

WHITE

| | | | | | | | |
|------|------|------|-----|-----|-----|-----|-----|
| | 1,12 | | 3,9 | | 5,6 | | 7,3 |
| 0,13 | | 2,10 | | 4,7 | | 6,4 | |
| | 1,11 | | 3,8 | | 5,5 | | 7,2 |
| 0,12 | | 2,9 | | 4,6 | | 6,3 | |
| | 1,10 | | 3,7 | | 5,4 | | 7,1 |
| 0,11 | | 2,8 | | 4,5 | | 6,2 | |
| | 1,9 | | 3,6 | | 5,3 | | 7,0 |
| 0,10 | | 2,7 | | 4,4 | | 6,1 | |

BLACK

FROM THEIR SIDE OF THE BOARD

X and Y = current position (X,Y)

Black Player: fwd right = $x+1, y-1$

fwd left = $x-1, y+2$

ONLY LEGAL FOR "Kings"

back right = $x+1, y-2$

back left = $x-1, y+1$

White Player: fwd right = $x-1, y+1$

fwd left = $x+1, y-2$

ONLY LEGAL FOR "KINGS"

back Right = $x-1, y+2$

back Left = $x+1, y-1$

- determine who wins by holding location of their 8 pieces in an array of checkers /

pieces. each "piece" will hold,

```
piece{
    int position[2]: {1,12},
    bool king: false, //will allow you to move backwards
    bool stillAlive: True;
    enum color: white //or can also be black
}
```

- to win the game all 8 pieces of your opponent have to hold "False" in all of the "stillAlive".

- determine if move is legal by testing in the above equations and king status

each square will hold the value represented on the above UI. The board will be an array of "Spot" ...it will start making boxers from bottom left and move right when building board.

```
Spot{
    int id[2] = {0,10},
    bool occupied: true,
    enum color: black; //black or red, if red it will set id to{0,0} which is not
                        allowed on the board
}
```