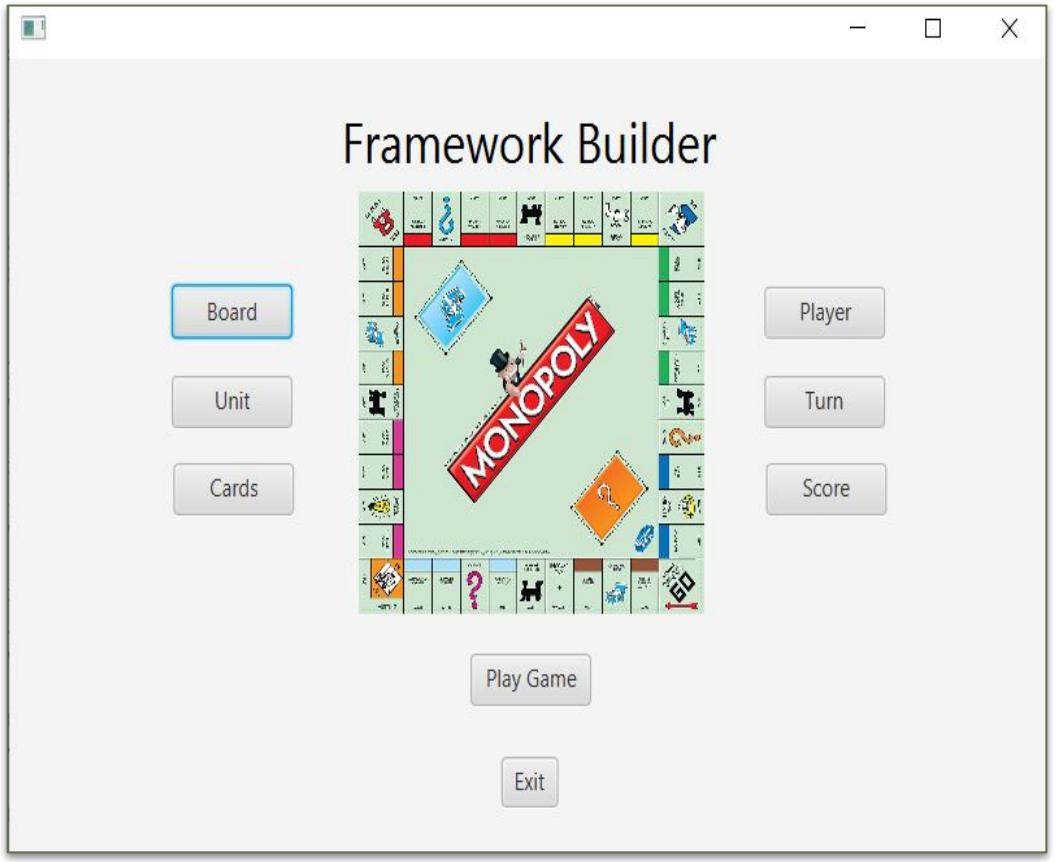
- Introduction
- View
- Models
  - Detailed overview
  - UML
- Controller

Introduction

- Goal of the first build was to make a generalized framework that can support any board game in future build
- Going to implement Monopoly game in our future build by using this framework
- Played and Analyzed total 3 games to extract out common features among them and then started design phase.
- We have implemented game framework using standard MVC(MVC-I specifically)
- Model and controller documentation using standard JavaDOC
- More than 10 test cases to test the veracity of different aspects of our code using Junit5
- We are going to explain build of each module by comparing its working in both Monopoly and RISK game.

View (MVC- I)

- **MenuScreen.Java**
- For ease of navigation we have included menu screen that helps us navigate different modules to configure.
- We have used JavaFX to visualize core components that need to be accomplished.
- Although it is not a finalized view of the game but it helped us in visualizing sequence of object creation and object passing among different modules and many other data flows.



Models

- Board model
- Tile model (Submodule of Board model)
- Unit model
- Score model
- Turn model
- Player model
- Dice
- Card

Board Module

- Board module had to be so generic that it can support any board game out there. **(RISK ,Monopoly or even CHESS…)**
- For Instance,
  - 1. In case of Risk it consists of Continents and armies
  - 2. In case of Monopoly each tile represents individual plots that have their own properties.
- Up on entering desired height and width of the desired board this module generates a grid of tiles of size height*width .
- It has methods to create connection between tiles, accessing individual tiles.

(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)	(0,7)	(0,8)	(0,9)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)	(1,9)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	(2,7)	(2,8)	(2,9)
(3,0)	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)	(3,7)	(3,8)	(3,9)
(4,0)	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)	(4,7)	(4,8)	(4,9)
(5,0)	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)	(5,8)	(5,9)
(6,0)	(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)	(6,7)	(6,8)	(6,9)
(7,0)	(7,1)	(7,2)	(7,3)	(7,4)	(7,5)	(7,6)	(7,7)	(7,8)	(7,9)
(8,0)	(8,1)	(8,2)	(8,3)	(8,4)	(8,5)	(8,6)	(8,7)	(8,8)	(8,9)
(9,0)	(9,1)	(9,2)	(9,3)	(9,4)	(9,5)	(9,6)	(9,7)	(9,8)	(9,9)

Board Module (Cont.)

Individual cell represents a tile on the board

Tile

It represents individual cell in a board metrics of the size Height and width passed as a parameter.

It is storing information related to individual tile such as,

- Tile name
- Internal value of tile
  - For RISK, it can be Army capacity of individual country
  - For monopoly, it represents Buying value of a plot
- List of units associated with individual tiles
  - For RISK, it can be Armies , types of armies such as Soldiers, cavalries, tanks etc.
  - For monopoly, it can be hotels, houses etc.

Tile (Cont.)

- List of its neighbours.
- Current Player
  - Risk -current player who attacks on this tile
  - Monopoly - Current player who lands on this plot
- Main player (Owner of the Tile)
- List of neighbours
  - It supports more than 2 tile connection in case of risk and exactly 2 connections in case of Monopoly

Unit

- Name of unit
- Property of the unit
- Description of the unit
- Tile where this particular unit is associated
  - (Ideally not necessary in RISK but in MONOPOLY Hotel, house can be associated with a particular tile)
- Amount associated with the unit
  - Risk - Soldier has 10 points, Cannon has 30 points , cavalry has 20 points
- During initial configuration of unit the amount entered is total number of that particular unit
  - For example, If the hotel unit is mentioned with 30 amount, there can be no more than 30 hotels in the entire game.

Player

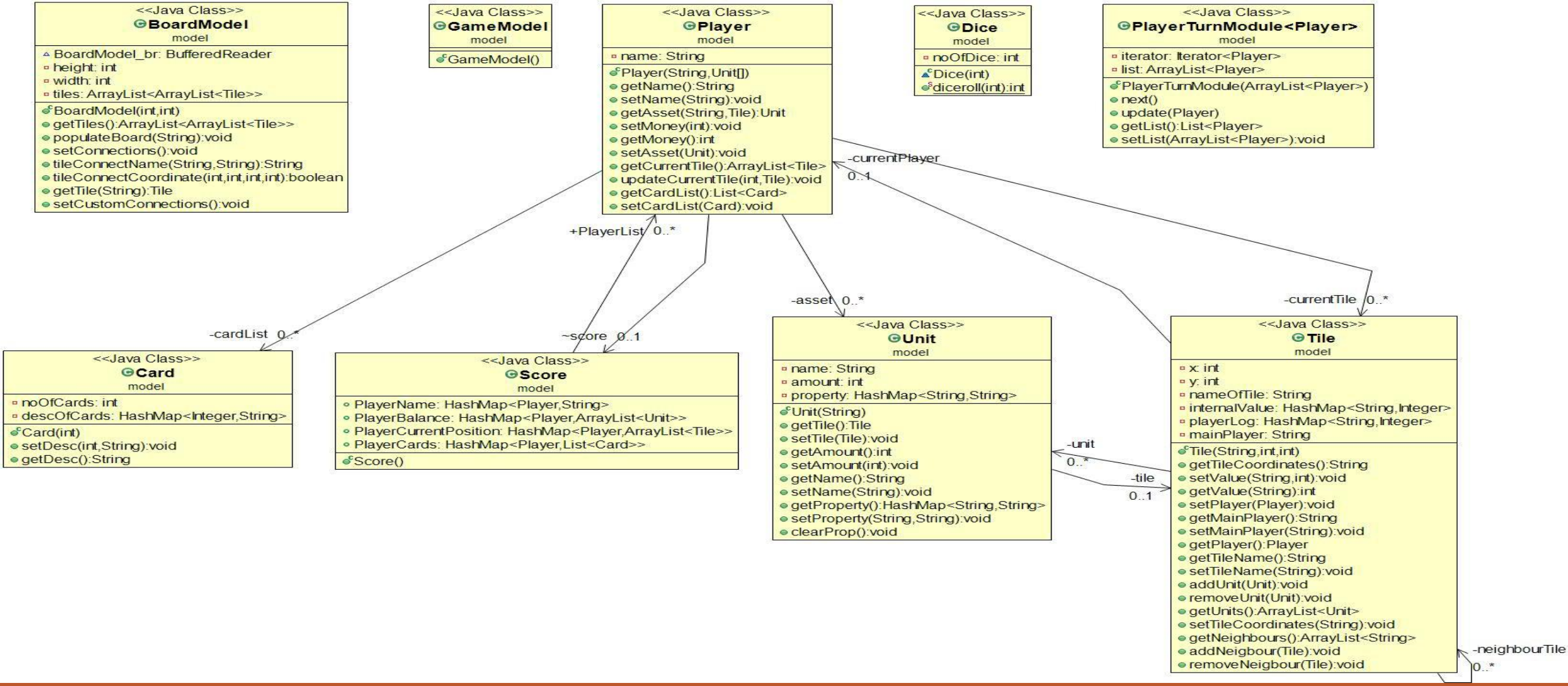
- Information such as
  - Player name,
  - current tile of player
  - balance,
  - Unit possession
  - Card possessions
- Uses List to store current tile of player because,
  - In RISK, a player can be at multiple tiles at the same time
  - In Monopoly, it can be at one tile at a particular time.
- Unit module is also implemented in List data structure since a player can own the same unit(For instance - Hotel) on multiple tiles.

Turn

- Works as a player factory which uses iterator to give sequential player objects as per their turn.
- Next method gives object of Player next in line in round robin fashion.
- Update method helps us in resetting the iterator.
- It can be used in case if we want to skip turn of any player.

Score :

- Contains data structures to store different measures associated with Players :
  - Players and their respective names,
  - unit amount,
  - their current tile
  - Cards owned by players
- It keeps track of above listed attributes whenever the associated function in the player module is called.



Model class diagram

Controller (Start up)

- Before the playable game begins, view is popped up so that the user can configure every module of the framework.
- Once all the modules are initiated and configured as per the game the individual objects are passed to the game controller.
- The game module does not begin unless all modules are initiated.

Controller (Gameplay)

- Two phases have been added namely reinforcement and fortification.
- Reinforcement : All the players receive equal amount of units that can be specified as per the game
  - (For example - Armies in RISK and Money in MONOPOLY)
- Fortification : Units can be exchanged according to the game rules.
  - In case of Monopoly, money can be exchanged between players, distributed among players or exchanged with bank object)
  - In case of Risk different kind of army units can be exchanged between players according to the strategy applied.

Controller class diagram