# Checkers game ”design”

Rules:

* Pieces only move diagonally
* One move per turn
* Normal white and black pieces move in opposite directions
* Normal pieces can only move 1 square at a time
* A normal piece can be made a king by getting to the end of the board
* King pieces don’t have movement limitation of 1 square at a time
* If an opposite color piece is one square away, and there is an empty space behind them, that piece can be eaten by jumping over it. You can continue the jump if there are other eatable pieces.
* Game ends when there are no opposing pieces, or they cannot be moved

## Basic turn structure blueprint

endTurn

removePiece

UI

-UI.cs

Engine

-engine.cs

movePiece

-board.cs

moveAllowed

-engine.cs

postMove

-engine.cs

isDiag

spaceOpen

rightDirection

spaceCloseEnough

isKing

true/false

isKingable

additional captures

## Classes

Board.cs responsible for state of game and gameboard.

GameEngine.cs responsible for rules like how pieces move.

UI.cs is responsible for how pieces and gameboard look. Maybe additional features like if piece is capturable, show piece in red, or allowing user to change game settings.