

**CS 8, Winter 2015**  
Homework Assignment #? (draft)

### **Assignment Overview**

This assignment focuses on the design, implementation and testing of a Python program which uses file processing, dictionary, string, and Turtle graphics to draw a pie chart.

### **Assignment Specifications**

A pie chart is a type of graph in which a circle is divided into sectors that each represents a proportion of the whole, often with adjacent sectors filled in with different colors and with a text label for each sector to delineate what the sector represents.

We will plot pie charts based on two data files that record the popularity of games from a random survey (game\_ball\_over\_net.txt and game\_catch.txt). Each line in these files records the name of a game followed by an integer representing the number of participants in the game according to the survey. Note that the name of the game can be a multi-word string with spaces, e.g., “Volleyball Beach volleyball,” and other symbols, e.g., “Rundown (aka Pickle).”

### **Assignment Deliverables**

The deliverable for this assignment is the following file:

pie\_chart.py – the source code for your Python program

Be sure to use the specified file name and submit it for grading via the **turnin** system before the project deadline.

### **Assignment Notes**

1. Pie chart is just a circle with lines in it to represent values. Turtle does have a circle function in it that can draw a circle or part of a circle – if you care to use it.
2. You should use only six colors: red, green, blue, yellow, magenta and cyan for different sectors in a pie chart. If you have more than six sectors, loop over these colors.
3. You need to normalize and convert the participants into certain percentages of occupation along the circle. To avoid cluttering the pie chart, if a sector is less than 10° across, do not draw it! Instead, delete this game and accumulate its size into a sector named “all others.”
4. The actual drawing of a circular sector should be coded in a function that, when supplied with the angular coverage, color and a text string, will draw the particular sector.
5. As dictionary does not have a fixed ordering, it is not critical if these sectors follow a fixed order (e.g., the order they appear in the input file). Furthermore, for small sectors, it is ok that the text string may be overwritten by adjacent sectors.

6. Your program should comprise the following functionalities: open and parse input files into a dictionary, compute sector locations and coverages based on numbers of participants in the survey, draw and fill sectors, and write the text strings. All these functionalities should be invoked appropriately by one function called “chart” which accepts one single argument that is the name of the input file.

**Sample Output:**

