

Code Clinic Tips

This page include helpful tips based on our experience in helping students in the "Code Clinic" (interactivepython@online.rice.edu). Be sure and take a look at these tips if you get stuck.

Tip #1

Many of you are tempted to make conditionals with 16 cases to handle every possible card position on the canvas. Please don't. Learn how to use loops. For example, if `cards` contains the list of card values, the following code is a good start towards drawing cards

```
for card_index in range(len(cards)):
    card_pos = 50 * card_index
    canvas.draw_text(str(cards[card_index]), card_pos, ....)
```

Note that we could also iterate directly over the cards via `for card in cards:`. However, iterating over the indices of the cards will make adding code to expose cards based on mouse clicks easier.

Tip #2

In your mouse click handler, you need to determine which card you have clicked on. Don't use a long if/elif/else statement here. Compute the index of the card via `pos[0] // 50`. You can then use this index when manipulating the list exposed that control whether a card is drawn or not.

Tip #3

As you get close to the end of this mini-project, you will need to determine if the two cards match when you transition from state two to state one. The key here is to keep track of the indices of two cards that were

clicked on during the last turn (in two separate variables) so you can test whether their values are equal during the next turn.

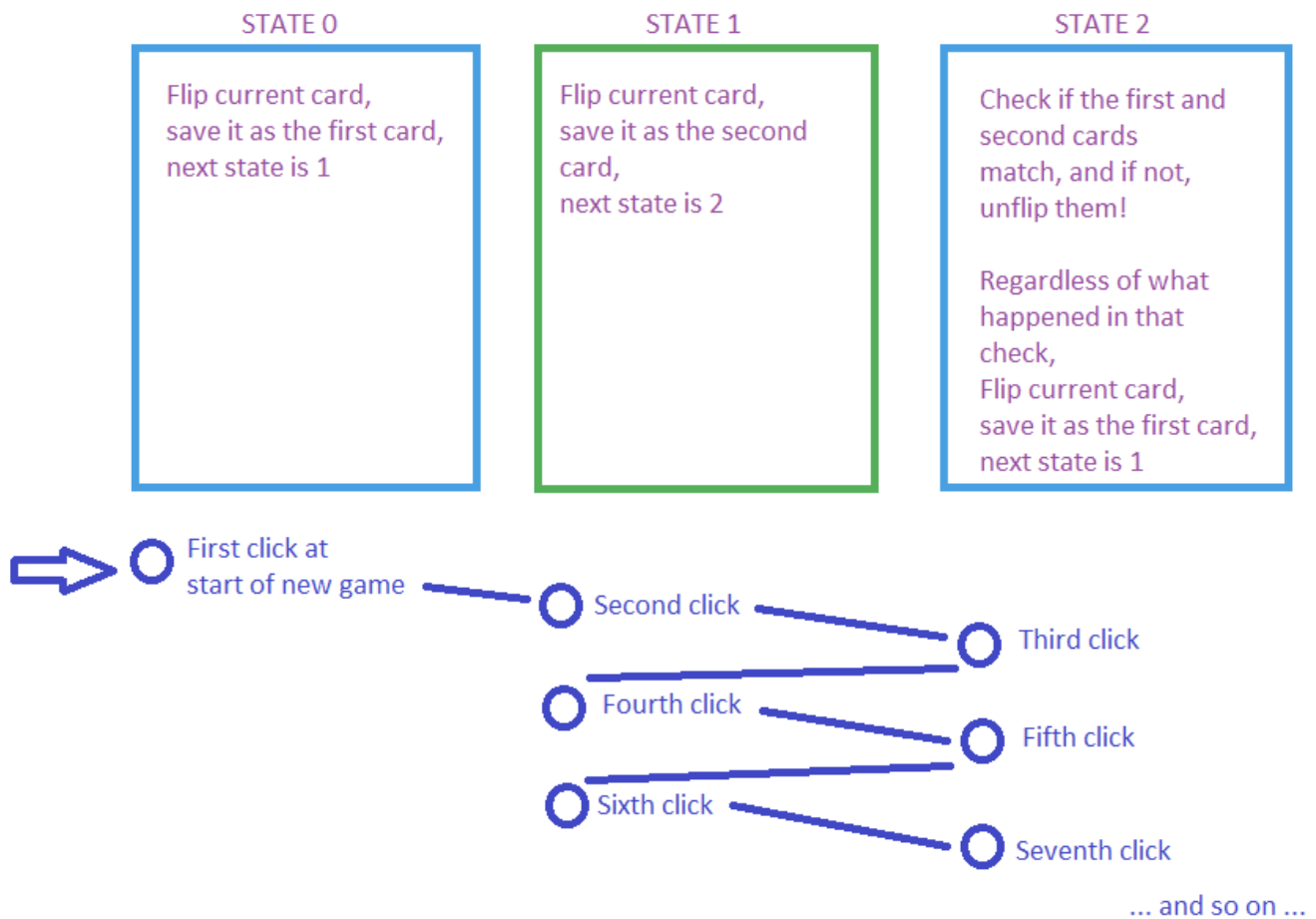
Tip #4 - Bonus tip for images

If you intend to use images in your mini-project, please remember to test your finished mini-project on a separate computer to make sure that your images will load successfully for your peers when they assess your project. It's your responsibility to make sure that the images load successfully.

Also, I would suggest that you hard code the size of your images into your mini-project. If you compute `my_image.get_width()` and `my_image.get_height()` before `my_image` has finished loading, these calls will return size zero. If you use a source rectangle of size zero in subsequent calls to `draw_image(my_image, ...)`, the call will throw a DOM error.

Tip #5 - A visual explanation of Memory's states (from post by Andrea)

I am seeing a lot of confusion over how the suggested implementation using states works in Memory. I suggest starting by reviewing [this bit of code](#) provided in the Memory lecture. Then I suggest that you review diagram below which is designed to show the the sequence of states that occur as the player clicks through the game. This is what steps 7 and 8 are asking you to code.



Tip #6 - Drawing cards using the list **exposed** and a **for** loop (from post by Rebecca)

I have seen lots of people confused by how to loop through the indices of the cards and draw the number or the blank card depending on the value of that card in the exposed list. I've written a version of the memory program that visualizes the indices of the cards, the values of the card list and the value of the exposed list, so you can see how these relate to each other, and how you can access them using the index of the cards. I hope this helps someone! Here is the start of the game: the game is in the first row, the card index is in the second row, the values of the cards in the third row, and the values of exposed in the fourth row. At the start of the game, all cards are face down, so the value of each element of exposed is False. (This is state 0.)

The game															
Index of cards (i): this is what is looped through in for i = range(0, 16):															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Values of shuffled card deck list. Access these in the for loop using card_deck[i].															
1	7	4	5	6	2	4	6	7	0	0	2	3	5	3	1
Values of exposed list. Access these in the for loop using exposed[i].															
False	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False

Below the first card has been clicked. The value of exposed[1] is True, so the draw handler draws the number at card_deck[1]. (This is state 1.)

The game															
	7														
Index of cards (i): this is what is looped through in for i = range(0, 16):															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Values of shuffled card deck list. Access these in the for loop using card_deck[i].															
1	7	4	5	6	2	4	6	7	0	0	2	3	5	3	1
Values of exposed list. Access these in the for loop using exposed[i].															
False	True	False	False	False	False	False	False	False	False	False	False	False	False	False	False

Below a second, non-matching card has been clicked. The value of exposed[1] and exposed [3] are True, so the draw handler draws the number at card_deck[1] and card_deck[2]. All other values of exposed are False, so blank cards are drawn for these card indices. (This is state 2.)

The game															
	7		5												
Index of cards (i): this is what is looped through in for i = range(0, 16):															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Values of shuffled card deck list. Access these in the for loop using card_deck[i].															
1	7	4	5	6	2	4	6	7	0	0	2	3	5	3	1
Values of exposed list. Access these in the for loop using exposed[i].															
False	True	False	True	False	False	False	False	False	False	False	False	False	False	False	False

Below the next unexposed card is clicked, the values of exposed in the first two cards are changed back to False. The new card index is 5, and now only exposed[5] is True in the exposed list so only card_deck[5] is drawn. (This is state 1.)

The game															
					2										
Index of cards (i): this is what is looped through in for i = range(0, 16):															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Values of shuffled card deck list. Access these in the for loop using card_deck[i].															
6	1	0	3	4	2	6	1	5	4	7	3	2	7	5	0
Values of exposed list. Access these in the for loop using exposed[i].															
False	False	False	False	False	True	False	False	False	False	False	False	False	False	False	False

Below when there is a match, the corresponding elements in exposed are changed to True. (This is state 2.)

The game															
					2						2				
Index of cards (i): this is what is looped through in for i = range(0, 16):															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Values of shuffled card deck list. Access these in the for loop using card_deck[i].															
1	7	4	5	6	2	4	6	7	0	0	2	3	5	3	1
Values of exposed list. Access these in the for loop using exposed[i].															
False	False	False	False	False	True	False	False	False	False	False	True	False	False	False	False

Below when another exposed card is clicked, with index 2, the exposed values for the matched cards stays as True, and exposed[2] is also changed to True, so the draw handler now draws the numbers at card_deck[2], card_deck[5] and card_deck[11]. (This is state 1.)

The game															
		4			2						2				
Index of cards (i): this is what is looped through in for i = range(0, 16):															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Values of shuffled card deck list. Access these in the for loop using card_deck[i].															
1	7	4	5	6	2	4	6	7	0	0	2	3	5	3	1
Values of exposed list. Access these in the for loop using exposed[i].															
False	False	True	False	False	True	False	False	False	False	False	True	False	False	False	False