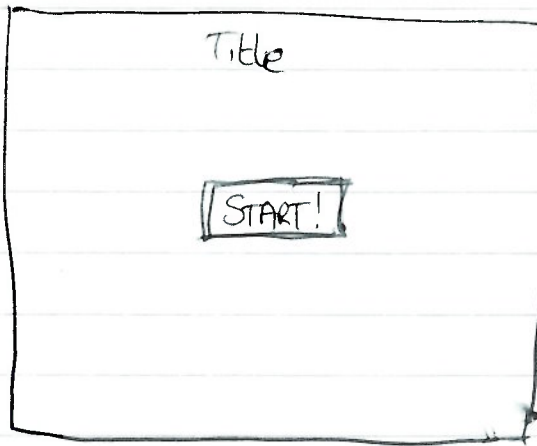


Main Screen

(Keep simple!)

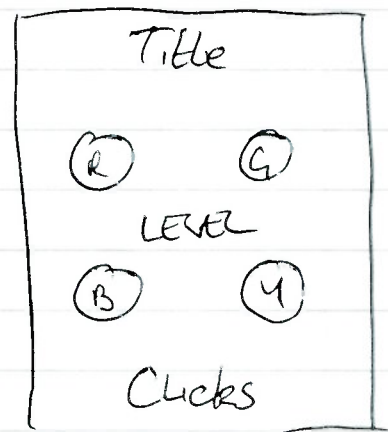
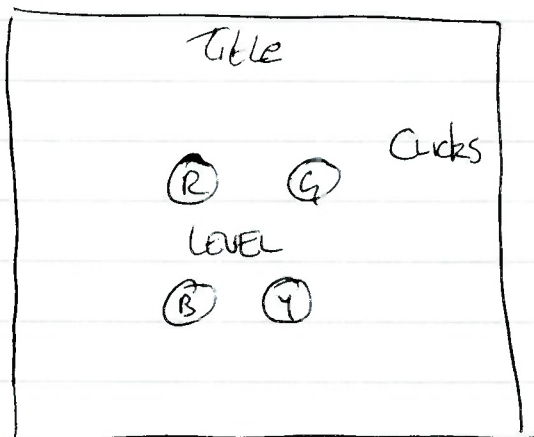


Simple start button! →
fade out to reveal
board

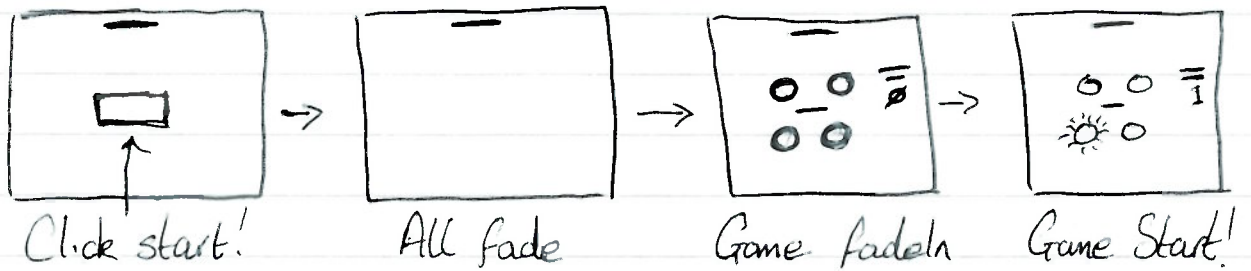
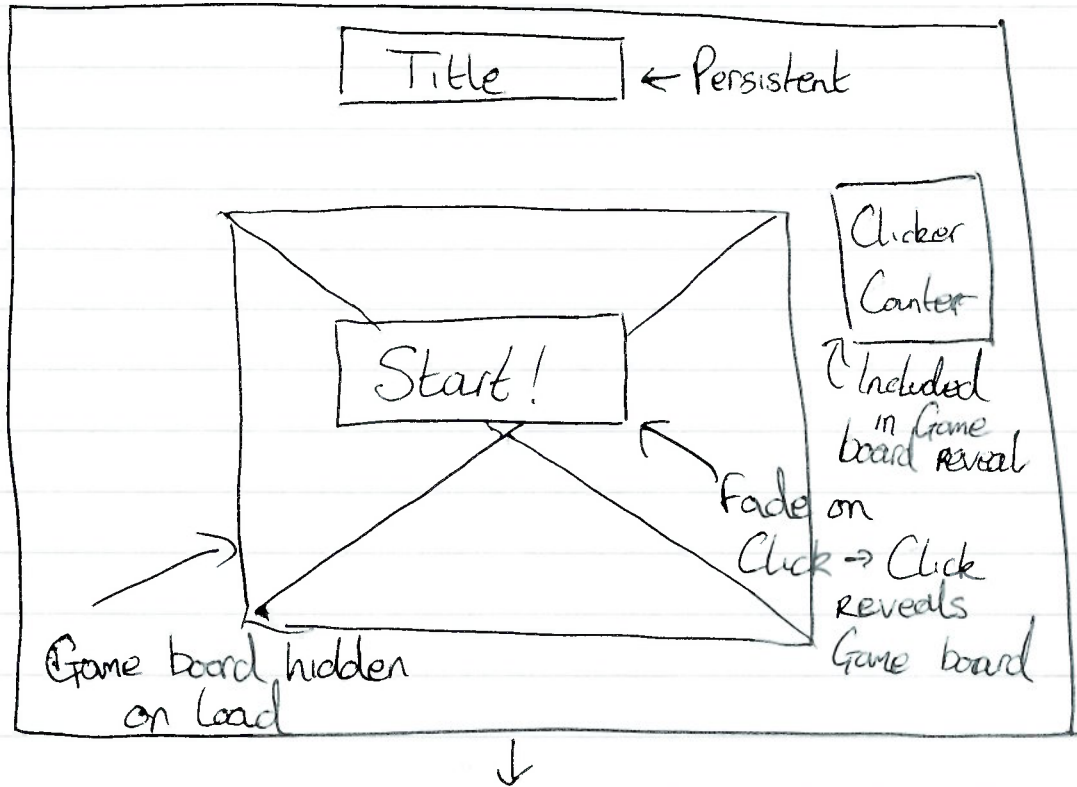
On desktop → Minimize board width for easy transition
to tablet

Desktop

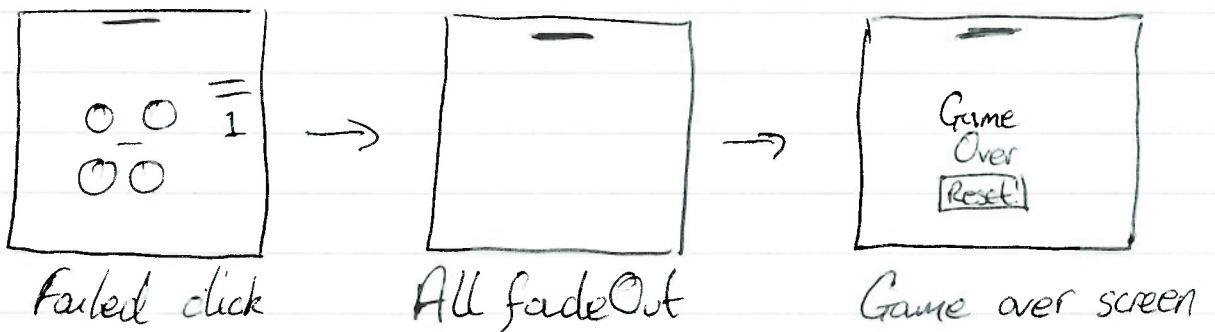
→ Tablet

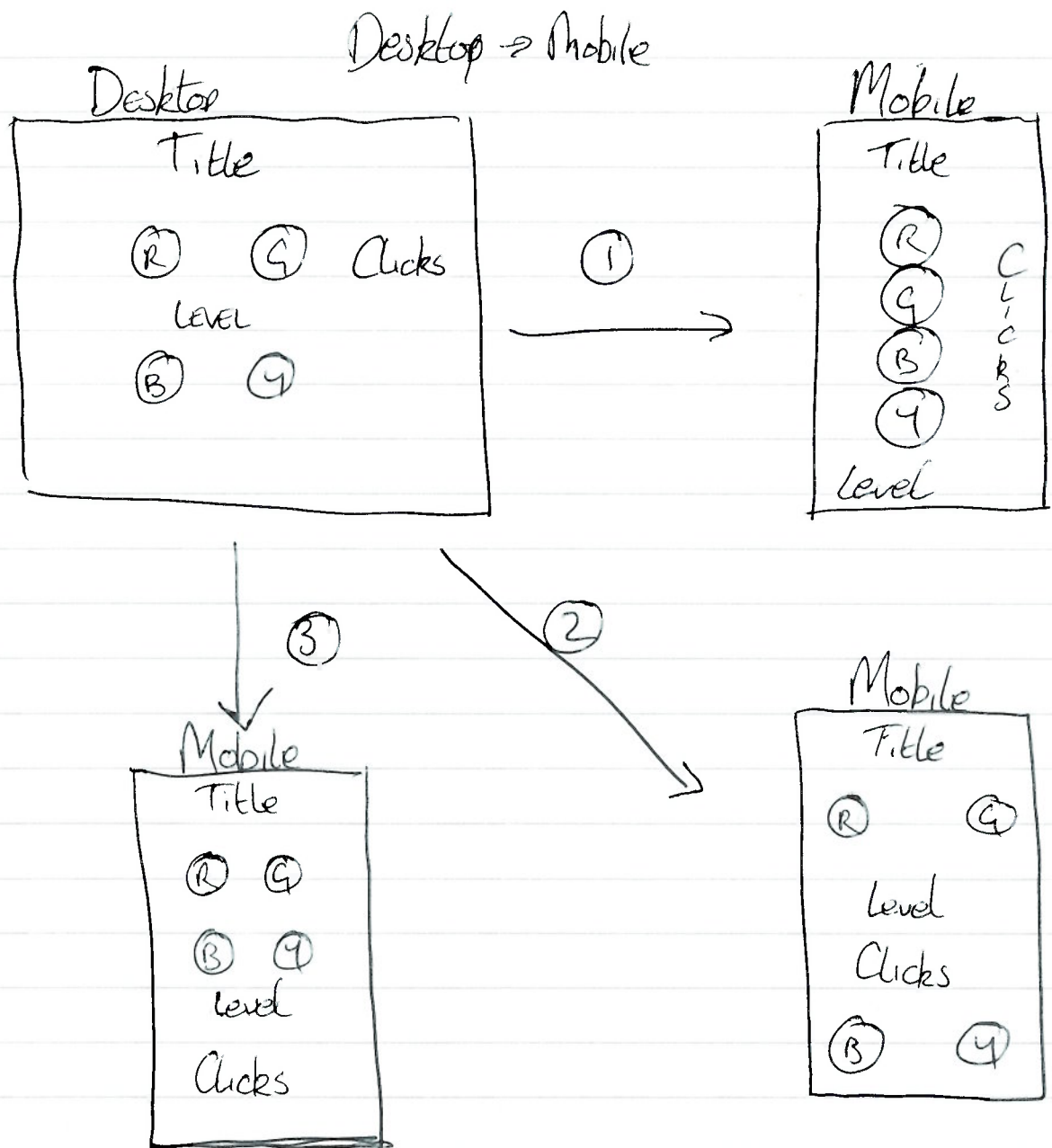


Desktop fades



On failure





Above to be tested for usability

Test functionality on mobile for what works best!

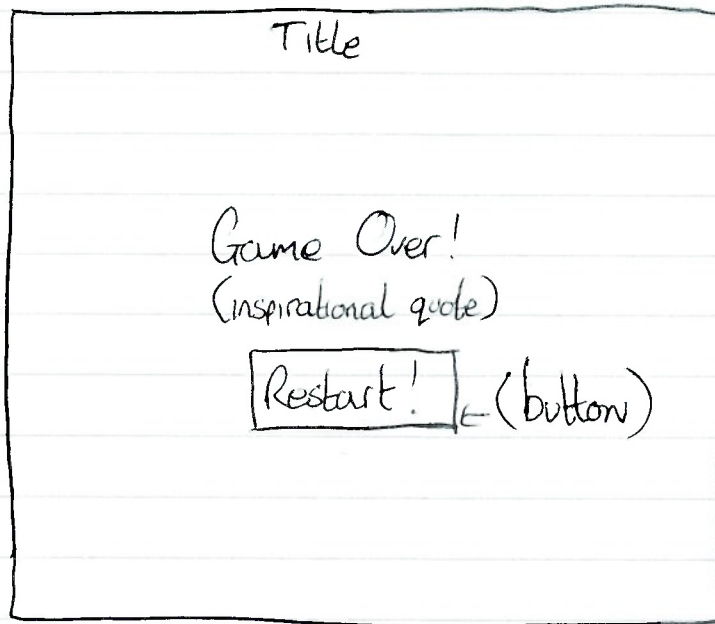
Important!

Check user moves against computer moves?

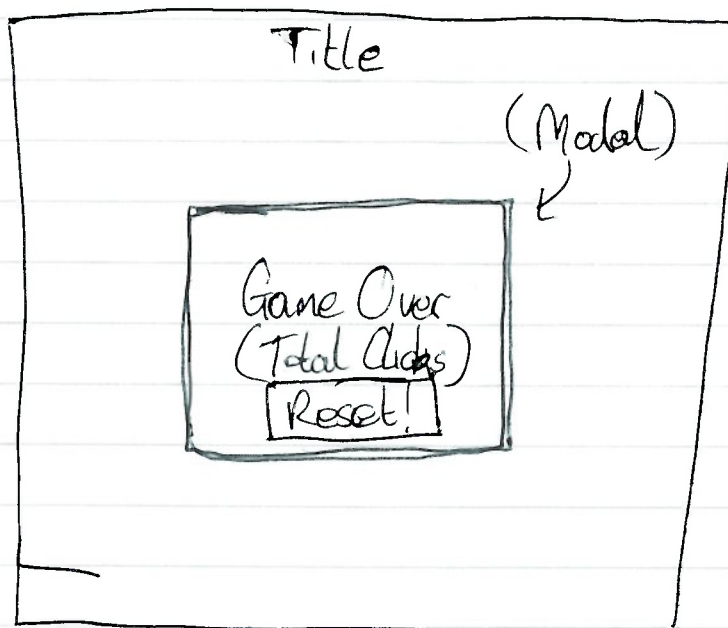
ArrayPC [i] } ?
ArrayPlayer [i] }

Check for input before iterations?

Possible Game Over screens



\ Or /

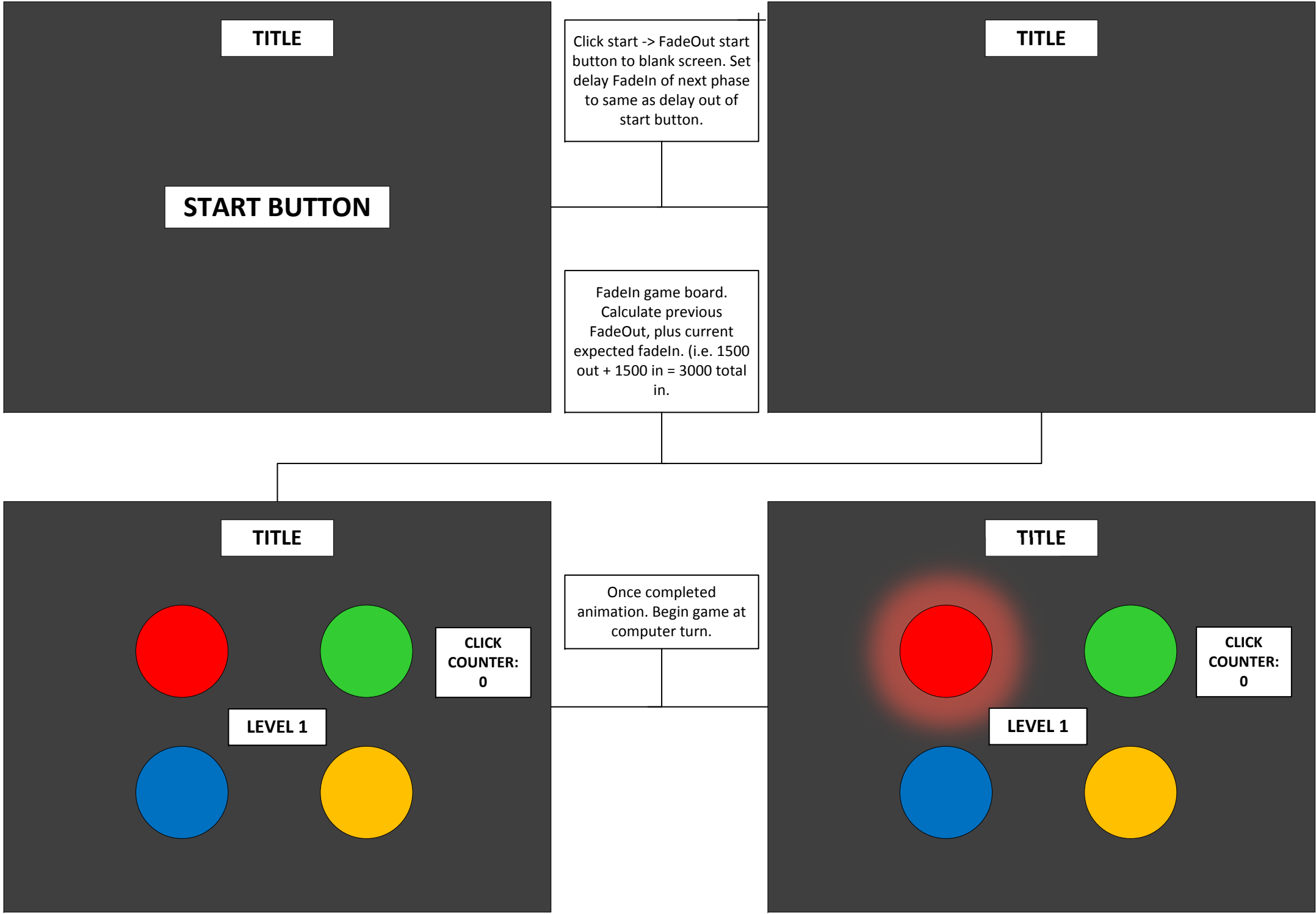


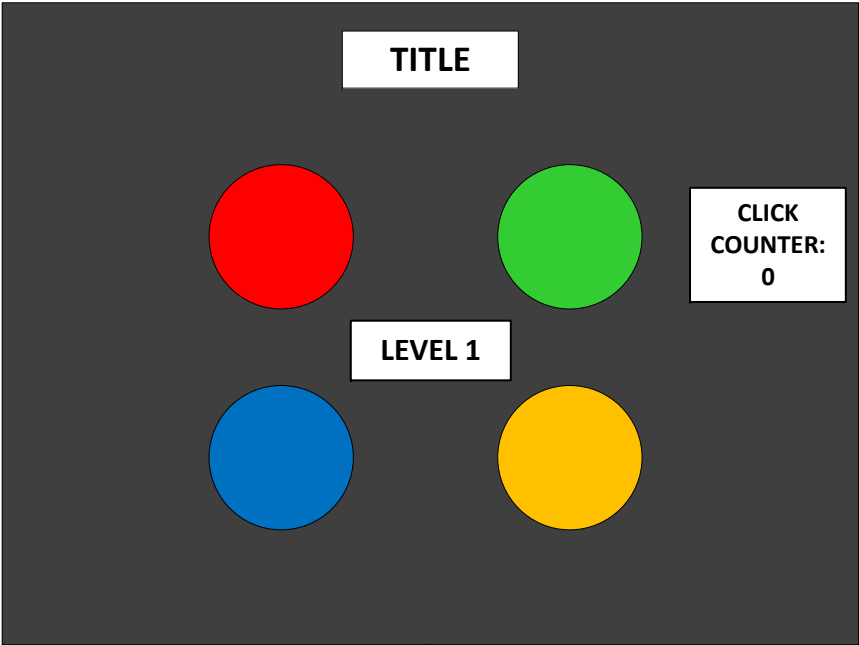
Find way to display final value of total clicks to user

! Important !

Redraw using Visio or other app!

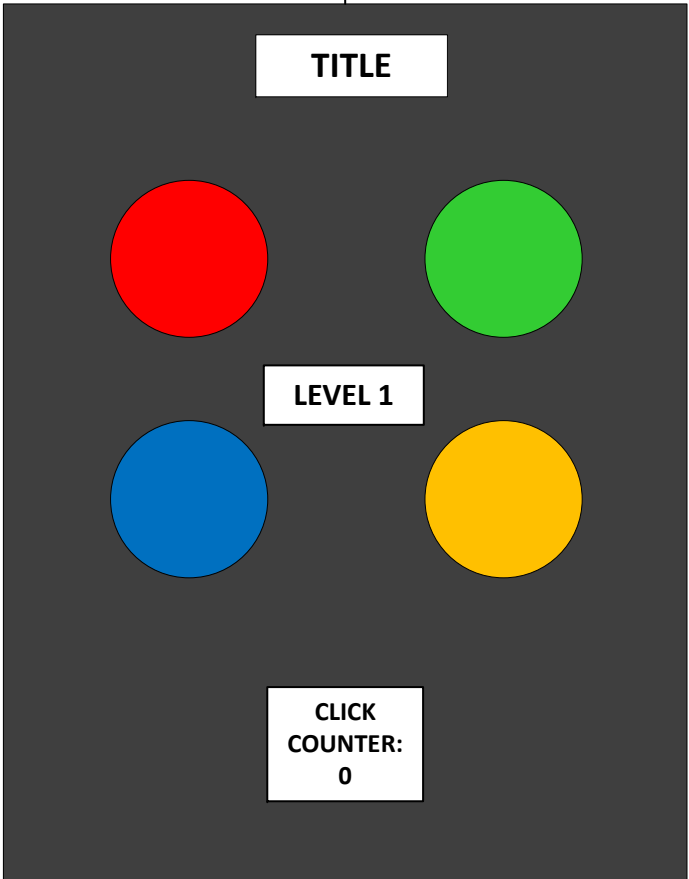


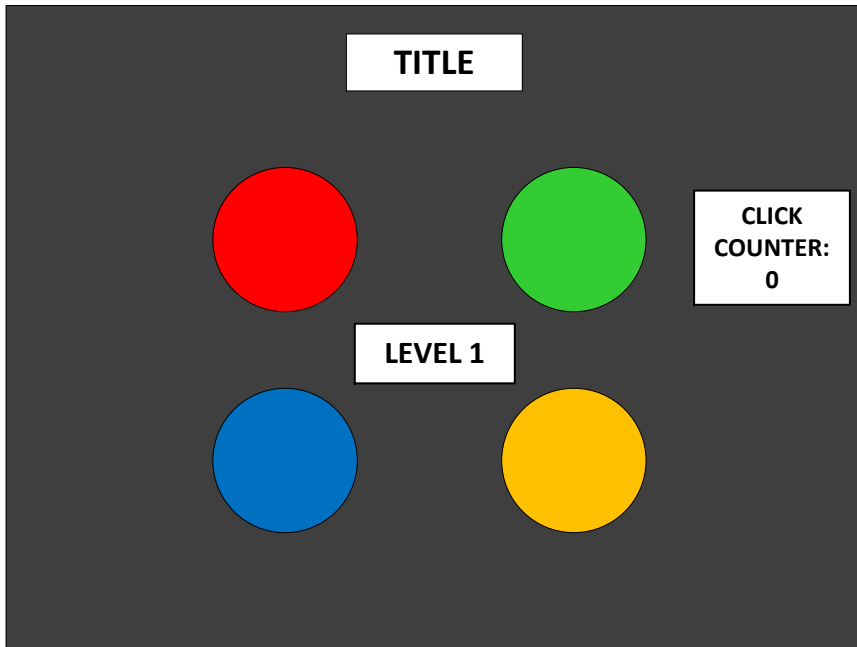




Desktop to Tablet:
All items align the same. In desktop, use an 8-col offset with 2. In tablet, use a col-12 to utilize all space.

Move Click Counter to bottom of page.

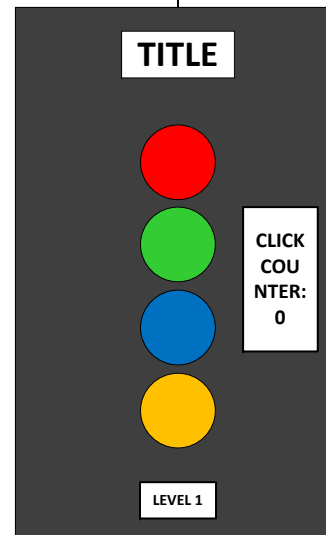
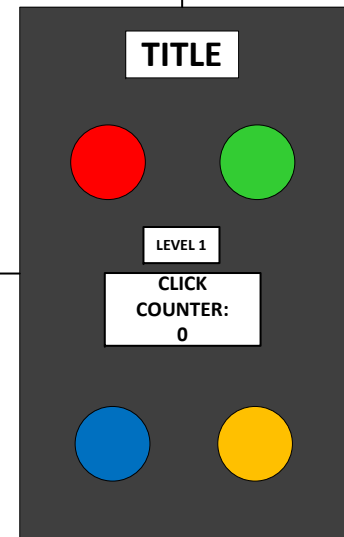




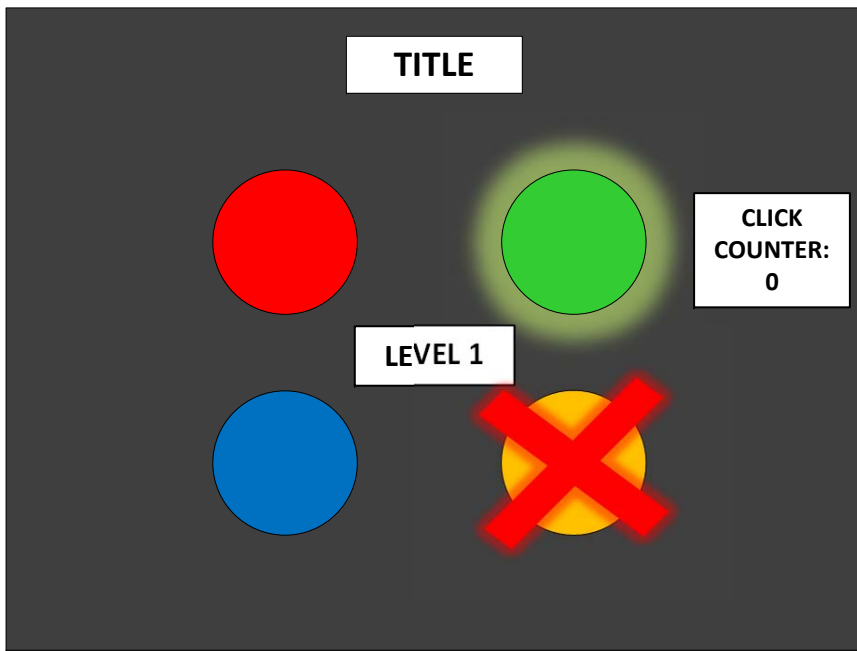
Move level counter, and clicker counter to bottom of page. May reduce size of buttons, and make usability low for clicking. Testing required.



Move clicker counter and level counter to centre page. Makes page look neater, with a circular theme to all the elements. Again, may reduce button size, and therefore usability. Testing required.



Centre align all buttons. Meaning greater size in buttons, making easier to click. Doesn't look the prettiest when lined up as opposed to square layout. Looks versus usability. Clicker counter also reduced space, meaning redesign if unable to fit.



When incorrect button is clicked, fadeOut game board.



1.



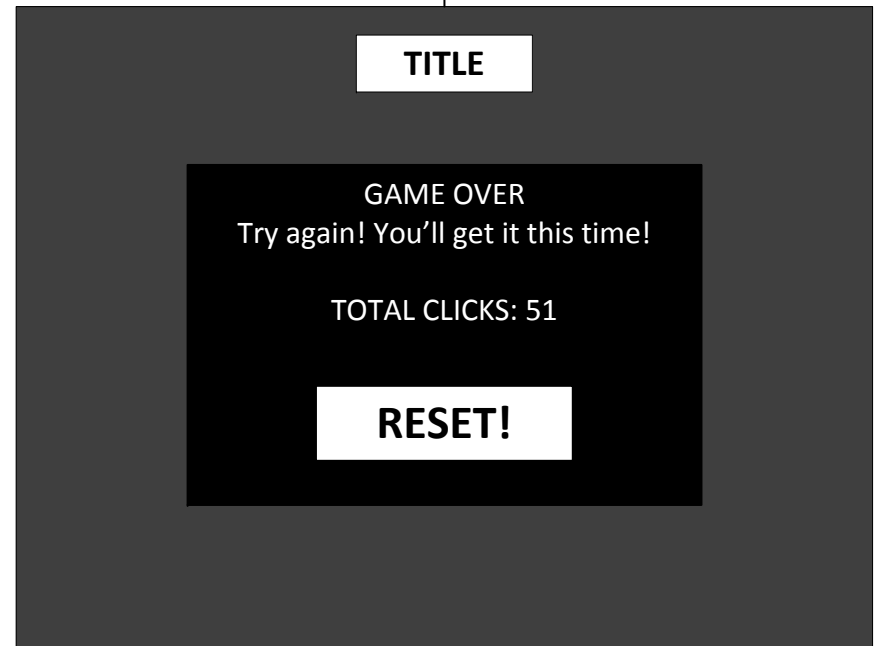
After fadeOut, one of two possibilities going forward.

1. fadeIn game over screen with final total clicks value. Include reset button below.

OR

2. Bring modal into view with current total click count and reset button.

2.



variable MovesMade=[] — "LEVEL" + movesMade.length

Constant possibleMoves=[] — red, yellow, green, blue

generateMove = $(\text{math.floor}(\text{math.random} * 4))$
Push above result to movesMade array



playerMoves=[] — how to make this match & check against movesMade array

every click generates a number which pushes to this array

~~every click generates a number which pushes to this array~~

$i = 0, i < \text{mM} < \text{Length}, i++$
\$

movesMade [].length === playerMoves.length
movesMade [] === playerMoves []

~~Jack~~