CS-396 **Fall 2020**

Project 01: Checkers

Total Points: 100 pts Due: Friday, October 16, 2020

DIRECTIONS

Create a 2D board and pieces to play checkers with in the browser. The game should:

- Place pieces initially in the proper places
- Follow the rules of checkers, some important ones:
  - Normal moving only to one of the two places towards the opponent's side of the board
  - Allow jumping by moving two places towards the opponent's side of the board with an opponent's piece in-between and the opponent's piece is removed
  - Turning a piece into a "king" once it reaches the opponent's side of the board which then allows that piece to move backward or forward
  - ***Note:*** you do not need to implement multi-jumps, forced jumps, or game-over (these are extra credit)
- Use the JavaScript `click` event for the user to select one of their pieces and then select its destination
  - If the initial click location is not over a valid square, do nothing
  - After the initial click an indicator is drawn in all of the valid destination squares
  - If a square with an indicator is clicked, then the piece is moved appropriately (possibly removing jumped pieces and/or being promoted to a king)
  - If another valid square is clicked instead then the indicators are updated for the new valid piece
- Users must be able to distinguish whose turn it is, selected piece, potential moves, and kings vs men at a glance
  - Kings and men must be the same color (like real game), difference must be something besides color

Graphics requirements:

- The vertices and base colors for a single square, a single man, and a single king may be the only things sent in bulk to the GPU (and then transformed from there)
- Additionally, uniforms may be used to send transformation colors and information about adjusting the color of pieces from the base
  - *Hint:* if you make your original pieces just from white and black then you can multiply those colors by a color to make the white turn into that color and black stay black

GRADING

For full credit, be sure to follow the directions above. There are 100 pts and they are broken down as follows:

- 15 pts for rendering the base checker board and pieces in the initial positions, including graphics requirements
- 15 pts for selecting a piece and highlighting it, including ignoring clicks that don't make sense and allowing for selecting an alternate piece when a piece has already been selected
- 10 pts for showing whose turn it is and having it switch
- 15 pts for showing the possible moves of the current selected piece
- 10 pts for moving a piece
- 10 pts for jumping
- 10 pts for king pieces
- 15 pts for code quality (including minimal code redundancy)
- +5 Extra Credit for each of: multi-jump, forced-jump, or using click-and-drag instead of click

Guidance

It is recommended you follow these steps when working on this project. You may not strictly follow it, but I wouldn't work too far ahead before making sure a particular step is working. Look at the video for details of how my game went. The code you are given is extremely bare-bones except for the definitions of colors that I used in the video (which you can use, change, or remove as you wish).

- Start with getting just the board to draw with the two colors and all the squares in the correct places using transformations
- Add 'static' pieces on top of them to make sure you can draw them
    - To mimic the appearance in the video you can draw two circles on top of each other slightly shifted
    - Test out a drawing of a king piece
        - The king version in the demo is just more circles layered and more shifting
- Setup the logic for storing the game board as a list-of-lists and have `render` draw the board from that
- Add the click handler to allow clicking a valid piece and causing the render to "highlight" the piece
    - Make sure inappropriate clicks are ignored (e.g. on blank squares and opponent's pieces)
    - Clicks on a new valid piece change the highlight
- At this point most of the "graphics" are done and the remaining work is with the logic of checkers
- Show the possible moves of the current selected piece (but don't worry about jumping or kings yet)
- Add logic to allow clicking on a possible move and executing the move
- Implement jumping, this requires modifying the code to setup the possible moves and code for when clicking on a possible move the need to remove a piece
- Implement kings, this mainly requires modifying the code to setup the possible moves (and also in the rendering to be able to render a different piece)
- Make sure the game is switching turns and have it display a visual cue for whose turn it is
- ***Clean up all of your code and eliminate all code redundancy***