

Touchless Solitaire:

I will be coding a slightly modified solitaire game where the user can move around cards with eyes only. In the solitaire game, the game will use backtracking to be able to check if the initial setup is solvable. The user then is able to choose cards to move to another pile (as per the rules of solitaire), flip a card from the deck, build up the A → K piles and wins when all four of the A → K piles are complete.

There will be 6 buttons placed around the screen. Using eye-tracking, I will identify which button the user is looking at. When the user looks at one button, and then I blink for an extended time, that button is selected and that button is one of 6 actions (move last card in pile 1, 2, 3, 4, move card that is upwards in the sidebar from the deck, flip from deck)

Similar Projects

Here's a solitaire made by google:

<https://g.co/kgs/9G2NFfT>

Features to learn from:

- The card flipping animation and card sliding animation format
- When you press on a card, if it can move, it'll automatically move to the right position
 - (no need to drag yourself)
- Flip deck gives one card at a time
- There is a move count

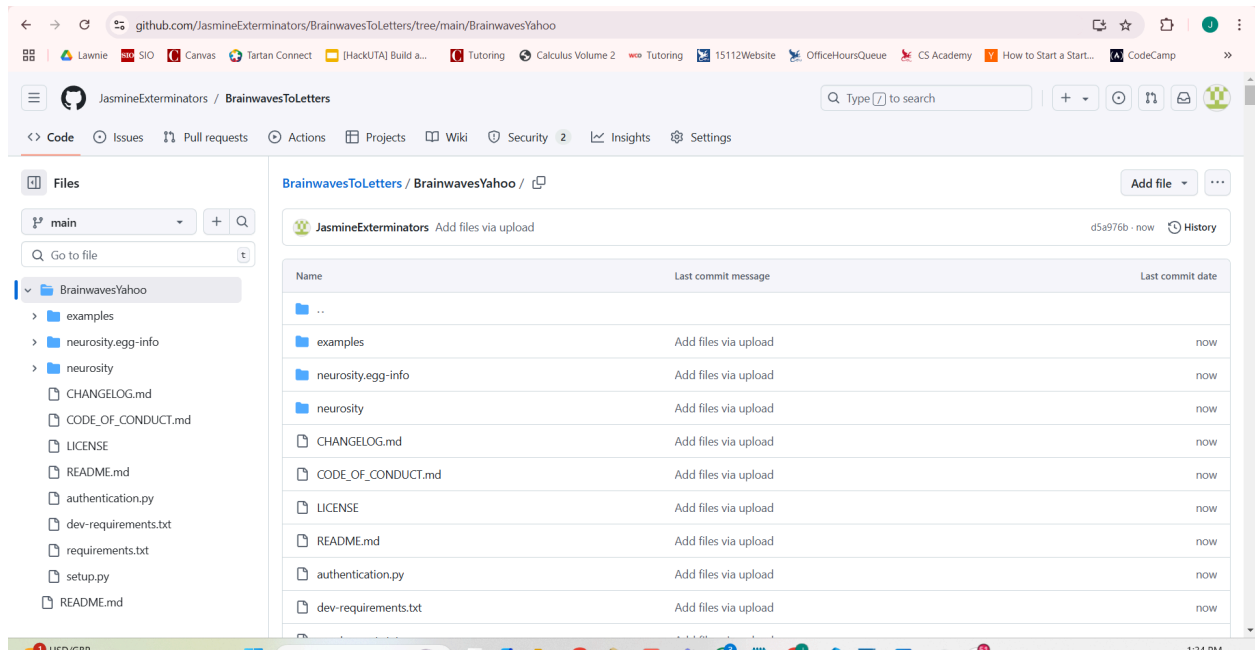
<https://www.solitairebliss.com/>

Features to learn from:

- You can undo
- You can ask for a hint and that highlights all movable cards

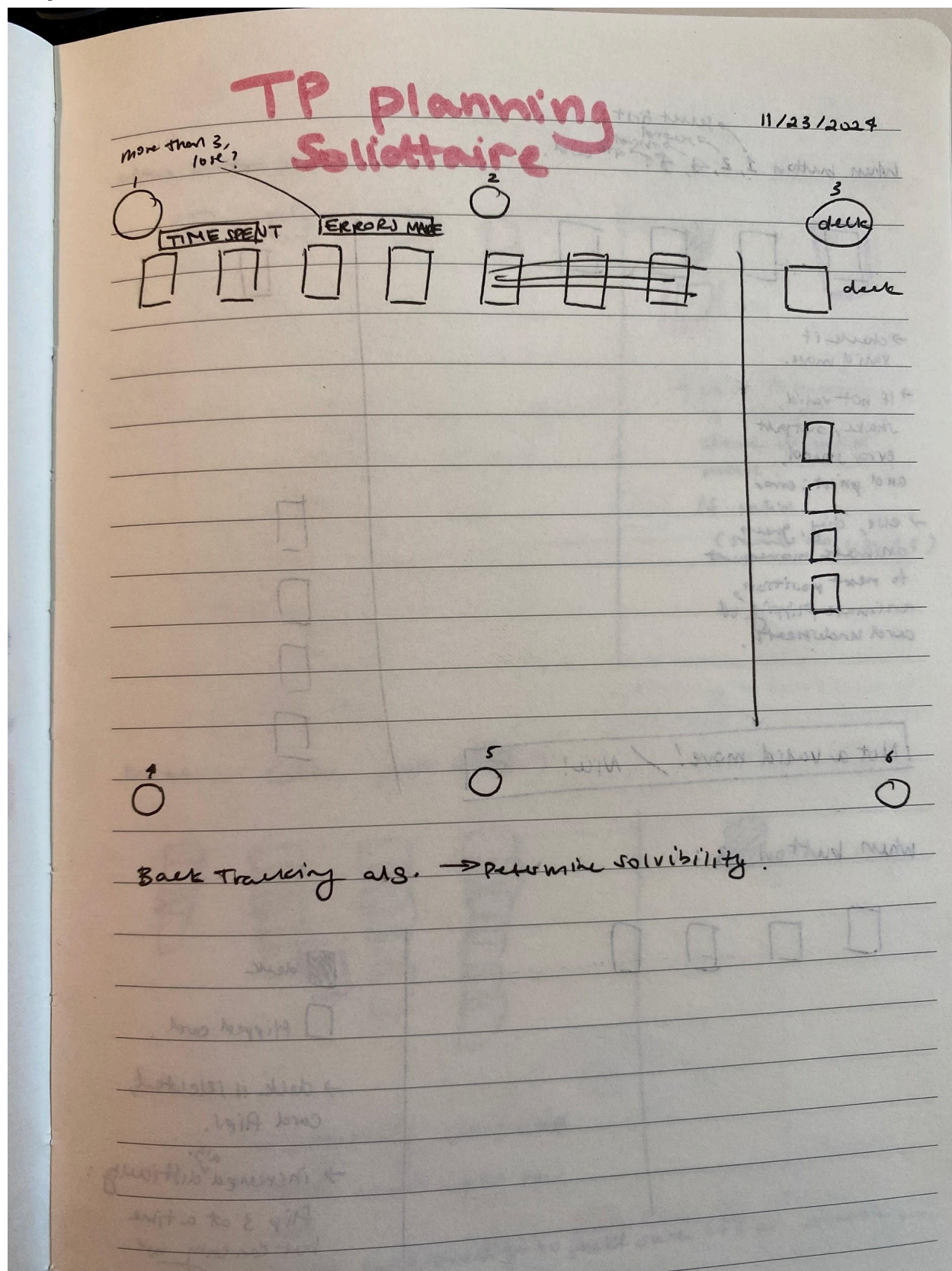
Version Control / Backup Plan

I'm using a Git Repository (backed up daily):

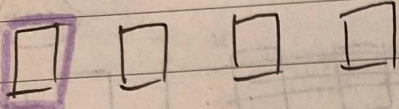


Tech List

Camera on Computer + OpenCV



When button 1, 2, 4, 5: ^{select first}
^{second}
^{third}
^{4th card.}

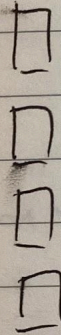


→ check if
valid move.

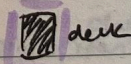
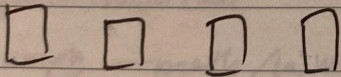
→ If not valid,
shake, output
error sound,
and print: error

→ else, dir, green,
animate movement
to next position,
animate flipping of
card underneath.

Not a valid move! / Nice!



When button 3:



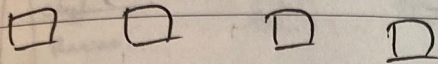
Flipped card.

→ deck is selected,
card flips.

→ increased ^{alg.} difficulty:

flip 3 at a time
but can only use
the last flipped.
(so some cards are passed
by)

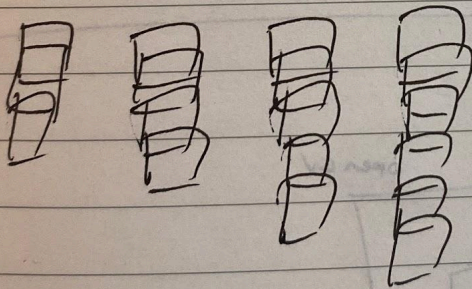
when button 6:



→ Card from deck
is selected,
check if valid
move.

if valid...
(same as 1, 2, 4, 5)

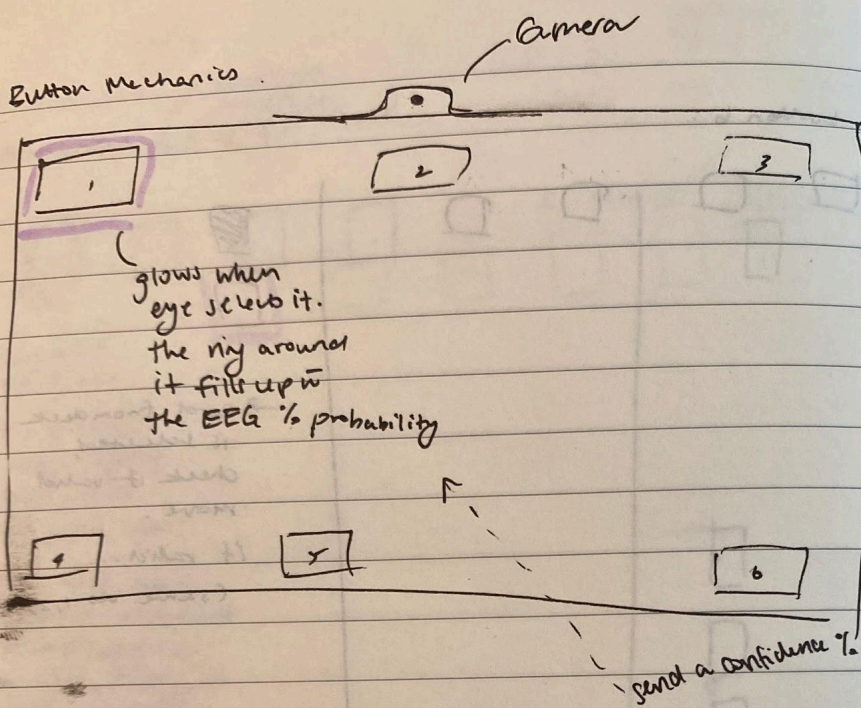
Reset: (when press ^{key 'r'} ~~button~~)



→ everything clear and
then shows up as new
set.

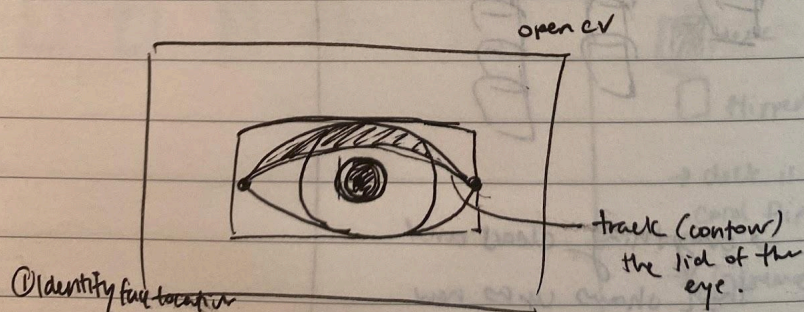
→ use backtracking to make sure it's a solvable one.

Button Mechanics



To make it less prone to error,
lag 2 sec (must be eye stationary
on button for 2 sec before point
complete)

Eye tracking:



① Identify face location

② Identify eye position

③ Isolate the pupil & track ^{angle} movement.

← ~~contour~~ contour detection

OPENING

generate
then that
more A.
at top

opening screen:

