



EXAM PROJECT

EXAM PROJECT

TIMELINE

- 26/01 11AM : Conference call on the project
- 29/01: Deadline for **code** and **individual paper** (3-5 pages):
 - Put code on github and look at timestamp
 - Mail Timestamp AND individual paper to gertdegeyter@gmail.com AND to your two peer evaluators (see the excel cheat)
 - **Code can not be adjusted after**
- 05/02: Deadline for **Peer-evaluation** paper and mail to gertdegeyter@gmail.com

EXAM PROJECT

- You can work on it either alone or in duo's
You will not be quoted differently if you make the project alone
- You can work on the code together, however, I expect an individual paper per person
- The code should be available on **GitHub!!!!!!**
- The project deadline is Sunday evening 29th of January (NO EXCEPTIONS!)
- The peer evaluation deadline is 5th of February
- The points are distributed as follows
 - 8 points on the code
 - 8 points on paper
 - 4 points the peer-evaluation

EXAM PROJECT

Code

- The code should be placed on github (one of the duo's is enough)
(I recommend also using it and not just putting it there in the end)
- Use the programming types (FP & OOP) and keep a clear structure in your code
- DOCUMENT(!!) your code
it should be clear what you do by just reading the actual code
- Try to follow the pep8 coding style guide
(pycharm will let you know as well)
- Should be handed in 29th of January together with the INDIVIDUAL paper
- Can not be adjusted after

EXAM PROJECT

Paper

- SHOULD CONTAIN THE TIMESTAMP OF YOUR GITHUB SUBMISSION!!
- Around **3-5 pages** (not including title page and references) explaining how you solved the problem
- Graphs of results, Diagrams of your program structure ...
- Should contain references to sites, papers, etc. that you have used
- Send it to instructor AND two peer evaluators

EXAM PROJECT

Peer Evaluation

- Everyone will receive two names
- After the deadline of the project everyone will have to evaluate 3 people
 - 2 fellow students:
 - Write about half a page per student on things you've liked and disliked
 - Keep in mind everything (paper, coding style, method, structure,...)
 - Give a grade 10 on project and 10 on the paper
 - Self assessment:
 - Now that you have seen the other projects how would you grade yourself?
 - What did you do well and what would you do different now?
 - Grade yourself as well
- The grades you give will NOT be used

The background is a dark gray gradient. On the left side, there are several concentric circular patterns. A large circular scale with tick marks and numbers (140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260) is visible. There are also smaller circles with arrows indicating a clockwise direction. The overall design is technical and modern.

PROJECT GOALS

PROJECT

Goals

In this project we will simulate a casino that generally can be seen as consisting out of games, employees and customers. We will start by simulating two games before going to the complete casino and defining the habits of customers and employees.

I suggest you first read the entire assignment, think of how it would look like in the end and then slowly start with something small. I advice you to work incrementally, write little parts of code and test them. Document what you did and make sure to occasionally track your changes. You can write your process in the paper, this doesn't only have to cover the entire casino project. Feel free to put in temporary results as well.

Make sure to look at all exercises again (especially the last one in session 2)

It's OK if you get stuck, just try to come up with a workable solution and explain what you did.

PROJECT

Setup

1. Start a new repository on github and give it a sensible name for this project
2. Open pycharm and setup this new repository as a project
3. Add a new file to your project called Roulette.py

(DON'T FORGET TO OCCASIONALLY COMMIT AND PUSH YOUR RESULTS!!)
(DOCUMENT YOUR CODE!!!)

4. In this Roulette.py add the following lines:

```
import random
```

```
# This is used to fixed the random generator so we can test the output
```

```
random.seed(3456)
```

Doing so will fix the random number generator so we get reproducible results
(don't worry we will turn this off later)

5. Let it print out 5 random integers between 1-10.

The output should be 7, 8, 5, 3, 9

if you get this remove this part (as it was just as a test) but leave the three lines from step 4

PROJECT

PART1 : Roulette

Now let's make a very basic Roulette game.

The roulette game should have the following properties:

- A minimum amount that should be betted. When an amount below this is betted, this results in an instant loss.
- A function AboveMinimum that takes a list of betted amounts and returns a list of booleans
- A function SpinTheWheel that has a variable 'bets' which is a list of integers ranging from 0 to 36 (both included). It should determine a random number between that range and return a list of the same length as bets indicating which bets were correct and which not. It should print out the location of the ball in the roulette wheel. If there are winners it should print out how many, if not it should be mentioned that no player won.
- When a player has a correct bet he gains the betted amount times 30.
- Create a function SimulateGame that takes a list of bets and a list of betted amounts. The function returns the total amount won by the casino (so basically lost bets) and a list with the amounts won by the players.

PROJECT

PART1: Roulette

Create a new file “Simulation.py”.

In this file you should always call the code and test the code you’ve written.

You should now import your newly created game into the simulation file. The following code should produce the output below:

Code:

```
amounts1 = [10, 85, 120, 65, 150, 122]
bets1 = [10, 24, 36, 0, 11, 24]
table1 = Roulette.Roulette(100)
print(table1.SimulateGame(bets1, amounts1))
print(table1.SimulateGame(bets1, amounts1))
```

Output:

```
Spinning the wheel...
Ball lands on 24
There are 2 correct bet(s)
[430, [0, 0, 0, 0, 0, 3660]]
Spinning the wheel...
Ball lands on 30
No winners this round
[552, [0, 0, 0, 0, 0, 0]]
```

PROJECT

PART2: Craps

Create a new file “Craps.py”. This is pretty similar to the roulette gam, however, here instead we have two dices that are thrown. We will again make a simplified version. It should have the following functions

- A minimum amount that should be betted. When an amount below this is betted, this results in an instant loss.
- A function AboveMinimum that takes a list of betted amounts and returns a list of Booleans
- A function Dices that returns the sum of randomly thrown dices
- A function RollTheDices that has a variables ‘bets’ which is a list of integers ranging from 2 to 12 (both included). It should print out the sum of the randomly thrown dices. If there are winners is should print out how many, if not it should be mentioned that no player won.
- We will discuss the amount that should be returned on win the next slide
- Create a function SimulateGame that takes a list of bets and a list of betted amounts. The function returns the total amount won by the casino (so basically lost bets) and a list with the amounts won by the players.

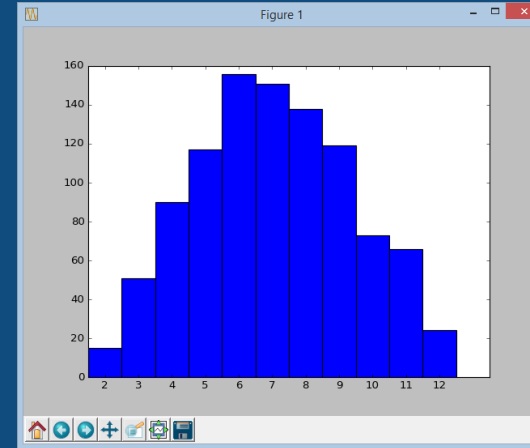
PROJECT

PART2: Craps

Open the simulations.py file again and now call your craps game. Create a list that takes the output of 1000 dice throws (so just the sum of dices, not the amount or bets). You can then use the code below to plot the distribution of the sum. You should then see something like this plot:

```
from collections import Counter
import numpy as np
from matplotlib import pyplot as plt

labels, values = zip(*Counter(thousandsthrows).items())
indexes = np.arange(len(labels))
width = 1
plt.bar(indexes, values, width)
plt.xticks(indexes + width * 0.5, labels)
plt.show()
```



It should not come as a surprise that, unlike roulette, not every combination is as likely as to be thrown. Of course this has to be reflected in the money that can be won so that when a player wins a bet when the dices are for example a total of 7, they should win less money compared to a correct bet on 12. Make a function that scales in such away that on average 90% of the money goes back to the players and only 10% profit for the casino. Prove that you have tested this. (You can unset the random seed now)

PROJECT

PART3: Casino – merging the games

Now lets make the preparations for the entire simulation. Notice that you can make more abstractions of the two games you've made before (and I advice you to do so). Try to have as little as possible duplication in your code.

There should be one function `SimulateGame` that only takes a list of betted amounts. It returns the amount won by the casino and a list of amounts won by the costumers.

The bets should also be completely randomly determined but of course adjusted to the game itself (so you can not throw 36 in a craps game).

The minimum bets for roulette tables are now randomly picked from 50\$, 100\$ or 200\$ while the craps tables have a minimum of either 0\$, 25\$ or 50\$.

PROJECT

PART3: Casino – Employees and customers

In addition to games your casino also has employees. To start they have croupiers that need to be present at every table. Every game in the casino gets a croupier assigned to it. Every croupier has a fixed wage and they get 0.5% of the amount that the casino wins every game. Of course if the casino didn't make any profit this is rounded to 0. A barman on the other hand gets a fixed wage and can keep the tips customers give when they buy drinks.

Of course every casino needs customers. Part of the customers are returning customers, these customers always place the minimum table bet. They never bet more but never less (unless they are out of money). They usually enter the evening with a random amount between 100\$ to 300\$. The one-time customers usually enter the casino with about 200\$ to 300\$ on them. They usually bet between 0 to one-third of their money at that time. Inexperienced as they are, they don't know that some tables have a minimum bet and that they instantly lose those bets. There is a special kind of one-time customers being the ones that are there on a bachelor/bachelorette party. These usually enter the casino with about 200\$ to 500\$. They bet an amount that can range from 0 to the entire amount of money they have at their disposal (at that time not starting budget). The casino has a special promotion that all of these bachelors get an additional starting budget for free (see later). Of course customers also buy drinks and tip the barmen. A drink costs 20\$ and the tip is randomly determined between 0\$ to 20\$. They buy either 1 or 2 drinks but only if they have more than 60\$ cash.

PROJECT

PART3: Casino – Taking it all together

When you create the casino you should enter the following parameters:
the number of Roulette tables, the number of Craps tables, the number of barmen, the employee wage (of one evening), the starting cash of the casino, the number of customers, the percentage of returning customers, the percentage of bachelors and the free starting budget given to the bachelors.

All of this should result in correct setup of the casino.
(it's up to you on how to write this)

The casino has one method called `simulateEvening`; In one evening the following stuff happens:

- A new set of customers is determined with the percentages roughly matching the ones mentioned during setup.
- 3 game rounds are played on all tables. Every round all customers are randomly distributed over all tables. Don't forget every table has a croupier who gets part of the profit.
- Customers go get drinks about 5 times where they get one of the random barmen available and tip him.
- The casino gets money from winning games and the sold drinks (suppose the drinks are pure profit). The casino pays the one time fee for bachelors, customers winning games and wages are paid per evening.

PROJECT

Part 4: Simulating

Now that you have setup everything let's do some simulations (in the simulation.py file)

Create a casino which has the following setup:

- the number of Roulette tables = 10
- the number of Craps tables = 10
- the number of barmen = 4
- the employee wage (of one evening) = 200\$
- the starting cash of the casino = 50000\$
- the number of customers = 100
- the percentage of returning customers = 50%
- the percentage of bachelors = 10%
- the free starting budget given to the bachelors = 200\$

Simulate 1000 evenings and show the evolution of the cash the casino earns.

PROJECT BONUS

Feel free to do just more than simulating and plotting the cashflow of the casino.

You could answer a lot of questions now:

- Should the casino invest in more craps or roulette tables?
- How much tips to croupiers and barmen usually get?
- How far can they increase or decrease the amount you win on a correct bet in craps and roulette?
- Should they focus in trying to increase the percentage of bachelors and can how far can they increase the free budget they get?
- ...



GOOD LUCK!!