

Mancala Board Game

Introduction

Through this article, I am going to explain the technology stack which I used to create Mancala web game, I will explore some code snippets and the internal system architecture. Also, you will find a demo video about how you can play.

For more information about Mancala board game <https://en.wikipedia.org/wiki/Kalah>.

1. Project Environment

- Spring Tool Suite 3.8.2.RELEASE
- JDK 1.8
- Apache Maven 3.3.9
- WebSocket (STOMP)

2. Project Package Structure

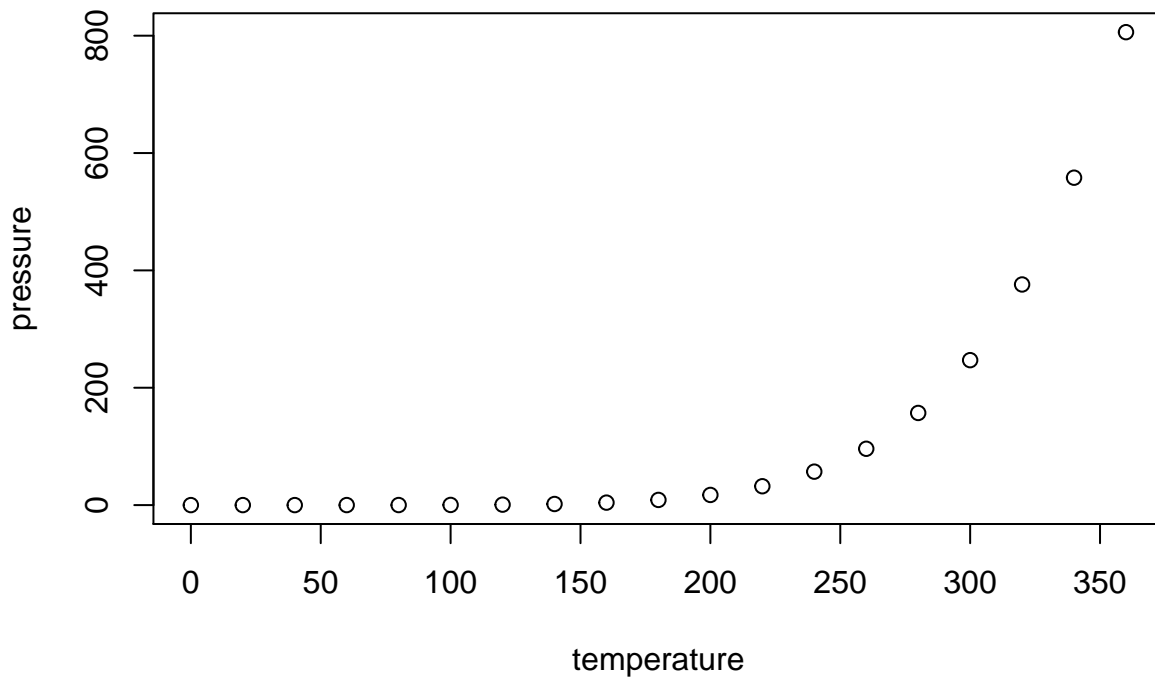
- src/main/java
 - com.backbase.mancala
 - * MancalaApplication.java
 - * MancalaApplication.java
 - com.backbase.mancala.controller
 - * ChatController.java
 - * GameController.java
 - com.backbase.mancala.domain
 - * Pebble.java
 - * Pile.java
 - com.backbase.mancala.dto
 - * GameBoard.java
 - * GameStatus.java (Enum)
 - * Message.java
 - * Winner.java (Enum)
 - com.backbase.mancala.service
 - * IBoardGame.java
 - * MancalaGame.java
- src/main/resource
 - static
 - * app.js
 - * index.html
 - * main.css
- src/main/test
 - com.backbase.mancala
 - * MancalaApplicationTests.java

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.    : 2.00
##  1st Qu.:12.0    1st Qu.: 26.00
##  Median :15.0    Median : 36.00
##  Mean   :15.4    Mean     : 42.98
##  3rd Qu.:19.0    3rd Qu.: 56.00
##  Max.   :25.0    Max.     :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.