CS2110 Lecture 37 April 21, 2021

- HW 9 due Sunday
- Quiz 4 Friday in class

Last time

- More basic GUI examples

Today

- Discussion of HW9 specification and demo of solution
- Other widgets
    - Radio buttons: radioButton.py
    - Slider: slider.py
- Scores so far
- Quiz 4
- Preparing for the last two weeks of the course

HW1	HW2	HW3	HW4	HW5	HW6	HW7	HW8 (14165		DS1	DS2	DS3	DS4	DS5	DS6	DS7	DS8	DS8x		Quiz 1	Quiz 2	Quiz 3		HW	DS	Quizzes		POINTS	%
6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00		3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00		15.00	20.00	20.00		48	24	55		127	100
6.00	6.00	6.00	6.00	6.00	6.00	6.00	5.00		3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00		14.00	20.00	20.00		47.00	24.00	54.00		125.00	98.43
6.00	5.00	4.00	6.00	6.00	6.00	6.00	6.00		3.00	3.00	3.00	2.00	3.00	2.00	3.00	3.00	3.00		14.00	20.00	20.00		44.00	22.00	54.00		120.00	94.49
5.00	6.00	4.00	5.00	6.00	4.00	5.00	6.00		3.00	2.00	2.00	3.00	3.00	2.00	3.00	3.00	3.00		14.00	16.00	20.00		41.00	21.00	50.00		112.00	88.19
5.00	5.00	4.00	4.00	5.00	6.00	5.00	5.00		3.00	3.00	2.00	3.00	3.00	3.00	3.00	3.00	3.00		3.00	14.00	20.00		39.00	23.00	37.00		99.00	77.95
5.00	6.00	5.00	3.00	6.00	2.00	5.00	4.00		3.00	3.00	0.00	2.00	3.00	0.00	0.00	3.00	0.00		15.00	20.00	13.00		36.00	14.00	48.00		98.00	77.17
4.00	6.00	4.00	0.00	6.00	5.00	0.00	6.00		3.00	1.00	3.00	0.00	1.00	3.00	2.00	3.00	0.00		11.00	20.00	17.00		31.00	16.00	48.00		95.00	74.80
5.00	5.00	3.00	0.00	0.00	2.00	6.00	6.00		3.00	3.00	3.00	2.00	3.00	3.00	3.00	0.00	0.00		14.00	19.00	14.00		27.00	20.00	47.00		94.00	74.02
4.00	5.00	3.00	2.00	4.00	6.00	4.00	6.00		3.00	3.00	3.00	3.00	3.00	3.00	3.00	0.00	3.00		8.00	11.00	12.00		34.00	21.00	31.00		86.00	67.72
4.00	6.00	2.00	2.00	3.00	2.00	6.00	2.00		3.00	3.00	3.00	3.00	2.00	0.00	3.00	2.00	3.00		14.00	13.00	9.00		27.00	19.00	36.00		82.00	64.57
5.00	5.00	3.00	0.00	6.00	5.00	0.00	1.00		3.00	2.00	2.00	3.00	0.00	2.00	3.00	0.00	0.00		7.00	17.00	17.00		25.00	15.00	41.00		81.00	63.78
5.00	5.00	5.00	2.00	6.00	5.00	4.00	2.00		3.00	3.00	0.00	2.00	0.00	2.00	3.00	1.00	0.00		6.00	8.00	16.00		34.00	14.00	30.00		78.00	61.42
2.00	5.00	4.00	4.00	4.00	4.00	4.00	3.00		3.00	2.00	2.00	2.00	2.00	3.00	3.00	3.00	0.00		8.00	12.00	7.00		30.00	20.00	27.00		77.00	60.63
3.00	4.00	2.00	3.00	4.00	5.00	3.00	3.00		3.00	2.00	2.00	2.00	2.00	3.00	2.00	1.00	3.00		10.00	8.00	10.00		27.00	17.00	28.00		72.00	56.69
4.00	4.00	0.00	3.00	4.00	5.00	4.00	5.00		3.00	3.00	2.00	2.00	2.00	2.00	3.00	3.00	3.00		6.00	4.00	8.00		29.00	20.00	18.00		67.00	52.76
3.00	5.00	0.00	2.00	4.00	2.00	5.00	3.00		2.00	3.00	2.00	2.00	2.00	2.00	2.00	1.00	3.00		8.00	6.00	10.00		24.00	16.00	24.00		64.00	50.39
4.00	3.00	0.00	0.00	4.00	2.00	0.00	0.00		3.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00		5.00	5.00	8.00		13.00	5.00	18.00		36.00	28.35
3.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00		2.00	2.00	2.00	0.00	2.00	0.00	0.00	0.00	0.00		3.00	4.00	0.00		4.00	8.00	7.00		19.00	14.96

Friday

- Quiz 4
  - A question on binary search and/or sorting
  - A question involving graphs (similar to the one on the next slide)
  - Perhaps another basic question like on prior quizzes - involving loops, lists, strings, and/or dictionaries

b. Write function createComplementGraph that takes as input a dictionary representation of a graph (as in part a) and returns a new dictionary representing the complement graph of the input graph. For a graph G, the complement graph has the same vertices, and a set of exactly those edges *not* present in G. Remember that for a directed graph, a node can have an edge to itself.

For example, if G is

```
{"R" : ["G", "B"],
 "G" : ["G", "R", "B"],
 "B" : ["B"]}
```

```
createComplementGraph(G) might return:
```

```
{"R" : ["R"],
 "G" : [],
 "B" : ["R", "G"]}
```

Note: I say "might return" because the specific order of the nodes in the adjacency lists is not important.

Note: you can do this problem without really knowing anything about graphs. It is really just a dictionary problem.

```
def createComplementGraph(g):
```

To access Google APIs in DS10, HW10, and HW11 you will need an API key

Two options (***option 2 provides 1 "make-up" point that will take the place of a homework or exam point you missed***)

1. Use my API key.
2. Use your own API key. To do so:
   a) Sign up for Google Cloud Platform "free tier" account. $300 free GCP services plus free first $200/mo of various maps services https://cloud.google.com/free/docs/gcp-free-tier
   b) Enable APIs and get an API key via your GCP Console: https://console.cloud.google.com. This GCP console and processes can can be quite confusing. Your goal is to create an API key with two APIs enabled: Static Maps and Geocoding. The next slide give detailed steps that should work.

YOU **WON'T** BE CHARGED! (If you use all the free credit – *I never have used it all* - it just disables the account and asks if you want to upgrade)

This might seem complicated but it's worth learning how to do these things.