

# GapBammon

Madis Pink  
Mikk Pavelson  
Uku Loskit  
Lauris Kruusamäe

# **1 Project Schedule**

## **08.10**

- 1 Write schedule
- 2 Acquire domain knowledge (backgammon rules and playing experience)
- 3 Storyboard
- 4 Write project requirements and aims
- 5 Infrastructure setup

## **09.10**

- 1 Create class diagram
- 2 Write code

## **11.10**

- 1 Write code
- 2 Test
- 3 Write repository / code outline
- 4 Documentation cleanup

## **12.10**

- 1 Release
- 2 Submit assignment

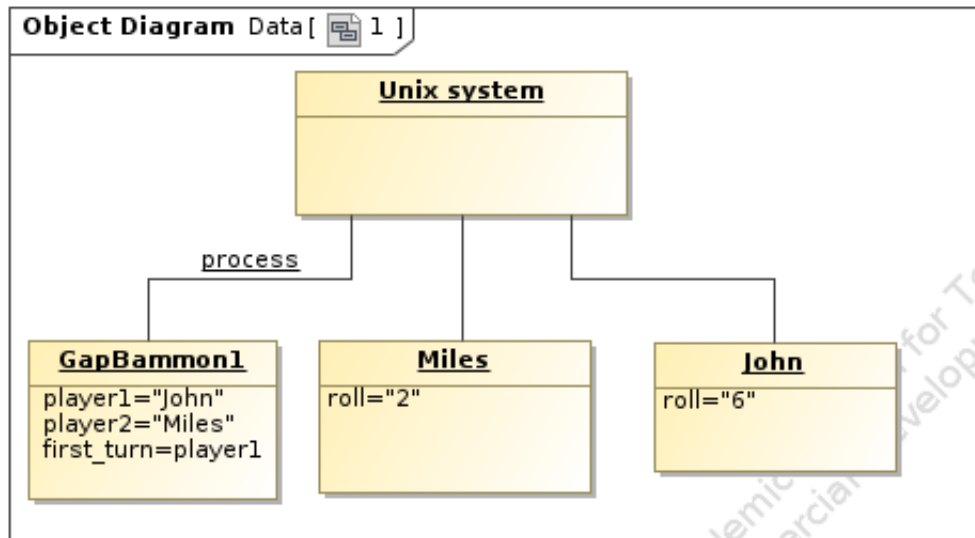
## 2 Storyboard

### Background

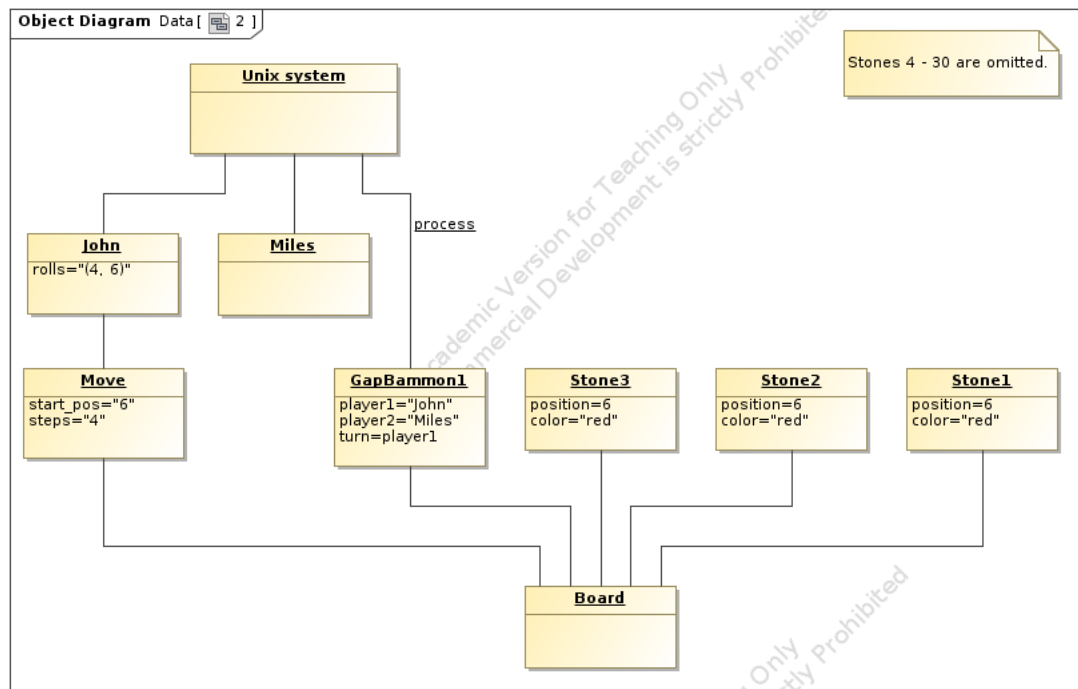
John and Miles are long-time friends and want to play backgammon together. Unfortunately they both live in different parts of the city and have access to a teletype terminal. So, they turn to GapBammon...

I John and Miles start a game of CLI backgammon by running a command in a terminal. John inputs his name first and becomes player1, Miles is upset because he is forced to be player2. Miles enters his name.

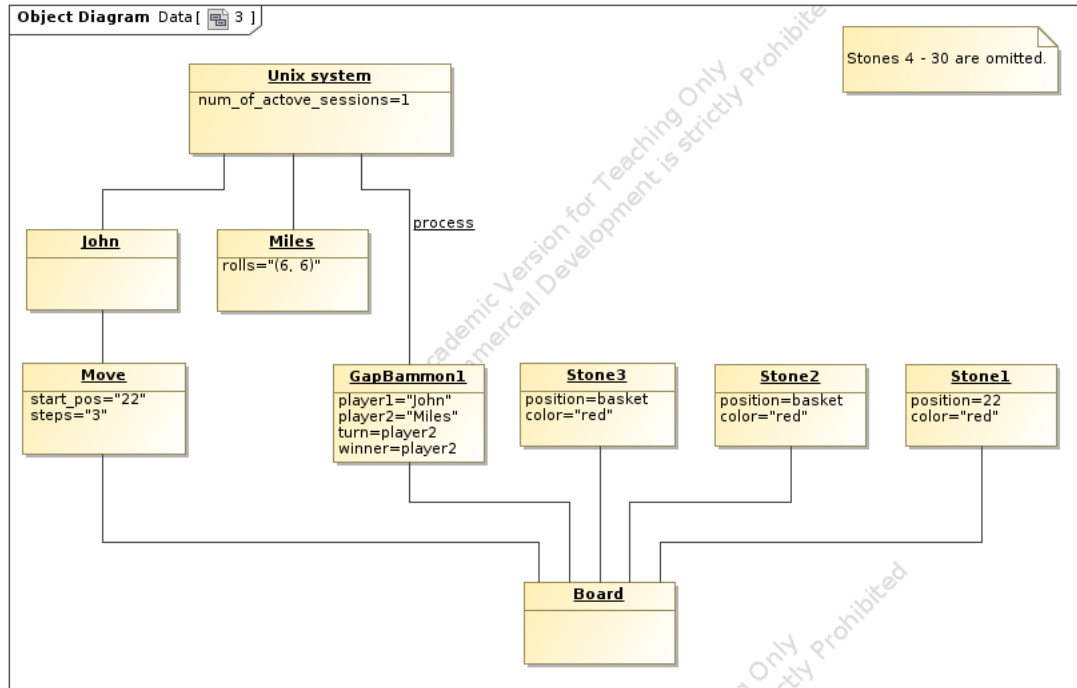
GapBammon rolls the dice for both players, John rolls a 6, Miles rolls a measly 2. This means that John is better again and gets to go first.



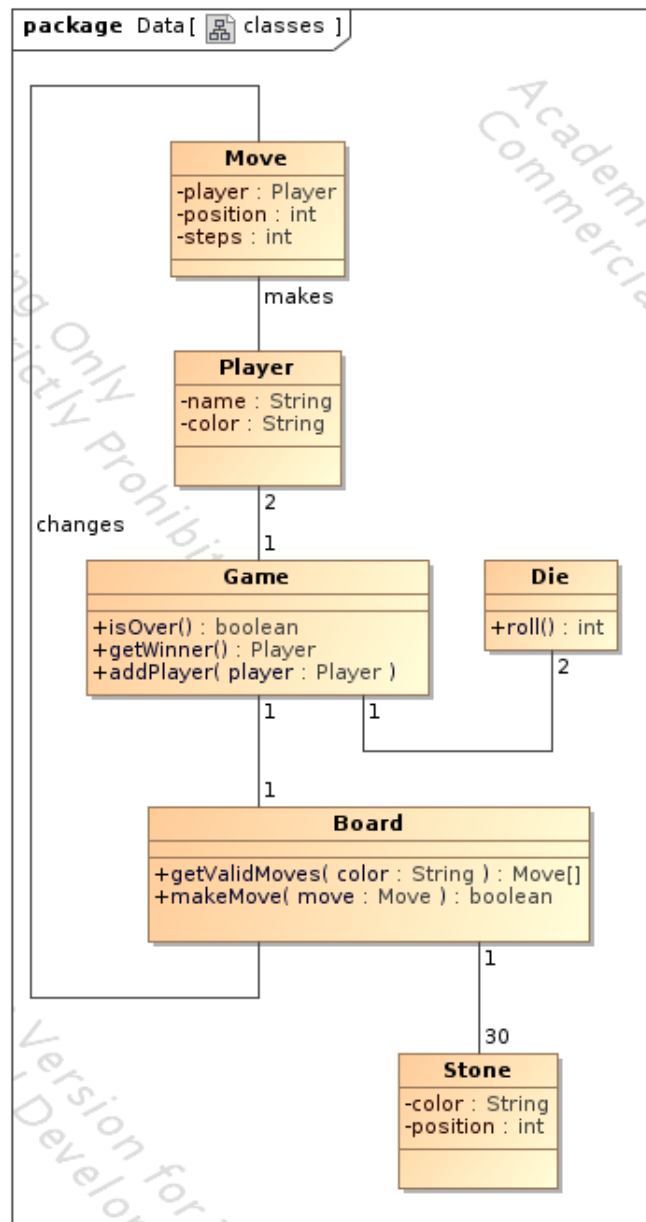
II Miles and John have started the game. John needs to make a move against Miles. A backgammon board is displayed on the terminal screen (ASCII art). The game rolls dice for John with the outcome of 4 and 6. Now John can see a list of all available moves and select two of them consecutively to finish his turn. And so he does. The board is printed again to reflect the new game state.



III 30 minutes of unearthly action follow. Suddenly, Miles is ahead and needs to make a final move. All but one of his stones have found their way into the basket. The game rolls doubles (two sixes) for Miles, which he easily turns into a win. The game ends and declares Miles the victor. John ragequits.



### 3 Class Model



## 4 Requirements

### Non-functional Requirements

- the application should not take longer than a second to process user input and display the results:
- the application should run on all operation systems that ruby 1.9.3 supports
- the application must not crash on more than 1 % user inputs
- the application must be packaged as a ruby gem

### Functional Requirements

- *Application*
  - on startup the game should ask for both players' names
  - the game must have a command-line interface
  - the game must declare a victor when a game has ended
  - the game must offer an option for rematch after the game has ended
- *Game rules*
  - the game must randomly roll the dice to determine the starting player
  - the game must support all standard backgammon rules
  - the game must have a backgammon board with two rows of triangles in alternating color and each triangle is numbered 1-24 starting with the lower-right triangle.
  - the game must support representing stones of two colours - red and black
  - the game must start with a standard backgammon layout:
    - \* 2 red stones on triangle 1
    - \* 5 red stones on triangle 12
    - \* 3 red stones on triangle 17
    - \* 5 red stones on triangle 19
    - \* 2 black stones on triangle 24

- \* 5 black stones on triangle 13
  - \* 3 black stones on triangle 8
  - \* 5 black stones on triangle 6
- the game must roll the dice for the player when it is player’s turn.
- valid moves are determined by the player’s dice roll according to backgammon rules.
- *User interface*
  - the user must see the following information at the beginning of his/her turn:
    - \* the current board and stones layout
    - \* the outcome of the random dice roll
    - \* the list of valid moves for the player
  - the user must be able to input two positive integers in the form of  $\langle \text{triangle}_i \text{ } \text{steps}_i \rangle$  to move the topmost stone on the specified triangle by the specified amount of moves
  - the game must print out the declared victor when the game ends



## 5 Minutes from Meeting with Customer

Customer: I have a couple of suggestions that could really improve Gap-Bammon.

Firstly, there currently is no match system for keeping track of how many points a given player has gathered. This brings us to another possible feature there is no doubling cube, which is a standard component of backgammon. Additionally, we managed to find some bugs. Namely, the game ended up in an infinite loop.

The customer also pointed out some possible improvements to the user interface. The visual board could be improved to show stones that have been put on the bar and stones that have reached the basket. At the moment, users have to keep track by reading this information from lines below the board.

Furthermore, the board is really basic and no way to easily differentiate between quadrants. This should definitely be fixed.