Class player

- Name
- Color
- Something to handle whether a computer or a person? Enum player or AI

Class checker

- Boolean king
- Enum color black, red
- checker(isKing, color)
- isKing()
- setKing()
- getColor()

Class square

- Instance variable checker
    - Will be null if there is no checker here
- isEmpty();
- getChecker();
- setChecker();

Class board

- Instance variable 2d array of squares
- Instance variable enum? Player turn
- Instance variable Boolean mustJump
- enum Winner   (red, black, draw)

```
Initialize()
Move(int x1, int y1, int x2, int y2)
    Starting position and end position
    If isMoveLegal (int x1, int y1, int x2, int y2) && is not a jump
        Move the checker in 2d Array
        Switch turn
    If isMoveLegal(int x1, int y1, int x2, int y2) && is jump
        Move the checker in 2d array
        Delete the checker that was jumped over
        Determine if there is a further jump
            mustJump = true
        otherwise switchTurn()

isMoveLegal(int x1, int y1, int x2, int y2)
    starting position end position
    use position to get the checker
    make sure checker color is same as turn color
    check that x2 and y2 are in bounds
    make sure target position unoccupied
    if( is move)
        if(mustJump == false && canJump == false)
            return true
        else false
    else if (is jump)
        if it jumps over the other color
            return true
        else false
    else
        return false
    not off the board, space is unoccupied, it is diagonal

canMove(color)
    iterate through the board, for every checker of the color
        if checker
```

```
            check if any of the two moves available
        if king
            check if any of the 4 possible moves available (4 normal)
    return true if one can move, false otherwise
canJump()
    iterate through the board, for every checker of the color
        if checker
            check if any of the two jumps available
        if king
            check if any of the 4 possible jumps available
    return true if one can move, false otherwise
countCheckers(color)
    iterate through the board, for every checker of the color
    return #

^^^^^ this will be called by UI after every move

switchTurn()
    changes color
    sets mustJump to false

isJump(int x1, int y1, int x2, int y2)
    if difference between x1 and x2 is two && difference between y1 and y2 is two
    return true
    otherwise return false
isMove(int x1, int y1, int x2, int y2)
    if difference between x1 and x2 is 1 && difference between y1 and y2 is 1
    return true
    otherwise return false
```

```
enum checkWinner()
    if countCheckers(red) == 0
        return black
    if countCheckers(black ) == 0
        return red
    if canJump(red) == false && canMove(red) == false
        if black canMove || canJump
            return black
        else
            return draw
    if canJump(black) == false && canMove(black) == false
        if red canMove || canJump
            return RED
        else
            return draw
```

draw condition?