

A3 – Gr 21

Kasper s194883, Julie, s194891, Jakob s194886

Aim

The aim of this A3 is to provide support for the structural Bim analyst groups by providing feedback on their A3 scripts.

All the structural A3 BIM tools are public and can be used by anyone, so we function as test unit to see if their code is easy to use for others.

Feedback

In general, I recommend most groups to add a short paragraph at the top of their scripts, describe the function, and any inputs/outputs. Technically the function would be described in the Read Me file, but a read me can often be skipped when browsing on the internet looking for a good code to fix your problem. In addition, it is very useful to know how the inputs/outputs are structured. Is it a list, array, string ... etc.

An example of this is from Gr 25:

```
def analyze_cross_sections(model_path):  
    """  
    Gennemgår IfcBeam-objekter og returnerer en oversigt over unikke tværsnitstørrelser,  
    antal bjælker pr. størrelse, og antal bjælker uden de nødvendige properties.  
  
    Parameters:  
    model_path (str): Stien til IFC-modellen.  
  
    Returns:  
    dict: En dictionary med tværsnitstørrelser som nøgler og antal bjælker, antal uden properties  
    """
```

Gr 22

Overall, a very interesting and useful tool. The only thing I can recommend is explain abit more on how to use speple in the Read Me file and add a general description to the tool.

Gr 23

A general description at the top would be nice. Potentially consider working more with a different structure for the output. Currently the output is printed in the console, but it could also be nice to provide a directory with the outputs, as it is easier to continue working with the data that way.

Gr 24

A general description at the top would be nice.

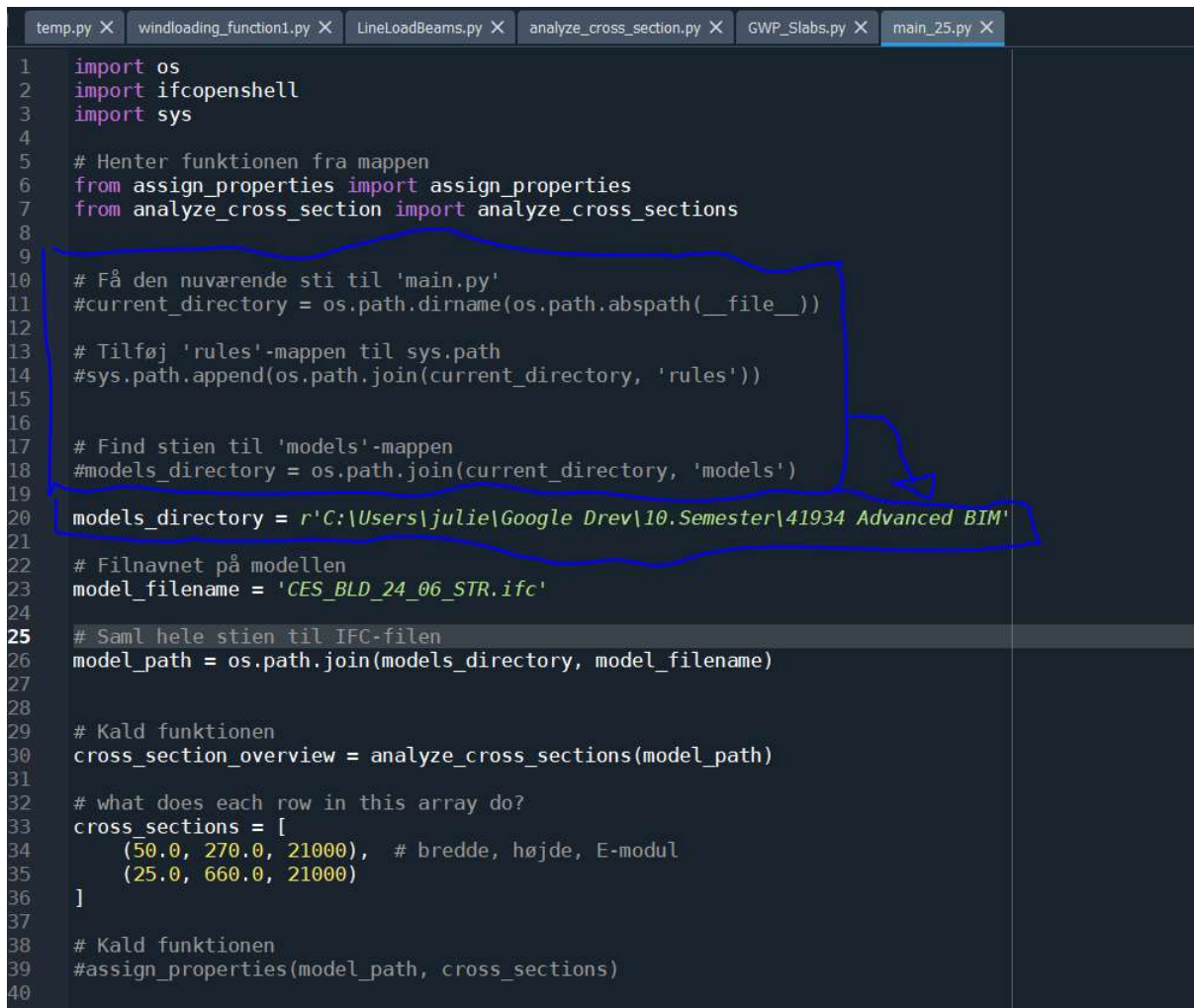
The user needs to input the file path their IFC model, witch should be placed at the top of the script.

Potentially consider working more with a different structure for the output. Currently the output is printed in the console, but it could also be nice to provide a directory with the outputs, as it is easier to continue working with the data that way.

Gr 25

In GitHub, you A3 main script should be in the A3 folder

Also, maybe simplify the way the model path is located, to make it more flexible and transparent for others



The image shows a screenshot of a code editor with several tabs at the top: temp.py, windloading_function1.py, LineLoadBeams.py, analyze_cross_section.py, GWP_Slabs.py, and main_25.py. The main_25.py tab is active, showing a Python script. The script is annotated with blue hand-drawn boxes and arrows. One box highlights lines 10-18, which define the current directory and append the 'rules' directory to the system path. Another box highlights line 19, which sets the models_directory to a specific path. A third box highlights line 20, which sets the model_filename to 'CES_BLD_24_06_STR.ifc'. A fourth box highlights line 25, which joins the model_path. The script content is as follows:

```
1 import os
2 import ifcopenshell
3 import sys
4
5 # Henter funktionen fra mappen
6 from assign_properties import assign_properties
7 from analyze_cross_section import analyze_cross_sections
8
9
10 # Få den nuværende sti til 'main.py'
11 #current_directory = os.path.dirname(os.path.abspath(__file__))
12
13 # Tilføj 'rules'-mappen til sys.path
14 #sys.path.append(os.path.join(current_directory, 'rules'))
15
16
17 # Find stien til 'models'-mappen
18 #models_directory = os.path.join(current_directory, 'models')
19 models_directory = r'C:\Users\julie\Google Drev\10.Semester\41934 Advanced BIM'
20
21
22 # Filnavnet på modellen
23 model_filename = 'CES_BLD_24_06_STR.ifc'
24
25 # Saml hele stien til IFC-filen
26 model_path = os.path.join(models_directory, model_filename)
27
28
29 # Kald funktionen
30 cross_section_overview = analyze_cross_sections(model_path)
31
32 # what does each row in this array do?
33 cross_sections = [
34     (50.0, 270.0, 21000), # bredde, højde, E-modul
35     (25.0, 660.0, 21000)
36 ]
37
38 # Kald funktionen
39 #assign_properties(model_path, cross_sections)
40
```

I was confused about this part of the code. The input to your `assign_properties` function should be more clearly explained

```
32 # what does each row in this array do?
33 # which sectionens get assigned to which beams in the IFC model?
34 # should the rows of the list correspond to amount cross sections in the model?
35 cross_sections = [
36     (50.0, 270.0, 21000), # bredde, højde, E-modul
37     (25.0, 660.0, 21000)
38 ]
```

Gr 26

In general a great idea, seems like a useful script. Consider adding a description at top of the script.

The choice to have a intermediate stage were the user interacts with a excel file is interesting. I can see both advantages and disadvantages

GR 27

Add bold text to your headers in the read me file

There was a problem with the script when I modified the input values. You should add a general description at the top and also specify if the input is an exact match, or if it is all column dimensions above the input value

```

# Final output message
if all_columns_match == True:
    print("All columns on the specified floor match the requirements.")
else:
    print("Some columns on the specified floor do not match the requirements.GlobalIDs of non-

import ifcopenshell
model = ifcopenshell.open("C:/Users/julie/Google Drev/10.Semester/41934 Advanced BIM/CES_BLD_24_06

# Example
specific_floor_level = 8 # Replace with the desired floor level elevation
req_depth = 20 # Replace with the required depth for the specific floor
req_width = 20 # Replace with the required width for the specific floor

checkRule(model, specific_floor_level, req_depth, req_width)
#The code checks that all the columns on a given floor match the dimensions given by the user.
#It should print out a message if all match the requirements or it should print out the GlobalID o
Traceback (most recent call last):
File ~\anaconda3\lib\site-packages\spyder_kernels\ipycompat.py:356 in
compatt_exec
exec(code, globals, locals)
File c:\users\julie\google drev\10.semester\41934 advanced
bim\try out as works.py:49
checkRule(model, specific_floor_level, req_depth, req_width)
File c:\users\julie\google drev\10.semester\41934 advanced
bim\try out as works.py:49 in checkRule
non_matching_columns.append(global_id) # Store GlobalID of the non-
matching column
NameError: name 'global_id' is not defined
In [5]:

```

Gr 28

A read me file is missing from GitHub

The input for IFC file location should be placed at the top of the script.

The script is cool and seems useful.

If possible, you should try to cut down the computation time by simplifying how many stories you calculate the wind load for. Most buildings have the same layout all the way up, so simplification can be made as the geometry is the same.

I was confused about why the script printed so many graphs.