Program #1 - Wheel of decision

Prof Bill - Jan 2020

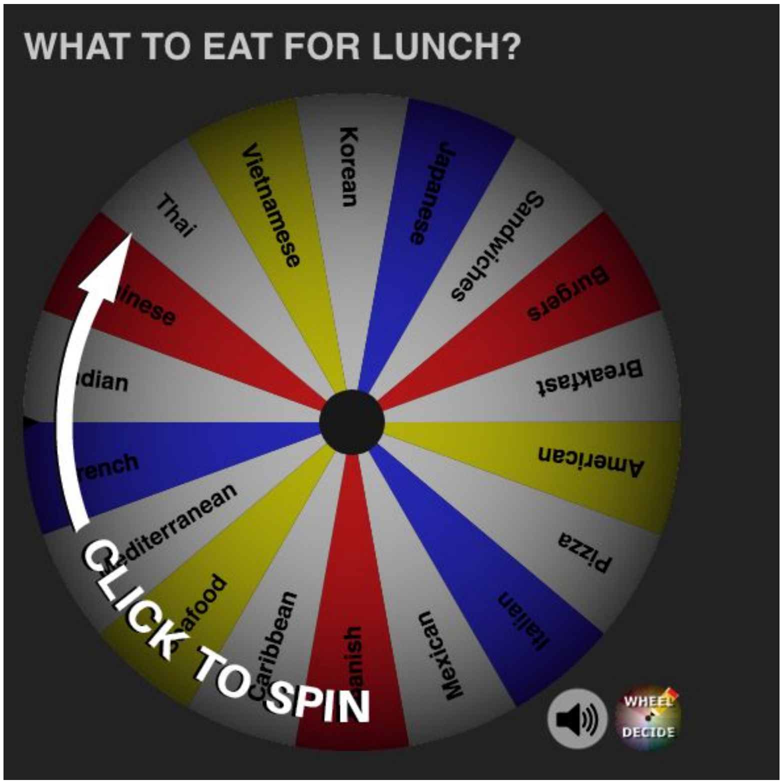
Program #1 logistics:

- Due: **Fri Jan 31, 2020** at the beginning of class (3 weeks)
- Worth: **8 points** (8% of your grade)
- Learn: Java, linked lists, debugger

1. Description

Let's code up our own Wheel of Decision, wheeldecide.com. Then, we can spin the wheel to randomly make important career and relationship choices. (smile)

We'll write Program #1 in Java and run it from the command line (text, no graphics). thanks… yow, bill



2. Commands

Your wheel program will run in the console. So, you'll type in your commands. Here are the commands your wheel should support.

Command <args>	Description
`add <item>`	Add item to the wheel (end of the list)
`name <wheel>`	Name your wheel (optional, what is default?)
`print`	Print wheel items (in order)
`size`	Print number of items in wheel
`first`	Print first wheel item
`last`	Print last wheel item
`random`	Print a random wheel item
`spin`	**Spin the wheel, show special effects, select item at random, remove it, show item**
`reload`	Reload the wheel back so original items are available
`reverse`	Reverse the items in the wheel list
`clear`	Clear all items from the wheel
`read <file>`	Read a wheel file*
`save <file>`	Save your wheel to a file*
`exit`	Exit the program

* We'll define a wheel file format. It will be simple, plain text. Something like... first line is the name, each remaining line in a wheel item. This may be a good feature to do last.

Little help. I will be running alongside you with helpful notes and snippets here:

Program #1 Helper (gdoc)

3. Requirements

Program #1 requirements are:

- ➢ Write your program in **Java**.
- ➢ Please write quality code: Java coding guidelines
- ➢ Use **VS Code** to edit/debug your program. We're trying something fun/new.
- ➢ Code your own **doubly-linked list** and implement the **wheel commands**.
- ➢ Flex **your creativity** in two ways…
  - ○ Cool "spinning" effect in your console
  - ○ Create one new/unique command or feature on your own.

4. Details

We'll work on this in class. More on the operation of your wheel:

- Design: What are the classes here? What should main() look like?
- Your "spin" should have some special effects related to it. Obviously, we can't show a spinning wheel, but we can print "spinning…" and add some delay. Your choice… get creative here.
- Commands with params are a little tricky. I recommend: Get the command, then ask for the parameter on a separate line. Then, you won't have to parse strings.
- Don't worry about elaborate error messages or checking. Do something logical. Don't crash and burn.
- Command shortcut? It looks like the first two letters are unique. Kust cayin.

How to succeed (writing any program):

1. Start early!
2. Don't be shy. Ask a question in class. Email me. Come to office hours.
3. Small bites. Divide and conquer your program into small, manageable tasks.
4. ABW. Always be working. Your program should always compile and run. Never leave your work in disarray.

5. Grading

To submit your work, create a **program1** folder on your k: drive.
This folder should contain:

- All your Java source files
- Your program1 executable
- Any wheel files that you have saved/created
- A **README.txt** file where you describes: 1) the status of your program, 2) the creative elements of your program, and 3) how to run it.

Remember our **plagiarism** guidelines as well. Getting help from google or stackoverflow or a friend is OK, but:

1. You must acknowledge any help you receive with a comment in your code
2. You must understand any code in your solution
3. Get help on program components, not the assignment (the tic tac toe philosophy)
4. If you have any questions in this area, contact me **before** you turn in your work, not after (when it's too late)

All your code must follow our class **Coding Guidelines**. An ugly design/code will be severely penalized. A program that doesn't even compile is probably worth 0 points.
thanks… yow, bill

PS - Program #2?