Names

Bishoy Nader Fathy (21)
Marc Magli Kamel (55)

AI Assignment 2 : RISK

24th November 2018

OVERVIEW

Implementing Simplified Version of RISK with various Agents which are:

- Non AI Agents
    - Human Agent
    - Passive Agent
    - Aggressive Agent
    - Pacifiest Agent
- AI Agents
    - Greedy Agent
    - A* Agent
    - Real Time A* Agent

Project Packages

We divided the project to this packages:

- Views: contains the fxml files of the UI.
- controllers: contains JavaFX controllers that connect UI with models.
- agents: contains implementation of the seven agents required.
- core: contains the logic of transition between states, adding armies, attacking and so on.
- models: contains the data classes of the vertex, board and so on.

Assumptions To be Clarified:

1. We assumed that each vertex initial army is a part of the input file.
2. We assume that both the 2 Armies Bonus and the Bonus from owning partitions must be added to the same vertex at the beginning of each turn,
3. At each turn the agent is allowed to attack one vertex only.

Data Structures Used:

1. We used linked Lists, sets and maps widely in the core and data classes.
2. We used Priority Queue in AI Agents to get state with best cost or heuristic value.

How each Algorithm Works:

Greedy Agent: Searches for the state with the best heuristic value and chooses to expand it first till it reaches the goal state and then returns to find the transition that opened this path.

A* Agent: Searches for the state with the best sum of heuristic value and actual cost and chooses to expand it first till it reaches the goal state and then returns to find the transition that opened this path.

Real Time A* Agent: Same as the A* but it expands states up to a certain depth and then chooses the best of the leaf nodes as the goal state and try to reach it.

HEURISTIC FUNCTION

For the AI Agents we used the following heuristic function:

h(state) = Vopp + Sigma (Vp * NBp * Ap,opp / Ap) For every Partition, where

Vopp: Number of Opponent Vertices

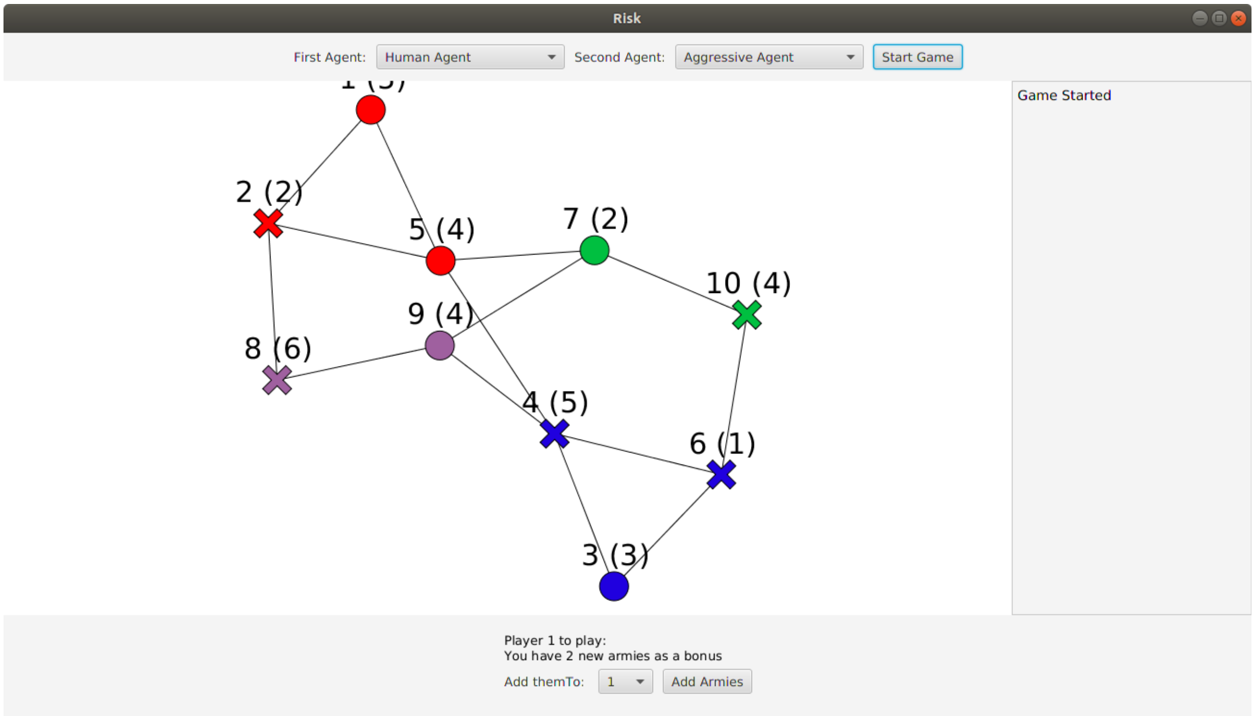
Vp: Number of Vertices in this Partition

NBp: Normalized Bonus of this Partition

Ap,opp: Opponent’s Total Armies in this Partition

Ap: Total Armies in this Partition

SAMPLE RUNS



Risk

First Agent: Human Agent

Second Agent: Aggressive Agent

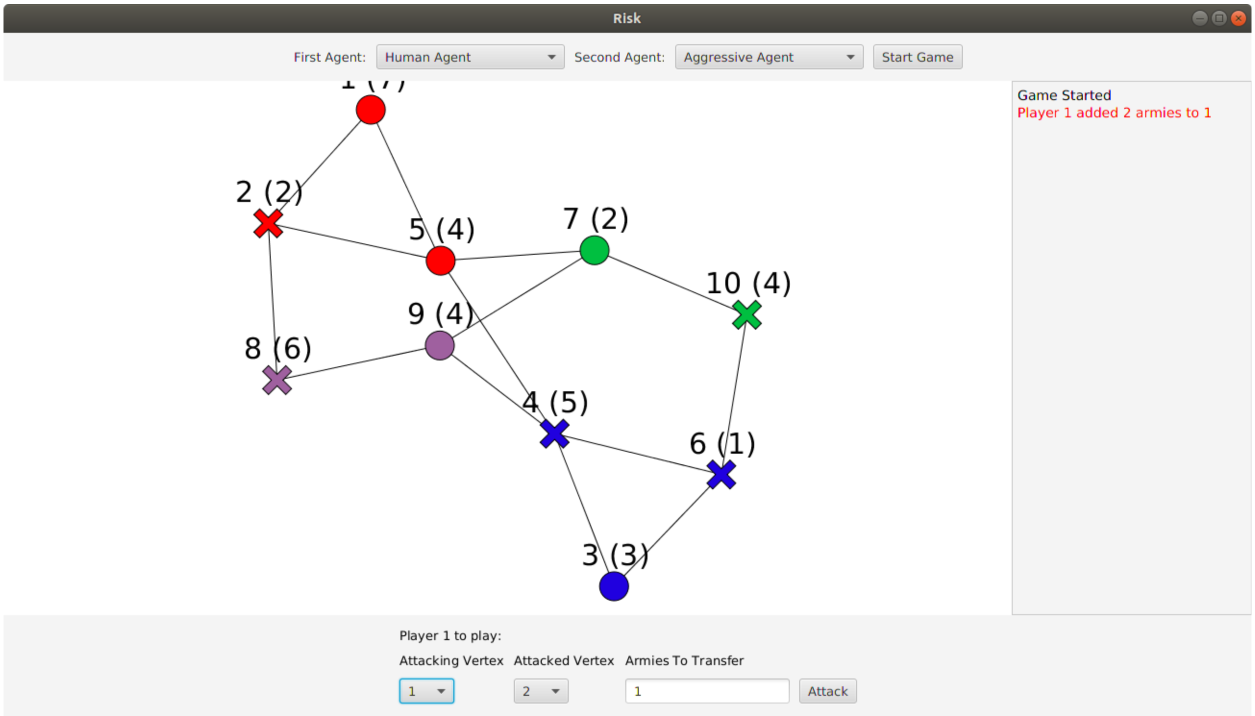
Start Game

Game Started

Player 1 to play:
You have 2 new armies as a bonus

Add themTo: 1

Add Armies



Risk

First Agent: Human Agent

Second Agent: Aggressive Agent

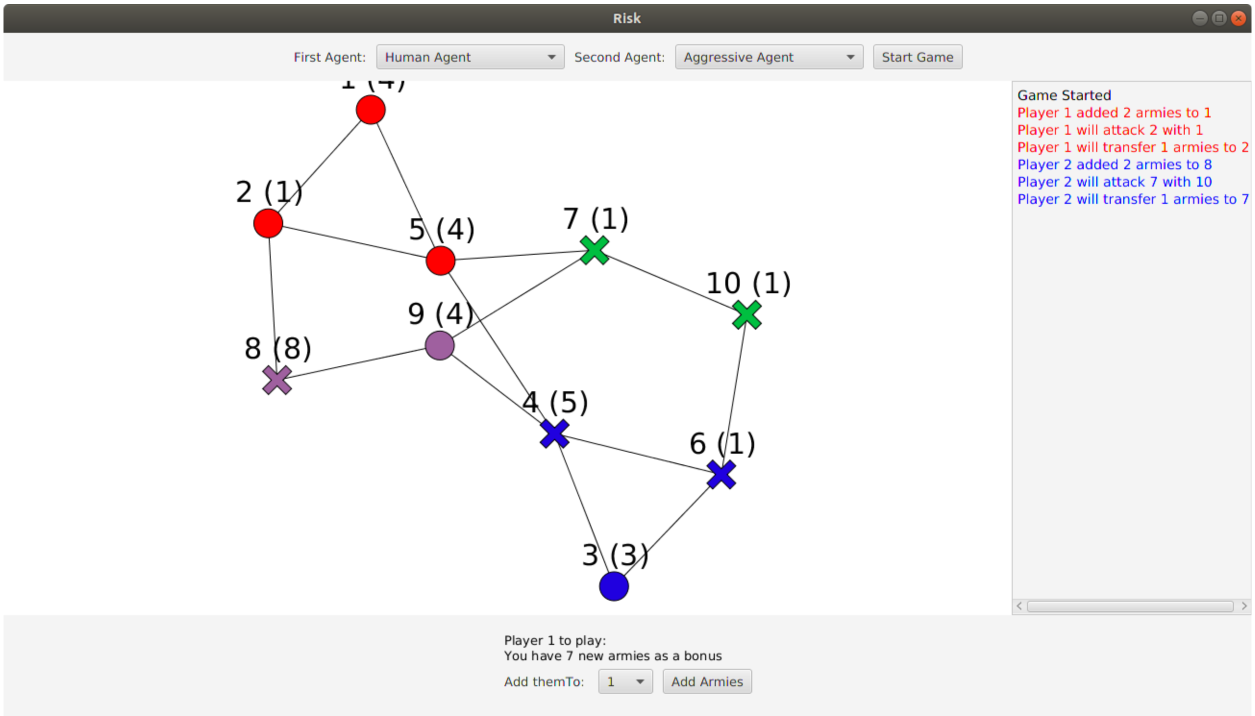
Start Game

Game Started
Player 1 added 2 armies to 1

Player 1 to play:

Attacking Vertex	Attacked Vertex	Armies To Transfer
1	2	1

Attack



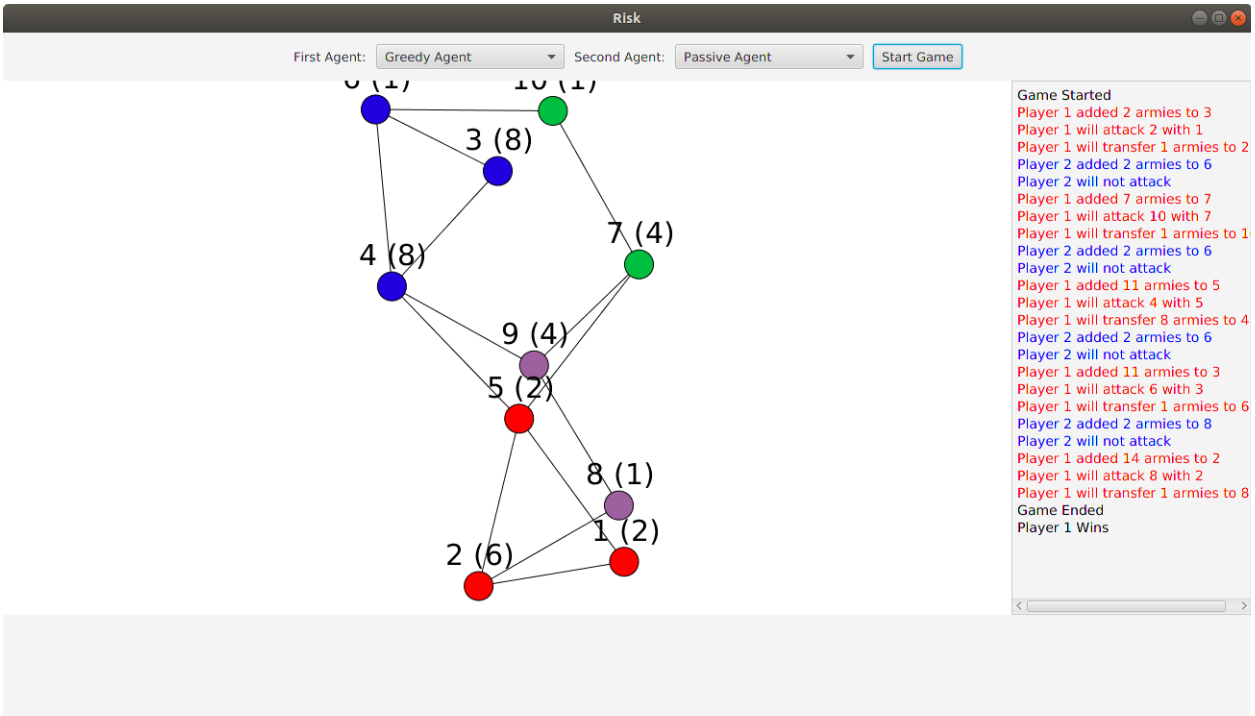
First Agent: Human Agent

Second Agent: Aggressive Agent

```
Game Started
Player 1 added 2 armies to 1
Player 1 will attack 2 with 1
Player 1 will transfer 1 armies to 2
Player 2 added 2 armies to 8
Player 2 will attack 7 with 10
Player 2 will transfer 1 armies to 7
```

Player 1 to play:
You have 7 new armies as a bonus

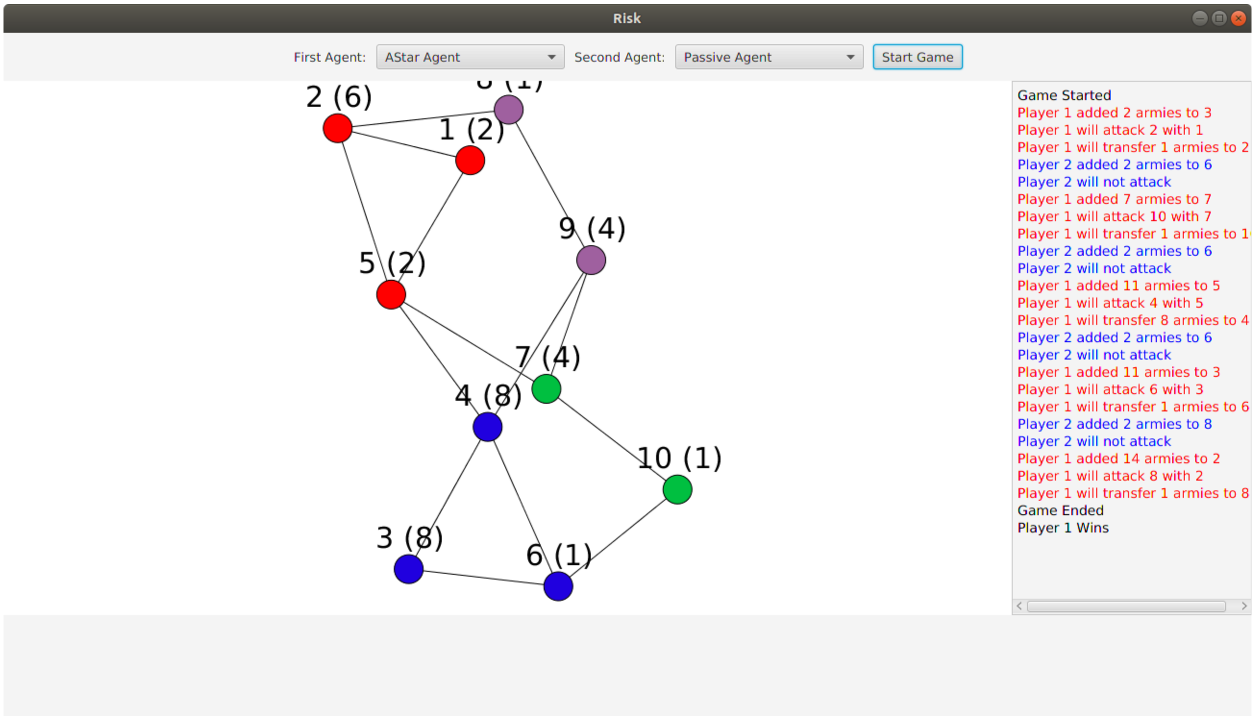
Add themTo: 1



First Agent: Greedy Agent

Second Agent: Passive Agent

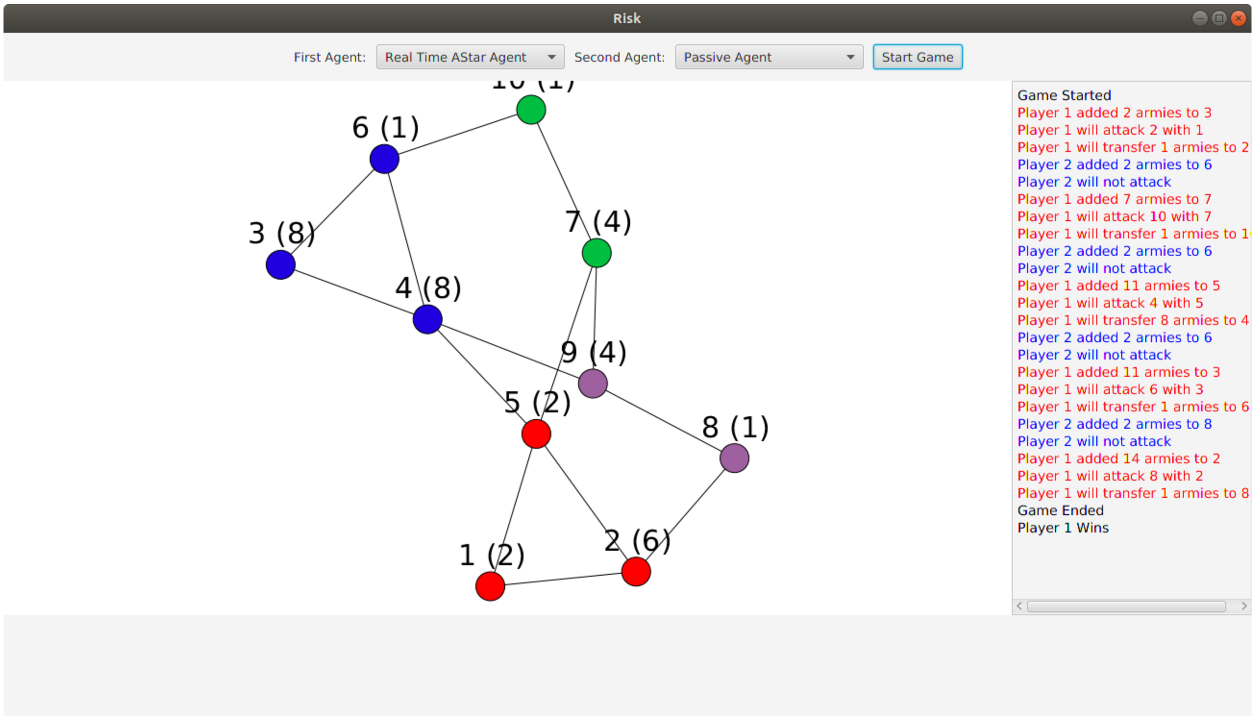
```
Game Started
Player 1 added 2 armies to 3
Player 1 will attack 2 with 1
Player 1 will transfer 1 armies to 2
Player 2 added 2 armies to 6
Player 2 will not attack
Player 1 added 7 armies to 7
Player 1 will attack 10 with 7
Player 1 will transfer 1 armies to 1
Player 2 added 2 armies to 6
Player 2 will not attack
Player 1 added 11 armies to 5
Player 1 will attack 4 with 5
Player 1 will transfer 8 armies to 4
Player 2 added 2 armies to 6
Player 2 will not attack
Player 1 added 11 armies to 3
Player 1 will attack 6 with 3
Player 1 will transfer 1 armies to 6
Player 2 added 2 armies to 8
Player 2 will not attack
Player 1 added 14 armies to 2
Player 1 will attack 8 with 2
Player 1 will transfer 1 armies to 8
Game Ended
Player 1 Wins
```



Risk

First Agent: AStar Agent Second Agent: Passive Agent Start Game

```
Game Started
Player 1 added 2 armies to 3
Player 1 will attack 2 with 1
Player 1 will transfer 1 armies to 2
Player 2 added 2 armies to 6
Player 2 will not attack
Player 1 added 7 armies to 7
Player 1 will attack 10 with 7
Player 1 will transfer 1 armies to 1
Player 2 added 2 armies to 6
Player 2 will not attack
Player 1 added 11 armies to 5
Player 1 will attack 4 with 5
Player 1 will transfer 8 armies to 4
Player 2 added 2 armies to 6
Player 2 will not attack
Player 1 added 11 armies to 3
Player 1 will attack 6 with 3
Player 1 will transfer 1 armies to 6
Player 2 added 2 armies to 8
Player 2 will not attack
Player 1 added 14 armies to 2
Player 1 will attack 8 with 2
Player 1 will transfer 1 armies to 8
Game Ended
Player 1 Wins
```



Risk

First Agent: Real Time AStar Agent Second Agent: Passive Agent Start Game

```
Game Started
Player 1 added 2 armies to 3
Player 1 will attack 2 with 1
Player 1 will transfer 1 armies to 2
Player 2 added 2 armies to 6
Player 2 will not attack
Player 1 added 7 armies to 7
Player 1 will attack 10 with 7
Player 1 will transfer 1 armies to 1
Player 2 added 2 armies to 6
Player 2 will not attack
Player 1 added 11 armies to 5
Player 1 will attack 4 with 5
Player 1 will transfer 8 armies to 4
Player 2 added 2 armies to 6
Player 2 will not attack
Player 1 added 11 armies to 3
Player 1 will attack 6 with 3
Player 1 will transfer 1 armies to 6
Player 2 added 2 armies to 8
Player 2 will not attack
Player 1 added 14 armies to 2
Player 1 will attack 8 with 2
Player 1 will transfer 1 armies to 8
Game Ended
Player 1 Wins
```

PERFORMANCE MEASURE

$$P = (f * L) + T$$

L: The number of turns it takes the agent to win the game,

T: The number of search expansion steps performed by search algorithm.

For Greedy Agent:

L = 5, T = 6

At F = 1 -> P = 11

At F = 100 -> P = 506

At F = 10000 -> P = 50006

For A* Agent:

L = 5, T = 7

At F = 1 -> P = 12

At F = 100 -> P = 507

At F = 10000 -> P = 50007

For Real Time A* Agent:

L = 5, T = 7

At F = 1 -> P = 12

At F = 100 -> P = 507

At F = 10000 -> P = 50007