Team Members:

* S. Shayan Daneshvar

He is 20 years old and is studying Computer Engineering at K.N. Toosi University of Technology.

started programming with C++ language when he was 15 years old, currently he is familiar with

VB6, C, C++, C # , Java and Python.

Interested in complex subjects and Artificial Intelligence.

* Houra Keshavarz

She is 29 years old and holds MS. Pure Mathematics degree, got familiar with Java during her freshman year.

* Stories:

Story 1: Implementing a basic abstraction of the game which included player movements

Story 2: Adding walls and their functionality to the game.

Story 3: Adding validator to validate players’ acts

Story 4: Adding Random Player and a simple AI and Tournaments

* Scenarios:

Scenario 1.1:

Players win when they reach the other side of the board.

Scenario 1.2:

When players put walls they remove the connection between two pairs of cells, hence we

implemented the game board using a simple Graph from the JGraphT library.

Scenario 2.1:

When player puts a wall the connection within two nodes is ruptured.

Scenario 2.2:

When a player trap him/herself or the other player the blockage should be deduced, hence

A Breadth-First-Search Algorithm was used in order to Detect such Blockages.

Scenario 3.1:

When players choose their next action, the action must be validated, Therefore the Validator Class was Added and different situations were considered;

Scenario 4.1:

Easy difficulty includes a random player.

Scenario 4.2:

A tournament comes with a set of games, e.g. 3 games.

Scenario 4.3:

Medium difficulty includes a player which is rather intelligent, hence we used Dijkstra algorithm

For finding the best way.

UML:

