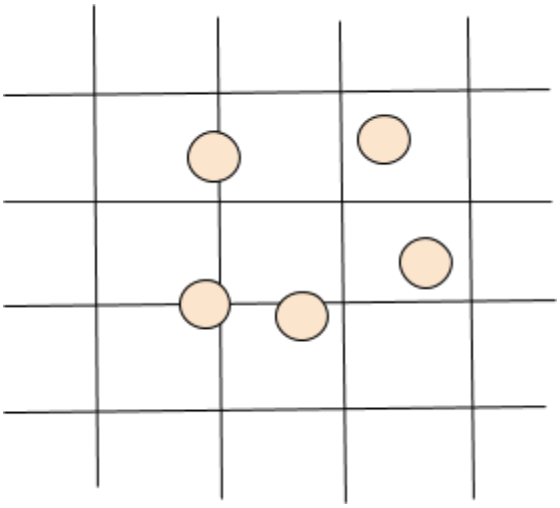
1026B: Assignment 1: Designing A New Carnival Game

Due: Wednesday, Feb. 6th, 2019 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 simulate the new game 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 (of very large size). If the toonie touches or crosses any line, they lose the toonie; if it does not touch any line, they would get a "reward" worth $*r* from the owner (note, the owner still keeps the toonie in this case). See an illustration below.



For example, if **d = 70 mm** (which is about the size of the popular square post-it notes) and **r = 6** (taking away $2 bet, the player gets $4, or doubles his/her bet), would the owner make money? In this case, three coins touch the lines and two do not. The owner would end up with 5*2 - 2*4 = $2, a small gain.

You need to write a Python program to ***simulate*** the game. Your program would take the input as follows (let's assume that input is always valid, and the input/output of your program are 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 count how many times the toonies touch or not touch the lines, and then how much the owner would gain or lose at the end of the 1000 toonie tosses. Note the diameter of a standard tonnie is **28 mm**.

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

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

Note that this problem could be solved analytically with math without simulation, but in this assignment you must run simulation with Python to solve it. This is because many real-world problems are too complex to have analytical solutions. The recent AlphaGo also use a Monte Carlo tree search algorithm (Monte Carlo means simulation) because it is too huge to analyze the whole game trees.

Hint: To simulate the random landing of the toonies, 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(100):      # loop 100 times
    print(random())       #print a random number between 0 and 1
    print(20*random())  #print a random number between 0 and 20
    print()               # print a blank line

print("Done")
```

Although the code above can generate random numbers between 0 and any given number, you need to think hard on how to use this to simulate a random coin toss, or a random coin landing. Also, how to deal with the grid? How to simplify the problem? This is the problem solving and abstraction that we emphasize in the lecture.

See the last few slides in ch04 for another example of running simulations to estimate the value of pi.

What You Will Submit and Be Marked On:

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 = 70 and r = 6 for toonies. You can copy/paste from the Output window. Show a few different outcomes of d and r, some of which the owner makes money, some loses money
• Discuss what good values d and r for the owner would be in your opinion? Write down your reasoning supported by the outcome of your Python program.
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 3.7 as the interpreter. TAs will not endeavor to fix code that uses earlier versions of Python.

Note: The assignment is to be done individually and must be your own work. Software may be used to detect cheating in your Python codes.

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