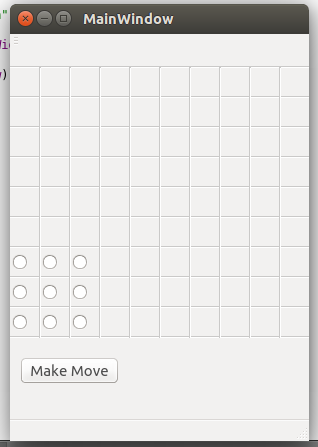
**November 18th Deliverable:**

1. Team members:
   1. Derek Phanekham
      1. Team chair, Older wiser programmer
   2. Johna Rutz
      1. General programmer, Task manager
2. Team chair:
   1. Derek Phanekham
      1. [dphanekham@smu.edu](mailto:dphanekham@smu.edu)
3. List of functionality:
   1. User Interface
      1. A GUI that shall offer the end user functionality only within the half of the game that they have control over. If an AI is connected, the only thing they shall be able to do is send instructions to the AI.
      2. When it is AI vs. AI the human at the computer only has one way to alter the state of the game through the make move button, the game engine controls everything else.
   2. Dual User
      1. Human vs. Human
         1. Humans shall be expected to be sitting side by side, interacting with the local GUI
      2. Human vs. AI
         1. The AI shall return proper formatting as specified below. Correct moves shall appear on the local GUI interface.
         2. The Human shall interact with the GUI
      3. AI. Vs. AI
         1. Both AI’s shall be supplied with board information and return proper formatting as specified below. Correct moves shall appear on the local GUI interface.
         2. If the AI does not return proper formatting or provides an incorrect move sequence, that AI’s turn will be “skipped” and the end-user shall be notified of the error. Moves shall appear on the GUI interface controlled by the human end user.
   3. Interference Rules Applied
      1. Penalties shall be given to users whom make move requests that result in the overlapping of two pieces.
   4. Timer
      1. Each user shall be given 30 seconds to make a move. If a move is not made within this time period the user shall be “timed out” and their turn will be skipped.
      2. At this point in time there shall be no direct penalty besides the loss of the current turn.
4. Results of tests for functional items:
   1. User Interface
      1. GUI as seen by the human end user with “make move” functionality



1. Finalized sample data formats:

a. Input:

i. Move Request:

{“from”: {“x”:2, “y”:4}, “to”: {“x”:9, “y”:10}}

ii. URL Request:

<https://www.website.com/halma.php>

b. Output:

i. Board information (http post):

{"boardSize": { "height": 9, "width": 18 },"yourBoard": { "pieces":[{ "x": 0, "y": 0 },{"x": 0, "y": 1 } ], "destinations": [{"x": 0,"y": 0 },{ "x": 0, "y": 3 },{ "x": 0, "y": 1}]}, "enemyBoard": {"pieces": [{"x": 0, "y": 0}, {"x": 0, "y": 1}], "destinations": [{"x": 0,"y": 0},{"x": 0, "y": 1},{"x": 0,"y": 1}]}}

1. Location of Repository: https://github.com/jrutz/halma3.0