**3D Chess**

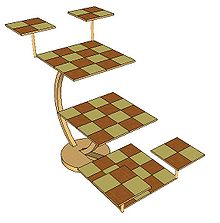
**Modules & Technologies**

Panda 3D, Blender, (Sockets)

**Introduction**

*Goal*

The goal of this project is to make a Star-Trek style 3D Chess that is functionable, interactive and playable. The player can choose to play against AI (one player) or against each other (two players). It should contain core features such that correctly displays the end of the game, only legal moves according to rules of the game are permitted, clicking on a piece would show possible moves and threats, and a deadly move (resulting in checkmate) is not permitted. It should also include captured pieces listed on the side of the canvas, redo the last move, restart the game, switching between 3D and 2D views, show the last move by the opponent, and a detailed tutorial regarding the game.

*Description*

The game starts with a splash screen with start-game, tutorial and settings. In start-game, there with be a sub-splash screen of the freedom to choose in among one player against computer, multiplayers from one computer and from multiple computers. After entering the game mode, the canvas is shown on the left with a starter chessboard set. The right is a toolbar of multiple tools used in the game.

*Tutorial*

Tutorials include basic operations of my game and rules of 3D chess.

Arrow keys spin the camera, clicking on squares select corresponding pieces, double clicking select attack boards, “u” undo last move, “r” redo last move, “q” gives initial view, “esc” quits the game.

A 3D chess tutorial is very much similar to regular chesses such as <https://www.thespruce.com/rules-of-chess-611533>. It contains same sets of pieces, and legal moves on a 2D basis. In addition, each piece can enter different levels of boards based on a set of rules from <http://www.thedance.net/~roth/TECHBLOG/chess.html>. A slightly more different feature is of 4 movable 2x2 boards(attack boards). While each contains strictly less than 2 pieces, a move can be made by moving the board up or down on the same column of different levels.

*Game Mode*

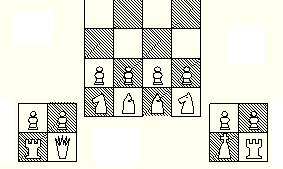
The chessboard is designed to be 3-D. A complete set of chess board contains 3 levels of 4x4 boards and 4 movable 2x2 boards starting on each side. The game starts with 4 pieces placed on 2x2 boards and 8 pieces on 4x4 boards as drawn below.

The view is designed to be rotatable, providing the player a better vision of the entire game. In the process, each move is recorded and can be explicated shown, redo/undo using the toolbar on the right.

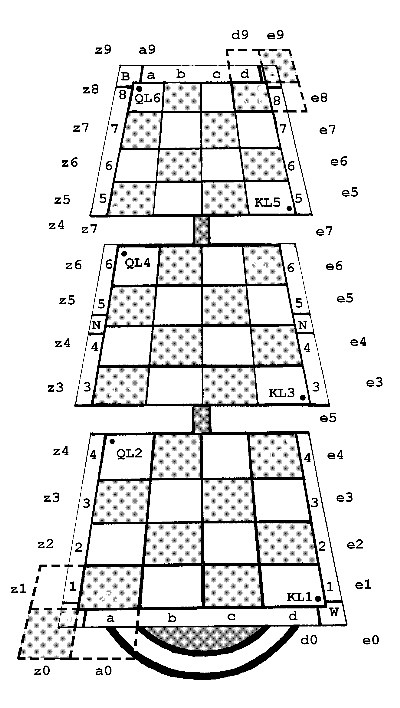
*Settings*

The game includes some basic settings of changing background, changing color of pieces, the option to display possible moves and threats.

**Task**

1. Display the boards on the canvas with correct labeling. Use mouse clicking to test availability.
2. Put pieces onto the board and with a mouse task of clicking on the square with pieces, select the piece as the current piece.
3. Find available trails of the piece and highlight them. If one of the highlighted square is clicked, move the piece onto the square. Display “alert” texts for illegal moves. If the one clicked contains a piece of the opponents’ piece, remove the piece by putting it into removed list.
4. If an attack board is double clicked, find possible moves and highlight them. The rest is similar to the moves of pieces.
5. Display all the removed pieces as a line on the side of the board using a removed list.
6. By detecting key events, spin the camera for a full overview of the board, redo/undo/rest the game, give suggested next moves.
7. In AI mode, the computer makes several potential random moves and check which one is the best and make the move.
8. Draw splash screen of start game, tutorial and settings.
9. Add multiplayer feature.

**Core Challenge**

1. *Displaying the game in 3D mode and detecting mouse click events*

I intent to use models of chess pieces, and generate the boards and connect parts by making 3D models in Blender. After importing these models, I will position them on the correct position as the game goes over time. As mouse clicking only returns 2D position, I will detect whether the position has a board object and return the actual place on the board. This return value should be unique within the entire board set; therefore, I will use a notation combining letters and rows and columns as the label of each square, and relate the chess to the squares.

1. *Moving attack boards based on the rules of the game*

In order to move attack boards, I will detect mouse events using “double-click” and highlight the board selected. Note that since I consider each square separately, I will have to move the entire board together as an object. In order to do this, I will create a class of attack board, and move the position of the board together.

1. *Spinning the complete set of model or camera so that different perspectives can help players analyze the game and make the move*

Since a better perspective is provided with rotation, I will use arrow keys to finish this task. I will try to move the camera around the set, so that I don’t have to change the position of the set.

1. *A smart game AI*

Let the computer analyze several potential moves using a tree of random moves, then choose the best one to make the move. The move can be made by checking the threats first to avoid the threats first, and then move to attack.