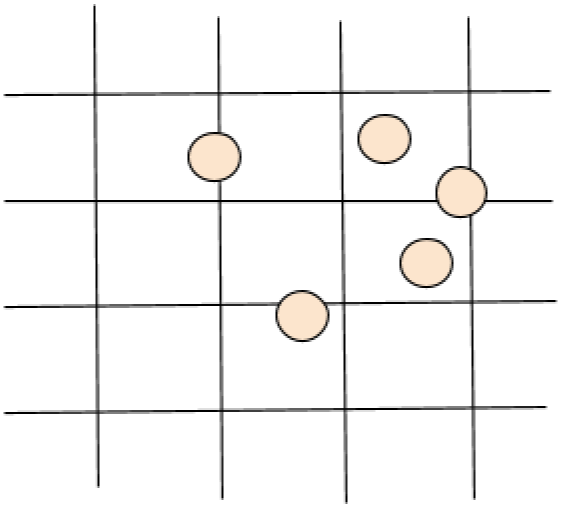
1026B: Assignment 1: Designing A New Carnival Game

Due: Wednesday, Feb. 7th, 2018 at 9:00 pm
Weight: 5%
Goal: Use Python to solve interesting real-world problem, with an emphasis on problem-solving and algorithm design skills

In many carnival games, such as Soda Toss with Rings, the owner sets up a game and people pay a small fee to play. If they “win”, they get a “reward” (such as a teddy bear). Your task is to help design a new carnival game by writing a Python program to ensure that the owner would “make a profit” in a long run.

In this new carnival game, square grids are drawn on the ground. The distance between lines is *d* mm. Players would throw the $2 toonie coins into the grid. If the toonie touches any line, they lose the toonie; if it does not touch any line, they would get a “reward” worth $*r* to the owner. The diameter of a toonie is 28 mm. See an illustration below.



For example, if d = 76 mm (which is about the size of the popular square post-it notes) and r = 4, would the owner make money? Would you play this game with your money?

You need to write a Python program to simulate the game (or build a model, run Monte Carlo). Your program would take the input as follows (the input/output of your program are shown in bold below):
Please enter the distance between lines (in mm):
Please enter the "reward" if the customer wins (in $):

Then your program would simulate 1,000 toonie tosses, and calculate how much the owner may gain or lose.

If the owner has a profit, your program would output the amount of money gained:
For 1000 toonie tosses, you may get $...

If the owner loses money, your program would output the amount of money lost:
For 1000 toonie tosses, you may lose $...

Note that this problem could be solved analytically with math, but you must run simulation (Monte Carlo) with Python to solve it. This is because many real-world problems may not have analytical solutions. The recent AlphaGo also use a Monte Carlo tree search algorithm.

Hint: you will need to use a simple loop statement (for) and a random number function called random(), which generates a uniformly distributed random number that is >= 0 and <1. Copy/paste the following Python code to your PyCharm to see the outcome:

```
# A sample program with loop and random numbers
from random import random

for i in range(10):       # loop 10 times
    print(random())       #print a random number between 0 and 1
    print(10*random())  #print a random number between 0 and 10
    print()               # print a blank line

print("Done")
```

See the last few slides in ch04 for an example of Monte Carlo to estimate the value of pi.

Experiment with *d = 76* and *r = 4* for toonie coins, and a few other values of d and r (which need not to be integers). If we also allow people to throw the $1 loonie coins into the grid, what would be a good value of r (if d = 76) so that the owner makes a profit in a long run?

<u>What You Will Submit and Be Marked On:</u>

1. A 1-2 page written part (with Word, text, PDF file) about:

- A brief problem-solving process you use in designing and implementing your Python programs
- The output of your program when d = 76 and r = 4 for toonies. You can copy/paste from the Output window. Show a few different sets of d and r for toonies, some of which make money, some lose money
- Find a good r (when d = 76) for loonies. Write down your reasoning and problem-solving process using your Python program.
- Discuss options and give your final recommendation to the owner.

2. Your Python source file. The name of the Python program you submit should be your UWO userid_Assign1.py. Make sure you attach your Python file to your assignment; DO NOT put the code inline in the textbox or the written part above. Make sure that you develop your code with Python 3.6 as the interpreter. TAs will not endeavor to fix code that uses earlier versions of Python.

3. Non-functional specifications: as described below

Non-functional Specifications:

1. Include brief comments in your code identifying yourself, describing the program, and describing key portions of the code.
2. Assignments are to be done individually and must be your own work. Software may be used to detect cheating.
3. Use Python coding conventions and good programming techniques, for example:
    i. Meaningful variable names
    ii. Conventions for naming variables and constants
    iii. Use of constants where appropriate
    iv. Readability: indentation, white space, consistency