**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**

**E. Data Formats**

To AIs from HalmaMessenger:

{

"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 "[-1, -2, 1, 2, 3, 4]SPLITSPLIT[-1, -2, 1, 2, 3, 4]"

-First array is x,y,damage,team

-Second and third array is fromX,fromY,JumpX,JumpY,....ToX,ToY

From CollisionAnalyst to Official to GameBoard:

[3,4,5,0,9,8,1,1]

-PieceX, PieceY, PieceDamage, PieceTeam (for each AI’s move)

From GameBoard to Gridworld:

Piece Object ArrayList

**F. Project Repository**

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