**A. Team Members and Roles**

Vipul Kohli: UI, gameplay

Andrew Socha: AI communication, AI updates, move validation

**B. Team Chair**

Vipul Kohli

**C. Prioritized List of Functionality**

1. Display the game board

2. Communication between the game engine and AIs

3. Enact moves submitted by the AIs

4. Collision Detection

5. Update the test AI to understand “damage”

6. Move validation (no cheating!)

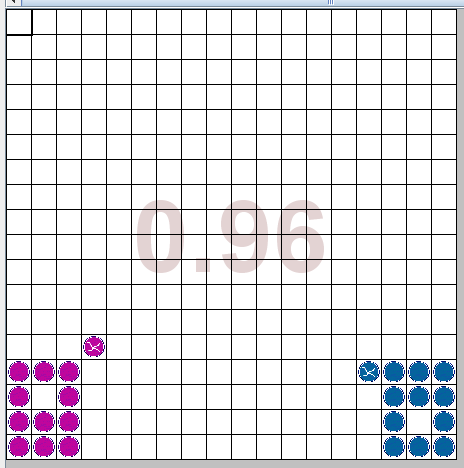
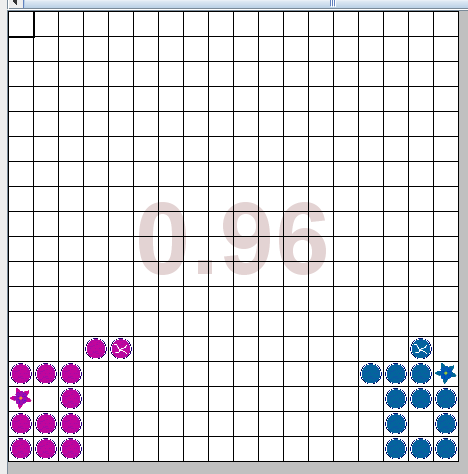
7. Victory conditions

8. Handling invalid moves, no AI response, invalid AI URL,… anything else game-breaking

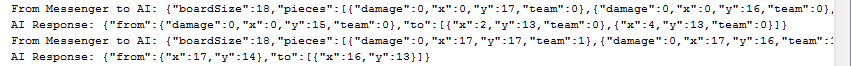
9. Game board customization (board size, number of pieces, etc.)

**D. Test Results**

The following is a test to see if the board updates properly based on data sent by the AIs. You can see the data sent from the game engine to each of the AIs, as well as each of their responses.



Board Before Board After

Game Engine and UI communication:

**E. Data Formats**

From HalmaMessenger to AIs:

{

"boardSize":18,

"pieces":[{"x":0,"y":0,"damage":0},{"x":1,"y":1,"damage":1}…],

"destinations":[{"x":0,"y":0},{"x":1,"y":1}…],

"enemy":[{"x":0,"y":0,"damage":0},{"x":1,"y":1,"damage":1}…],

"enemydestinations":[{"x":0,"y":0},{"x":1,"y":1}…]

}

From AIs to HalmaMessenger:

{

"from":{"x":0,"y":0},

"to":[{"x":1,"y":1},{"x":2,"y":2}]

}

From HalmaMessenger to Official to CollisionAnalyst:

JSON from AIs to "[3, 3, 5, 5, 7, 7]SPLITSPLIT[10, 10, 10, 8, 8, 8, 6, 6]"

-Each array is one AI’s move

-Array is formatted: fromX,fromY,JumpX,JumpY,....ToX,ToY

From CollisionAnalyst to Official to GameBoard:

[3,4,5,0,9,8,1,1… 7,6,3,0]

-PieceX, PieceY, PieceDamage, PieceTeam (for the entire updated game board)

From GameBoard to Gridworld:

Piece Object ArrayList

**F. Project Repository**

<https://github.com/vipulkohli/JHalma>