**Course: Introduction to Data Science (DS2006) - Laboratory 06**

**Student: Philip Haglund**

We want to expand on our Battle of Dices project to make the Battle of Dices a multi player game.

* **Task 1: Think and reflect individually** about whatare the main limitation(s) in your current code (battle\_of\_dices\_cooler.py) to implement the multiplayer version of the Battle of the dices? (If you did this activity in class, just paste your individually submitted answer here)

Currently the issue is how we store information but we can utelise nestled lists to store it easily for X amount of players. This player amount should have a minimum of 2.

* **Task 2: Think and reflect within your group** about whatare the main limitation(s) in your current codes (battle\_of\_dices\_cooler.py) to implement the multiplayer version of the Battle of the dices? (If you did this activity in class, just paste your team submitted answer here)

I missed the lecture, the submitted content for the NoTeamYet group is:

So we have a clear limitation related to the hard-coded number of variables, so we have a hard time related to dealing with more players. Also we will have an issue related to the if-statements since we have a fixed number of cases we check for and so if we try and ad multiplayer mode we well have to accompany for how the code can vary in length with the if-statements for instance.

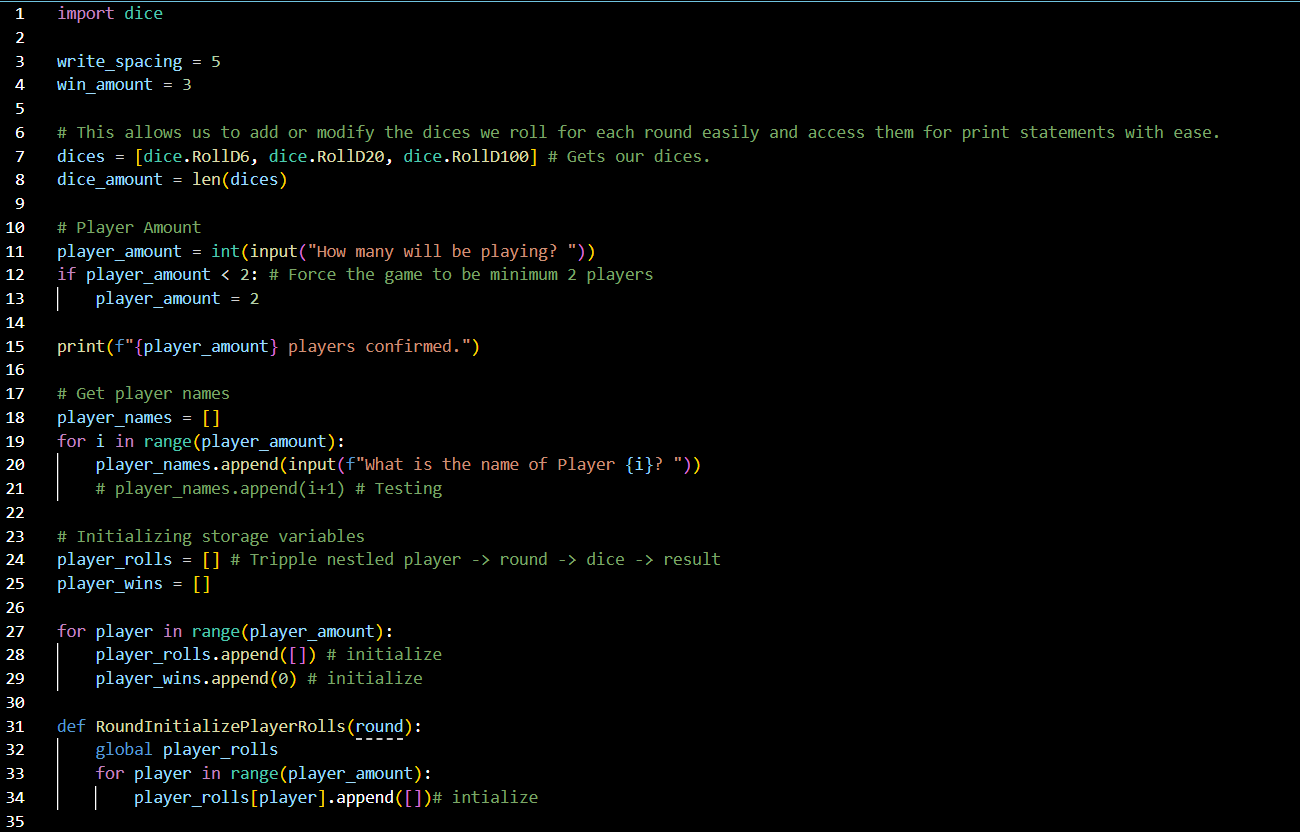
I have to agree with this statement but most of the logic can simply be fixed by using for loops with few issues.

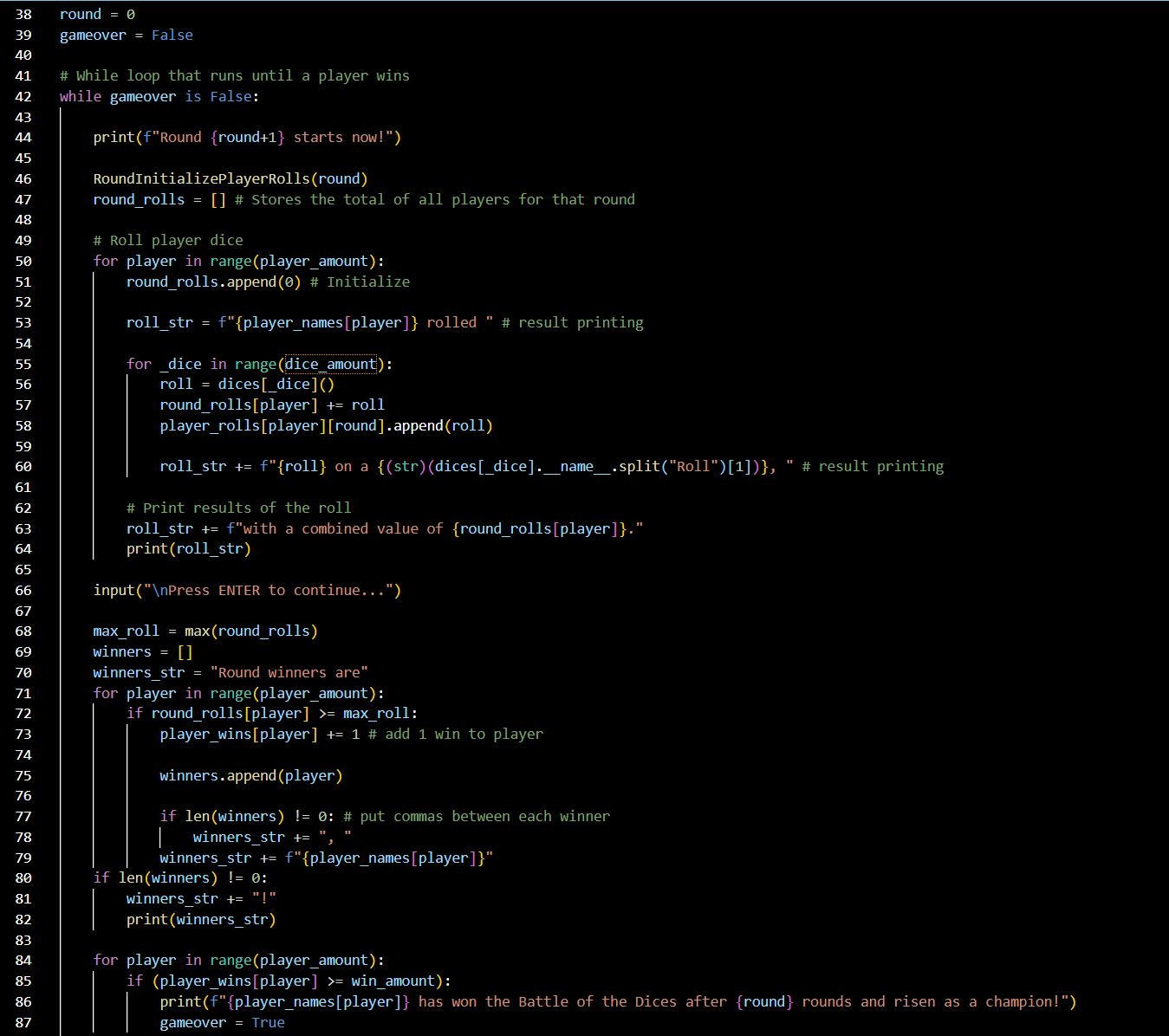
* **Task 3:** Create a file named[**multiplayer-battle-of-dices.py**](http://multiplayer-battle-of-dices.py)and implement the solution presented in the slides of Lecture 06.

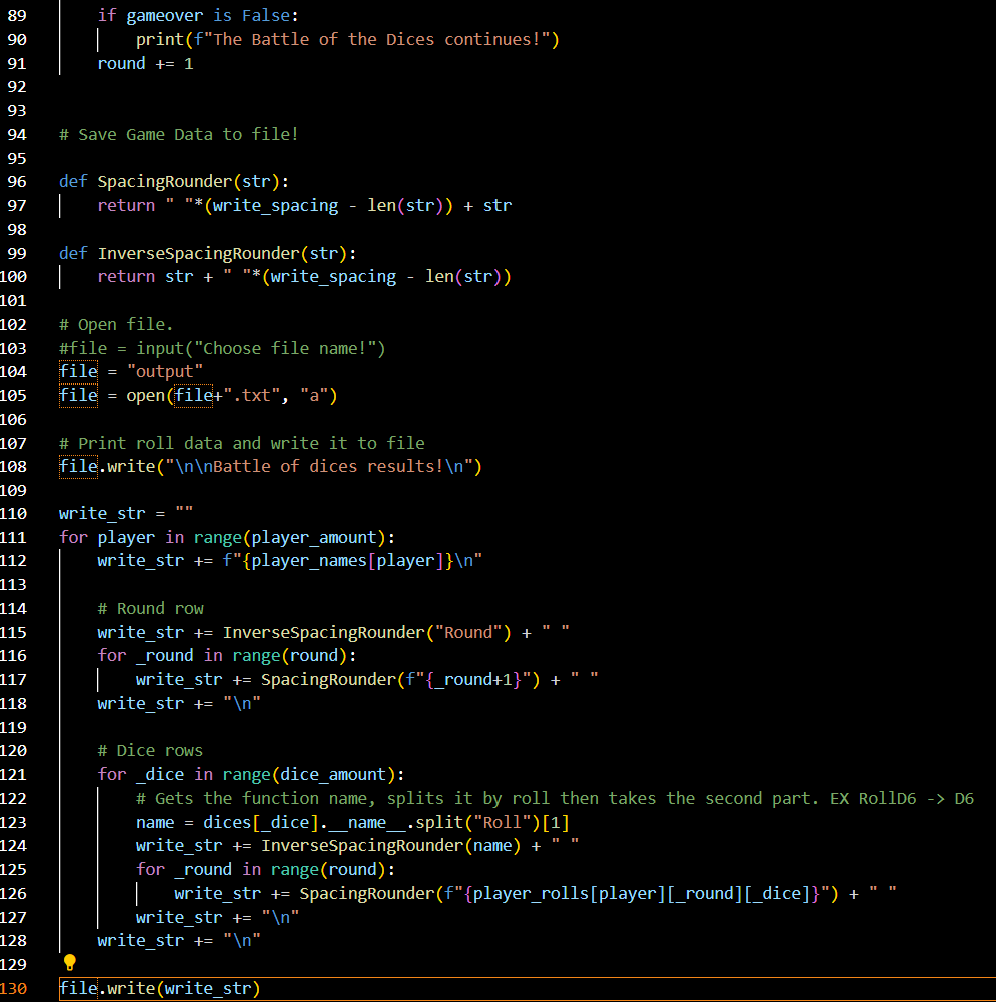
A screen shot of a computer program

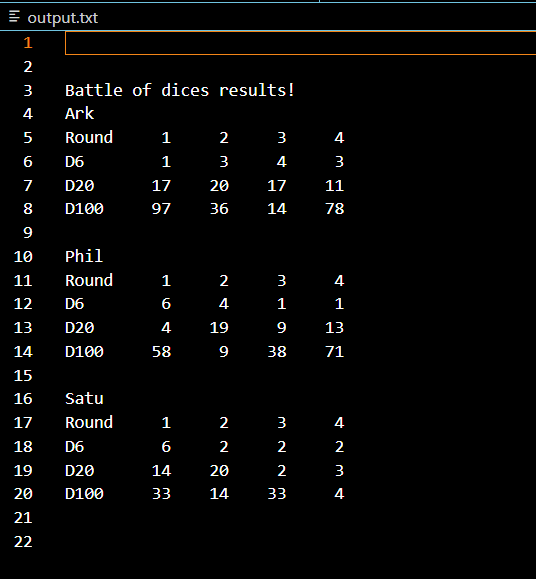
AI-generated content may be incorrect.

* **Task 4:** Create a file named[**cooler-multiplayer-battle-of-dices.py**](http://multiplayer-battle-of-dices.py)and refactor the code from Task 3 to use the same ideas from your previously developed **battle\_of\_dices\_cooler.py .**

****

****

****



Now we want to refactor the code from your **better\_calculator.py** (Laboratory 05) into a file called [calc.py](http://calc.py). In your current code you only deal with the mathematical operations of 2 numbers given as parameters. Now instead of using 2 numbers as parameters, we want new functions to receive a list as a parameter and perform the mathematical operations with the elements of the list as follows:

* **Task 5:** Make a new function that performs addition, to sum all the numbers in the list passed as parameter and return an integer with the result of this sum. For example if the input is [10, 5, 5] the return of the function is going to be 20.

A computer screen shot of numbers

AI-generated content may be incorrect.

* **Task 6:** Make a new function that performs subtraction to subtract from the first element in the list (index 0), all the values in the remaining elements in the list and return an integer with the result of this subtraction. For example if the input is [20, 10, 5] the return of the function will be 5 because 20 -10 -5 = 5.

A computer screen with text

AI-generated content may be incorrect.

* **Task 7:** Make a new function that performs multiplication, to multiply all the numbers in the list passed as parameters and return an integer with the result of his multiplication. For example if the input is [1, 2, 3], the return of the function will be 6.

**A screen shot of a computer

AI-generated content may be incorrect.**

* **Task 8:** Make a new function that performs division, to divide all the numbers in the list by 2 and return a list with the result of this division by two. For example, if the input is a list with [4, 8], the return of the function will be a list with [2, 4].

A computer screen with text and numbers

AI-generated content may be incorrect.

For the next set of activities we are going to implement some short python code snippets that were used in the Kahoot to better understand some of the things that were going on. For this part of Laboratory 06 you should create a file named [**python-revision-02.py**](http://python-revision-01.py) .

* **Task 9:** What is the output of the code shown in Figure 1?

places = ["Malmo", "Halmstad"]

places.append("Gotenburg")

print(places)

Figure 1. Code snippet

[‘Malmo’, ‘Halmstad’, ‘Gotenburg’]

* **Task 10:** What is the output of the code shown in Figure 2?

places = ["Malmo", "Halmstad"]

places.append("Gotenburg")

places.insert(0, "Curitiba")

print(places)

Figure 2. Code snippet

[‘Curitiba’, ‘Malmo’, ‘Halmstad’, ‘Gotenburg’]

* **Task 11:** What is the output of the code shown in Figure 3?

places = ["Malmo", "Halmstad"]

places.append("Gotenburg")

places.insert(0, "Curitiba")

places.append("Curitiba")

print(places)

Figure 3. Code snippet

[‘Curitiba’, ‘Malmo’, ‘Halmstad’, ‘Gotenburg’, ‘Curitiba’]

* **Task 12:** What is the output of the code shown in Figure 4?

places = ["Malmo", "Halmstad"]

places.append("Gotenburg")

places.insert(0, "Curitiba")

places.append("Curitiba")

places.remove("Curitiba")

print(places)

Figure 4. Code snippet

[‘Malmo’, ‘Halmstad’, ‘Gotenburg’, ‘Curitiba’]

* **Task 13:** What is the output of the code shown in Figure 5?

places = ["Malmo", "Halmstad", "Gotenburg", "Curitiba"]

print(places.count("Rio de Janeiro"))

Figure 5. Code snippet

0

* **Task 14:** What is the output of the code shown in Figure 6?

places = ["Malmo", "Halmstad", "Gotenburg", "Curitiba"]

places.clear()

print(places)

Figure 6. Code snippet

[]

* **Task 15:** What is the output of the code shown in Figure 7?

places = ["Malmo", "Halmstad", "Gotenburg", "Curitiba"]

places.sort()

print(places)

Figure 7. Code snippet

[‘Curitiba’, ‘Gotenburg’, ‘Halmstad’, ‘Malmo’]

* **Task 16:** Create a file named **visited\_places.py**. In this file we are going to create a software that allows the user to save a list of all the cities they have visited. Your code should have a “menu” with the options shown in Figure 8 (feel free to change the layout of the menu):

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* My Travel List \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* Please choose one of the following options: \*

\* \*

\* (1) Add a new city to the list of visited cities. \*

\* (2) View your list of visited cities. \*

\* (3) Sort the list of visited cities. \*

\* (4) Shows the number of visited cities \*

\* (5) Remove a given city from the list of visited cities \*

\* (6) Remove all cities from the list of visited cities \*

\* (7) Save the list of visited cities to a file. \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Figure 8. Example menu.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.