Project 01 - Checkers

Due: Friday Oct 17
Total: 100 pts

Checkers Game Logic

You will create a 3D version of the game checkers. To do this, you will first create the logic for the game in a separate class. Separating logic from rendering is part an important aspect of the Model-View-Controller (MVC) design pattern, allowing us to test the logic of the game completely separately from how the game is displayed. I have provided a basic 2D rendering for the game for you to be able to test with.

The game logic has the following rules / features:

- Red starts first
- Game can be reset at any point in time
- Pieces for the current player can be selected, figuring out the valid moves for that piece
- The valid moves can be queried for the currently selected piece
- The selected piece can be deselected
- The selected piece can be moved:
    - Regular pieces can move "forward" and jump/capture "forward" over an opponent's piece
    - Regular pieces that reach the other side turn into kind pieces
    - King pieces can move and jump/capture in any direction
    - When a piece is jumped over (or captured) it is removed from the game
- Eventual movement rules (do not need to implement immediately)
    - Multi-jumps – jump any number of times in a row, they don't have to be in the same direction
    - Forced jumps – if any piece for the current player can jump, the only moves allowed are jumps

When implementing the class, nothing that needs to be accessible outside of the class is allowed to be accessible. You must be efficient in the code (i.e. no iterating and evaluating every square as a valid move). To make your code work, you may not change the checkers2d.html/js files at all.

Remember: this class only maintains the state for the game and implements the logic, it knows absolutely nothing (and does not care) about how it will be displayed. This code would work equally well in a 2D game, 3D game, or even a terminal-based game. It may not accept any information about how things are being displayed.

Game Display

Once you have the game logic mostly work (I would recommend all but the multi-jumps and forced jumps) you can work on the 3D display for the game. The 3D display must have these features:

- Two views: the main one shows the game in 3D and a small view in one of the corners that displays an overhead view of the board. These must be two views of the same scene.
- In the main view, the board must be on a table composed of simple shapes. Uses orbit controls to allow the user to turn the board to different views. At the start of each turn, resets the view to what a player sitting at the table should see.

- The pieces must be fairly detailed, composed of either at least 3 distinct simple shapes or a custom geometry (not loaded model).
- King pieces must look similar but distinct, they must be distinguishable in both views.
- Selected pieces must be indicated in some obvious way (but not by the square changing color like in the 2D version). They must be distinguishable in both the main view and the overhead view.
- Valid moves for the currently selected piece must be indicated with a "ghost" piece (semi-transparent piece in fluorescent green). They must be distinguishable in both the main view and the overhead view.
- All geometry and materials must be reused as possible, i.e. if you have two cylinders, they must ultimately be the same geometry object. Similarly with materials.
- There must be decent lighting to view all parts of the game and world in nice manner. You will require at least 2 lights, possibly more to make it decent.
- There does not need to be any indication of whose turn it is or if the game is over.
- The reset button must work.

CHECK-INS

There are weekly check-ins on the following dates where you will be expected to show the following progress:

- **Oct 6:** `CheckersGame` complete (except force jumps and multi-jumps), check with working 2D game.
- **Oct 13:** 3D display renders the initial board game state including both views, table, and pieces. King pieces, selected pieces, and valid moves are not necessarily working.

Each check-in is worth up to negative 10 pts if not successfully completed.

RUBRIC

20 pts	`CheckersGame` is complete (except force jumps and multi-jumps)
5 pts	Force Jumps
5 pts	Multi-Jumps
10 pts	Main View works with view controls and changing players
5 pts	Overhead View
10 pts	Board and table render
10 pts	Pieces render as described
5 pts	King pieces render as described
5 pts	Selected pieces render as described
10 pts	Valid moves render as described
10 pts	User being able to select pieces and valid moves
5 pts	Lighting

Bad code style, including not reusing geometry and materials reused as possible, are deductions on top of completing features. Each check-in is worth up to negative 10 pts if not successfully completed.

EXTRA CREDIT

Only apply if the vast majority of the project is in great working order.

+5 pts	Use an SVG image and an extrude to assist in creating fancier pieces.
+5 pts	The table is clearly a wood table including texture.