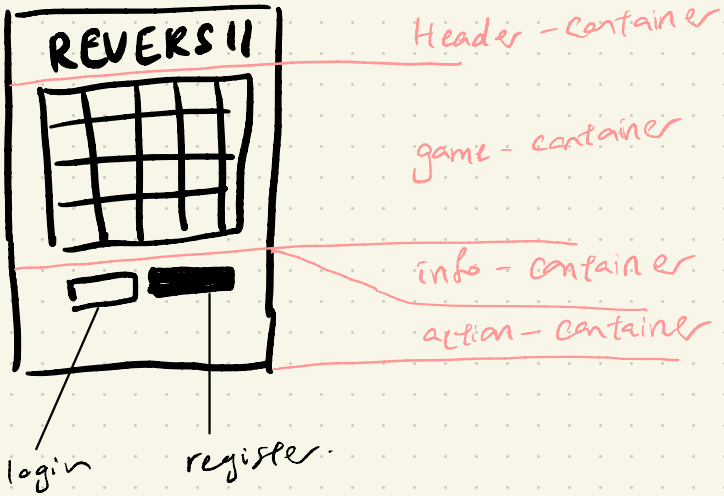
src:
index.js

axios.get('/items')

router
app.get('/items', itemController)

.index

./controller/item.mjs

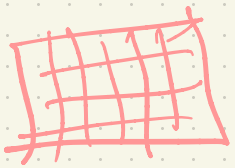


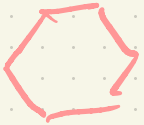
River sea ~
otters

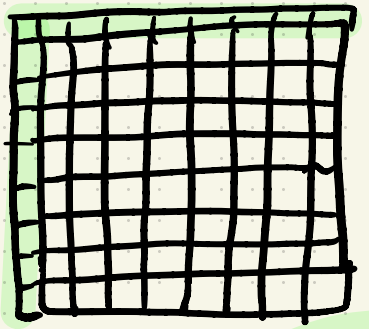
clam
rating : num ō
fish

Territories:
→ Bishan
→ sungei bulloh

otter - linear





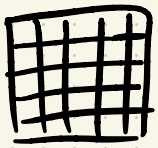


click on cell

board is refreshed every turn? → better to evaluate & redraw vs checking / emptying divs & overwriting?

board is incrementally added to every turn

initboard



render board from board data

Turns	B	W
0	✓	✓
1	✗	✗
2		
...		

prev game data not updated?

Always retain first & last [curr game]

- prev game not updated
  - W white / subsequent moves

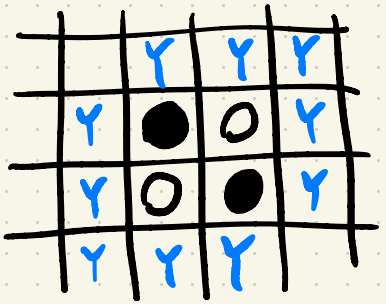
board data →

data →

move
→ row
→ col
→ isBlack?

i →

j ↓



1. run through all cells

Checks

2. adjacent to white?

direction ō adjacency

[T, L, B, R]

Find opponent → find vacant adjacent cells
search all cells, search for adjacency → check all adjacencies

→ use object for ez ref

[T, L, B, R, D_1, D_2, D_3, D_4]

const adjacency = [false] * 8

if adjacent[k] :

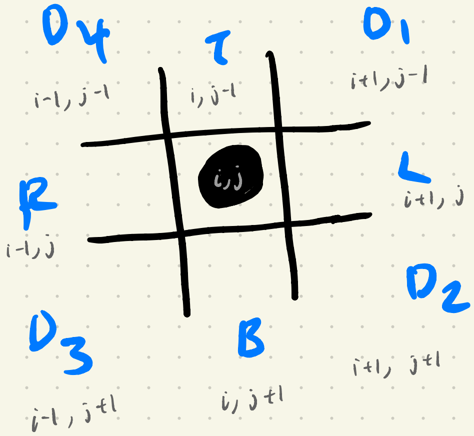
check for ●

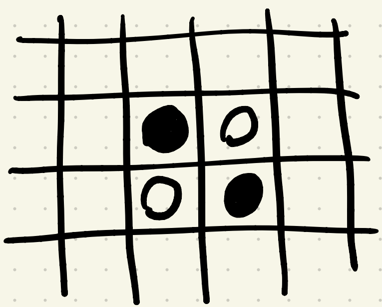
T: [i, j-2] → [i, 0]

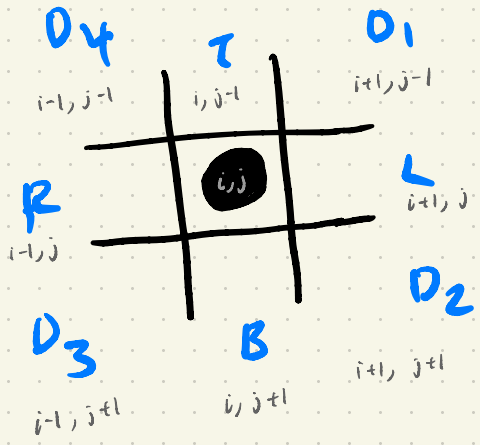
L: [i+2, j] → [7, j]

B: [i, j+2] → [i, 7]

R: [i, j] → [0, j]







$[\ldots, D_1, D_2, D_3, D_4]$

$D_1 : [i+2, j-2] \rightarrow [i+3, j-3]$ $i \rightarrow 0$ $j \rightarrow 0$

$D_2 : [i+2, j+2] \rightarrow [i+3, j+3]$ $i \rightarrow 7$ or $j \rightarrow 7$

DTL
DBL
DBR
DBL

$D_3 : [i-2, j+2] \rightarrow [i-3, j+3]$ $i \rightarrow 0$ or $j \rightarrow 7$

$D_4 : [i-2, j-2] \rightarrow [i-3, j-3]$ $i \rightarrow 0$ or $j \rightarrow 0$

Return matching coord. ←

else return;

If found, break all

validMove = { coord : [i, j],
matchingCoord : [i', j']

T : [i', j']
L : [i', j']
R :
B :
DTL :
DBL :
DBR :
DBL :

}

← if not in, num

save in gameState

validMoves = [.....] → return to client to display

client picks move among display

3. Flipping

check if click is in valid moves

↙ Yes

↘ No, return

↓

find move object in existing

FlipTokens (moveObj, boardData)

→ for cells from move to match.

!

↓ count tokens black & white

↓ return

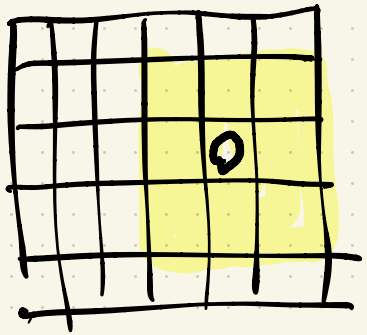
bool[8][8]

→ double counting

check adjacencies

check each cell adjacency

highlight cell around white



bool[i][j]==null)

⋮

diff i & j for inputing adjacency

cells : [ij : [T :
L :
B :
R :
DTL :
... 3

77 : ...]

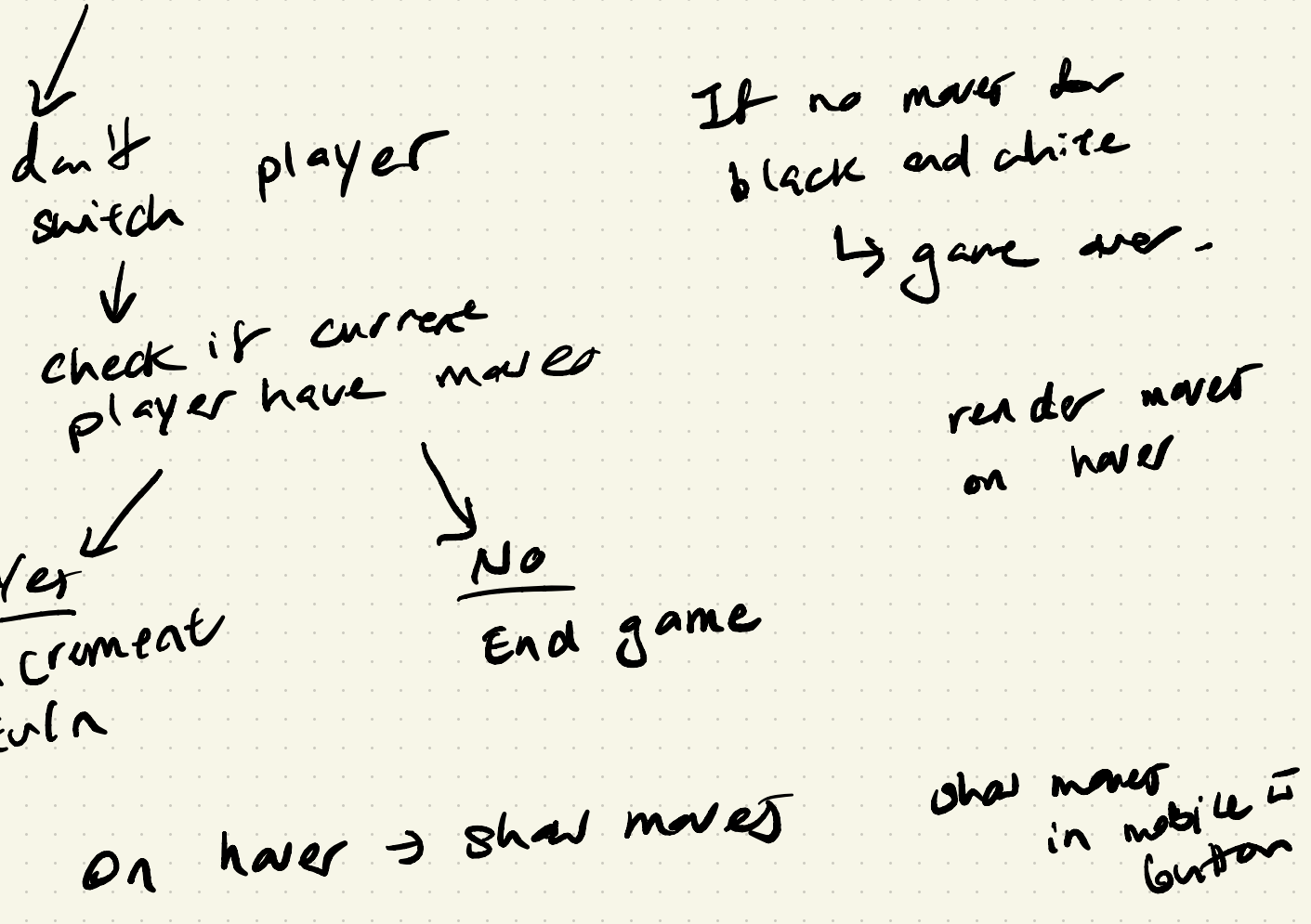
XHR PUT

error in
click on
cell

- restrict moves to valid
- flip everything in between
- check available moves o opponent

moves.length === 0

if moves.length > 0
white turn ⇒ turn increment
isBlackTurn = !isBlackTurn


switch player
check if current player have moved
Yes
increment turn
No
End game
If no moves for black and white
↳ game over.
render moves on hover
On hover → show moves
show moves in mobile ☐ button