CS 1110 A2 Grading Guide

February 20, 2015

1 Windchill

1.1 Grading Guidelines

20 Points Total

- Test cases 1 point for each one they pass (10 points) (automated)
- 6 points if their test cases cover different representative situations
  - Total of seven possible situations
    - temperature ends in 'F'
    - temperature ends in 'C'
    - temperature is just a number string
    - wind speed ends in 'mph'
    - wind speed ends in 'kph'
    - wind speed is just a number string
    - temperature is greater than 50 OR wind speed is less than 3
  - 0 points for including only one of these seven (cases aren't representative at all). 1 point for each additional case.
- 1 point for writing at least 10 test cases (regardless of representativeness)
- 1 point for including module docstring
- 1 point for including WindChill function docstring
- 1 point for at least one non-skeleton `#` comment in WindChill (regardless of quality)
- -1 point for manually writing `y = 1.8*x + 32` instead of `y = Weather.to F(x)`
- -1 point for leaving print statements in WindChill

To run the grading bot, place the Submissions folder (that has all the netid folders inside) inside the a2grade folder, then run autoGradeBot.sh from the command line. It will output several lines of text, indicating how many of the 10 test cases each of the submissions passed. Copy and paste this into the feedback.

Cases to watch out for:

- If a student forgets to round, the grading bot will give them 0/10 or 1/10. If this happens, fix the single line and see what the grading bot gives them now, then take off 1 point for forgetting to round. For example, if a student gets 1/10 but fixing the line gives them 9/10, they will get 8/10 for this portion of the GenWeather grade.
- If a student's submission crashes the bot, check if one or a few lines fixed in their submission will solve the problem. They lose 2 points for each line fixed this way.
- Most trickily, per the above, there are several ways for students to make a one-line typo that causes the automated grader to take off more than 2 points. For example, trying to convert a temperature from Celsius to Celsius instead of Celsius to Fahrenheit causes 4 of the 10 test cases to fail. Please make sure to check all of these so that students can get the most credit possible; in this case the student would get 8/10 (from the 2-point penalty to fix 1 line in a bugged program) instead of the automated grader's 6/10. Ideally, make single-line fixes until the student's program is perfect, and check if this score (with -2 for each line fixed) is lower or higher than what the automated grader outputs, and give the higher one.

1.2 Sample Solution

```
# GenWeather.py
# Charles VanLoan (cfv3)
# February 4, 2015
""" Contains a function for computing windchill
with inputs that are either metric or English and
an Application Script that tests for correctness.
"""

import Weather

def WindChill(T,W):
    """ Returns the windchill associated with the temperature
    and wind that are encoded in T and W.

    If T represents a Celsius temperature, then the returned value
    represents windchill in Celsius. If T represents a Fahrenheit
    temperature, then the returned value represents windchill in
    Fahrenheit.

    Precondition:
        T is a valid temperature string, e.g., '30F', '-2C', '16'.
        'F' indicates Fahrenheit and  'C' indicates Celsius,If
        T is just a string encoding of an integer, it is assumed that
        the integer represents a Fahrenheit  temperature.
        W is a valid wind string, e.g., '30mph', '20kph', '16'.
        'mph' indicates miles-per-houor and  'kph' indicates kilometers
        per hour. If W is just a string encoding of an integer,
        it is assumed thatthe integer represents wind speed in
        miles-per-hour..

    """
    # Assign the Fahrenheit temperature to Temp
    n = len(T)
    if T[n-1]=='F':
        Temp = float(T[:(n-1)])
    elif T[n-1]=='C':
        Temp = Weather.to_F(float(T[:(n-1)]))
    else:
        Temp = float(T)
    # Assign the windspeed in mph to Wind
    m = len(W)
    if W[m-3:]==’mph’:
        Wind = float(W[:(m-3)])
    elif W[m-3:]==’kph’:
        Wind = Weather.to_M(float(W[:(m-3)]))
    else:
        Wind = float(W)
    # Compute the Fahrenheit wind chill temperature
    if Wind<=3 or Temp>=50:
        WC = Temp
    else:
        WC = Weather.WCF(Temp,Wind)
    if T[n-1]==’C’:
        # Input temperature was in Celsius, so convert wind chill to Celsius
        WC =  Weather.to_C(WC)

    return round(WC)

def Test(T,W,TrueWC):
    """  Prints T, W, WindChill(T,W) and the True Windchill

    Precondition: T is a  valid temperature string, W
    is a valid wind string, and TrueWC is
    the actual wind chill asociated with T and W.
    """

    WC = WindChill(T,W)
    print (’WindChill returns %4d     Actual = %5.1f    Input = (%s,%s)’
           % (WindChill(T,W),TrueWC,T,W))

#Application Script
if __name__ == ’__main__’:
    """ Confirms the correctness of WindChill in
    ten different representative cases."""
    Test(’10’, ’20’,-9.0)               # 1 point
    Test(’-5F’,’20’,-29.0)             # 1 point
    Test(’-15C’,’20’,-26.0)            # 1 point
    Test(’10’, ’30mph’,-12.0)          # 1 point
    Test(’-5F’,’30mph’,-33.0)          # 1 point
    Test(’-15C’,’30mph’,-28.0)         # 1 point
    Test(’10’, ’40kph’,-11.0)          # 1 point
    Test(’-5F’,’40kph’,-31.0)          # 1 point
    Test(’-15C’,’40kph’,-27.0)         # 1 point
    Test(’-15C’,’0’,-15.0)                # 1 point
```

2 Sun

2.1 Grading Guidelines

10 Points Total

- Draw Sun 5 points.
  - 2 points for rotating the two additional stars the right way.
  - 2 points for the correct coloring of the rays according to the parameters passed into DrawSun (0 points if parameter names not used at all, 1 point if parameter names used incorrectly).
  - 1 point for the center disk being drawn correctly.
- Nested Stars 5 points.
  - 2 points if all the stars have the correct radii. 1 point for slight errors (e.g. the alpha multiplier is applied early causing all six stars to be .62 times smaller, wrong alpha value was used, etc). 0 points if all the stars have the same radii, etc.
  - 3 points for the right "shift" of colors for the inner stars. 2 points if the shift is counterclockwise instead of clockwise. 1 point if the colors of each star are shifted in some way but neither properly clockwise nor counterclockwise. 0 points if colors are not shifted.

NOTE: Due to an inconsistency between the assignment handout and the skeleton file that was given to the students, it will be considered OK if the stars in the original Sun are drawn counterclockwise instead of clockwise (i.e. magenta, cyan, orange). The nested stars must still be drawn clockwise for full credit.

There is no grading bot for this part of the assignment, so you will have to run as well as open each Sun.py file manually. Note that students can choose their own set of 5 colors for this assignment.

As before, crashes are -2 points per line it takes to fix the crash, as are any typos that cause catastrophic cascading errors.

Loops were not intended to be used on this assignment but there is no penalty (also do not write any comment) if a student did use them.

Problems that deserve a comment but no lost points include hardcoding values for the parameters instead of using the variables given in the script portion, including alpha, the three colors, the radius, and the x and y coordinates.

2.2 Sample Code

```
#!/usr/bin/env python

from simpleGraphics import*

def DrawSun(x,y,r,c1,c2,c3):
    """ Draws a Sun with radius r centered at (x,y).

    The "rays" have colors c1, c2, and c3.

    Precondition: x, y, and r are floats or ints. The arguments
    c1, c2, and c3 are rgb arrays.
    """
    # The rays are tips of these three stars:
    DrawStar(x,y,r,c1,stroke=0)
    DrawStar(x,y,r,c2,rotate=-24,stroke=0)
    DrawStar(x,y,r,c3,rotate=-48,stroke=0)
    # Place a yellow disk to block out all but the rays.
    DrawDisk(x,y,.62*r,YELLOW,stroke=0)

#Application Script
if __name__ == '__main__':
    """ Applies DrawSun six times.
    """
```

```
r = 5.
x = 0
y=0
alpha = .62
# Ray colors
c1 = RED
c2 = GREEN
c3 = BLUE
MakeWindow(6,bgcolor=BLACK)
DrawSun(x,y,r,c1,c2,c3)
MakeWindow(6,bgcolor=BLACK)
DrawSun(x,y,r,c1,c2,c3)
r = r*alpha
DrawSun(x,y,r,c3,c1,c2)
r = r*alpha
DrawSun(x,y,r,c2,c3,c1)
r = r*alpha
DrawSun(x,y,r,c1,c2,c3)
r = r*alpha
DrawSun(x,y,r,c3,c1,c2)
r = r*alpha
DrawSun(x,y,r,c1,c2,c3)

ShowWindow()
```