Software Item Description

For

BIQ2Passover Project

Created By:

Marina Shabanov

Yuval Lev Naor

Shahar Tal

Contents

[1. Multi-Thread Support Overview 3](#_Toc11230060)

[2. Approach of the Solution 3](#_Toc11230061)

[3. Planned Changes 4](#_Toc11230062)

# Multi-Thread Support Overview

We are adding new feature to the project – Multi Thread Support.

This feature will introduce new functionality that will allow to run few players + mazes in parallel. The user will supply three parameters through the command line:

* -mazes\_folder <mazes\_folder> 🡪 where to find the mazes files
* -players <player\_package> 🡪 which package contains the players
* (Optional) -threads <threads\_number> 🡪 the number of threads to run with

Once the parameters are supplied, the project will make a Cartesian product of the players and valid mazes and run all the pairs with the threads supplied (if not supplied, will be running in serially in 1 thread).

# Approach of the Solution

Our solution will contain new main that will support the new command line arguments, along with new class that will parse those arguments and validate them.

Once we get all the mazes and players, we will conduct validation on the mazes to get only the valid ones (according to the validations already defined before).

We will add a new class that will run the pairs of players and mazes in threads. This class will be called from a new method in GameLoader class that will handle a Thread Pool for supporting running in Multi-Threads.

# Planned Changes

Following are the planned changes to support the new feature:

* New Classes:
  + ***MazeGameMultiThread*** 🡪 new main to support new command line arguments
  + ***CommandLineParser* 🡪** new class that parses the new command line arguments and validates them
  + ***GameManagerRunner*** 🡪 new class that implements Runnable
  + ***GameManagerThreadPool*** 🡪 new class that runs game managers in multi thread with thread pool
  + ***GameManagerSingleThread*** 🡪 new class that runs game managers in 1 thread only
  + ***GameManagerFactory*** 🡪 new class that decides on game manager strategy (between ***GameManagerThreadPool*** and ***GameManagerSingleThread***)
  + ***GameResult*** 🡪 new class that saves result of a player in a game
* Classes to change:
  + ***GameLoader***
    - Add new class member, list of mazes
    - Add new class member, list of game managers
    - Add new method, startGames that gets list of players and number of threads
    - Split validateAndStartGame method to two new methods, validateArguments and parseMaze
    - Add new method, getMazesNumber
    - Add new method, parseMazes that gets list of mazes and runs parseMaze
    - Refactor start method
    - Add new method, printResults that prints table of maze / player results
  + ***MazeGame***
    - Refactor due to changes in GameLoader class
  + ***GameManager***
    - Add new constructor that gets player package and maze
  + ***PlayerRuleSet***
    - Move initialization from constructor to member declaration and remove constructor
* Tests to add:
  + ***GameLoaderTest*** (tests for new startGames, parseMaze and parseMazes methods)
  + ***CommandLineParserTest*** (new class)
  + ***GameManagerRunnerTest*** (new class)
  + ***GameManagerTest*** (test for new constructor)
* Tests to change:
  + ***GameLoaderTest*** (due to changes in class)