Table of Contents

[Introduction 1](#_Toc6518540)

[Architecture 1](#_Toc6518541)

[Assumptions 1](#_Toc6518542)

[Technologies 2](#_Toc6518543)

[Design 2](#_Toc6518544)

[Environment 2](#_Toc6518545)

[Extendibility 2](#_Toc6518546)

[Folder Hierarchy 2](#_Toc6518547)

[Usage 3](#_Toc6518548)

[Input 3](#_Toc6518549)

[Screenshot 3](#_Toc6518550)

# Introduction

This implementation is a 2-player turn-based game of well know Mancala board game, which users can enjoy via shared screen.

# Architecture

## Assumptions

* Users share the screen and plays together.

## Technologies

* Spring Boot: Main orchestrator
* Tomcat: Using to serve the game
* Junit: Testing purposes

## Design

Spring boot used for to serve single page web application. Rest api preferred while interaction between frontend and backend side. Design can be split under two main steps;

• Front-end: Visualization part. This part used for to prepare and post the current board to the backend and reflect the results of the rules to the player’s boards.

• Back-end: Operation part. This part is used for consuming the actual board state and apply rules to the board data. There is no persistency layer implemented to make it stateless.

## Environment

* This application designed to run on embedded tomcat.

## Extendibility

* New rule can be added easily
* Code is well documented and tried to be as clean and easy to understand as possible

# Folder Hierarchy



# Usage

## Input

Clone to your local repository: git clone https://github.com/rdemirkoparan/mancala.git

Change directory: cd mancala/

Compile: mvn compile

Test: mvn test

Run: mvn spring-boot:run

Play: open you favorite browser and browse http://127.0.0.1:8080/index.html

## Screenshot



