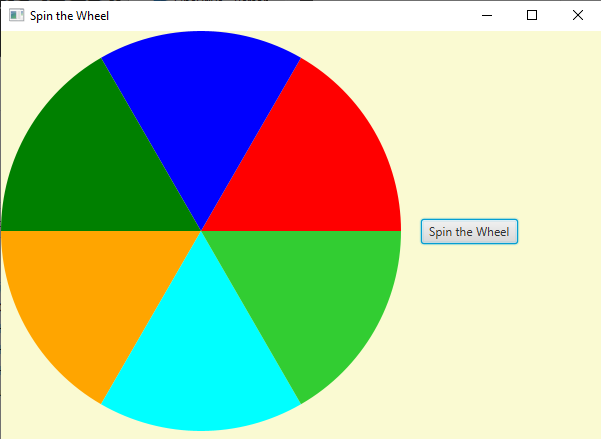
**CSE 220 Homework Assignment 3 (Due 3/11/22)**

For this homework you will be constructing a simple JavaFX GUI that includes a few basic graphics and/or shape objects, has a button or similar control that performs an action when used, and performs a brief animation on demand.

There will be some flexibility in what you want to design (see options at the bottom of the document), but the easiest and “default” option is outlined first:

Build a JavaFX-based class SpinningWheel.java with the following scene specifications:

1. The class should display a colored wheel with at least 6 distinct colors assigned to at least 6 wedges of the wheel.
2. A button or other control should be present in the scene.
3. When the above control is interacted with, the wheel must “spin” for at least 2 seconds, and complete at least two full revolutions (i.e. 720+ degrees).

A plain example of such a GUI is depicted below (snapshot only, of course).  


Note that this the barebones requirement project, and while it is not tremendously difficult, it can require some work if you’re new to GUIs, controls, and animations. It is heavily recommended that you refer to slides and sample code when constructing your GUI. You may also want to consider one of the extra credit opportunities at the document’s end (*you may only complete one*).

**When finished, upload all .java files to Blackboard in a zipped format with the filename “LN\_FN\_3.zip” where LN is your last name and FN is your first name. Include a short/minimal README telling me which file has the main class to run.**

Extra Credit Opportunity:

For those whose exam scores did not meet their expectations, you can improve them a bit by completing one of the tasks below (note that this opportunity is worth proportionally more credit to those with a lower grade):

1. Construct the SpinningWheel problem as above, and include i. an arrow indicating a “chosen position” on the wheel (AKA a “flapper”), ii. make the degree of rotation when the control is used random, iii. assign a point total to each of the wedges, and iv. cumulate the point total as spins occur. Note that this is trickier than it may seem, because you’ll need to track the location of wedges *after* spins – I recommend looking into “setOnFinished” to perform calculations after the wheel finishes spinning and “getRotation” to track which wedge is currently chosen.
2. Construct an “Elevator” GUI that moves an elevator car/cab from one level of a “building” depending upon either i. A number being entered into a text box ii. A specific button (out of many) being pressed. Note that your elevator and floors should be reasonably well-defined (i.e. not basic rectangles).
3. Construct a “Transport Robot” GUI that has a pile of some objects on one side of the screen, and shows the robot carrying one object from that side of the screen to another whenever a button/control is used. (Note that is likely the most difficult of the tasks).