Assignment 1, Part B: Men At Work[1]
(5%, due 11:59pm Sunday, April 22nd, end of Week 7)

Overview

This is the second part of a two-part assignment. This part is worth 5% of your final grade for IFB104. Part A which preceded it was worth 20%. This part is intended as a last-minute extension to the assignment, thereby testing the maintainability of your code from Part A and your ability to work under time presssure. If you have a neat, clear solution to Part A you will find completing Part B easy. For the whole assignment you will submit only one file, containing your combined solution to both Parts A and B, and you will receive one grade for the whole 25% assignment.

Motivation

One of the most common tasks in "Building IT Systems" is modifying some existing code. In practice, computer programs are written only once but are subsequently modified and extended often during their operational lifetime. Code changes may be required in response to internal factors, such as the need to correct design flaws or coding errors, or external factors, such as changes in consumer requirements. A common situation is when some new feature must be added to an existing program. This is the scenario simulated in this part of the assignment.

This task requires you to extend your solution to Part A of the assignment by adding an additional feature. It tests:

- Your ability to work under time pressure; and

- The quality and clarity of your code for Part A, because a well-written solution to Part A will make completing this part of the assignment much easier.

Goal

In Part A of this assignment you were required to develop a program which could draw cityscapes by following a set of building plans encoded as a randomly-generated Python list. However, if you look towards Brisbane's skyline from QUT's Gardens Point campus you will see not only completed buildings, but many that are still under construction. Therefore, in Part B of this assignment you will simulate this real-life scenario by modifying your program to draw buildings that are yet to be completed in your cityscapes. These buildings are indicated in the city plans by the presence of an 'X' as the fourth value in the description of a building.

In Part A of the assignment you were required to draw buildings consisting of one or more floors and a roof structure. To complete this additional task you must extend your solution to

---

[1] "Men" is used here in its inclusive form as a plural noun, i.e., "human beings", "people" or "persons".

Part A so that instead of the roof structure it draws an image showing that the building is still under construction for any building which has an 'X' at the end of the data set describing it.

You have a free choice of what to draw to indicate the building's incomplete state, as long as it is reasonably complex and realistic. Possibilities include:

- a building crane (as in our example below);

- a "Men at Work", "Danger", "Under Construction" or some other kind of sign;

- building equipment such as cement mixers;

- smoke or sparks generated by welding equipment;

- an incomplete framework of the roof structure currently under construction;

- stick figures of the construction workers; or

- something else suitable to visually indicate construction work in progress.

You will complete this part of the assignment by extending your code for Part A. No additional Python template file is supplied for this part. As per Part A, your program must work for any "city plan" that can be generated by the provided function `random_plan`. Your extended code must work correctly for *any possible* city plan.

Illustrative example

To illustrate the requirements we'll continue our example from the Part A instructions. Recall that we developed a solution which drew four styles of building inspired by superhero comics and science fiction movies. To indicate that such a building is still under construction, our Part B solution draws a building crane at the top of the building instead of its usual roof structure.

For instance, consider the following data set, generated by calling function `random_plan`.

```
[[1, 'B', 4, 'O'],
 [3, 'B', 1, 'O'],
 [4, 'D', 3, 'X'],
 [5, 'C', 4, 'O'],
 [6, 'A', 9, 'X'],
 [9, 'C', 6, 'X'],
 [10, 'B', 7, 'O']]
```

Normally these instructions would lead us to drawing seven complete buildings. However, the 'X' in the instructions for the buildings on Sites 4, 6 and 9 indicates that these buildings are still under construction. (The buildings in question are the *Back to the Future* clock tower, the Gotham City Police Headquarters and the Ghostbusters Firehouse, in our illustrative example.) Therefore, instead of drawing their usual roof structures we draw an image of a building crane at the top of each of these three buildings as shown overleaf.
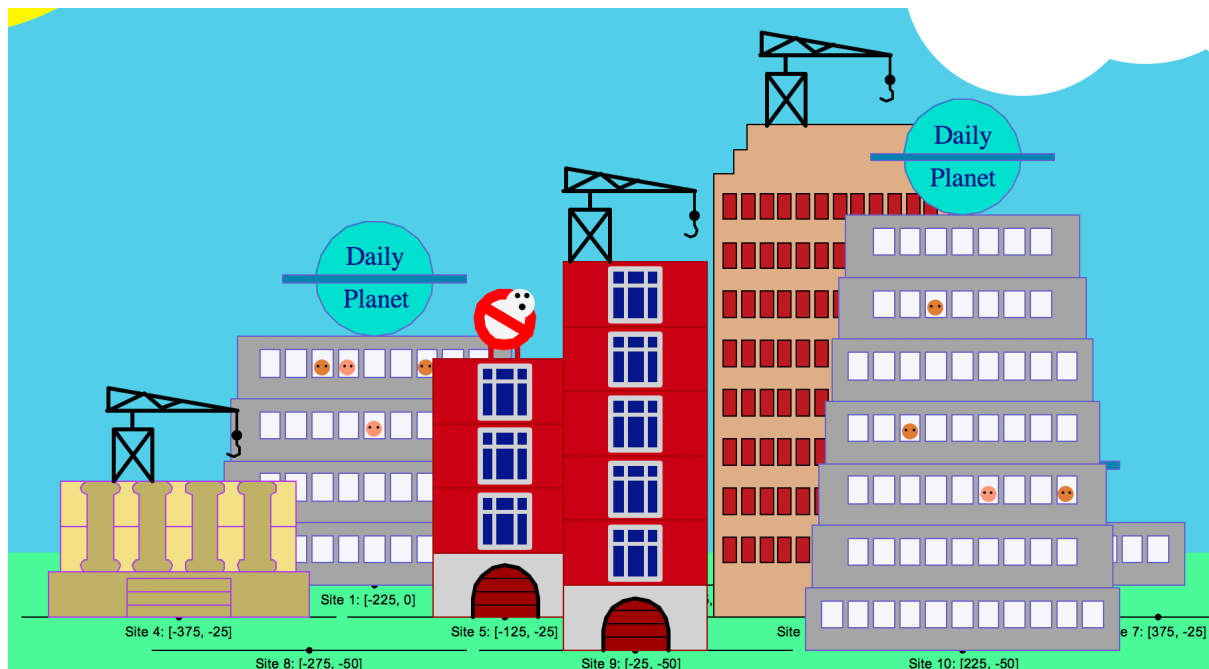
On Site 4 the clock has been replaced with the crane; on Site 6 the printed "Police HQ" sign and the Batsignal have also been replaced; and on Site 9 the Ghostbusters logo is missing and the crane appears in its place. In each case all of the usual floors of the building are still drawn; only the roof structure is changed.

As another example, consider the following randomly-generated city plan.

```
[[2, 'D', 10, 'O'],
 [5, 'B', 6, 'O'],
 [6, 'B', 5, 'X'],
 [7, 'C', 3, 'O'],
 [8, 'D', 1, 'X'],
 [9, 'B', 7, 'X'],
 [10, 'A', 7, 'O']]
```

In this case two copies of the Daily Planet building, at Sites 6 and 9, have yet to have their ringed-planet logo installed, and a single-story version of the Clock Tower, at Site 8, is also still under construction, as shown overleaf. All of the other buildings are complete.
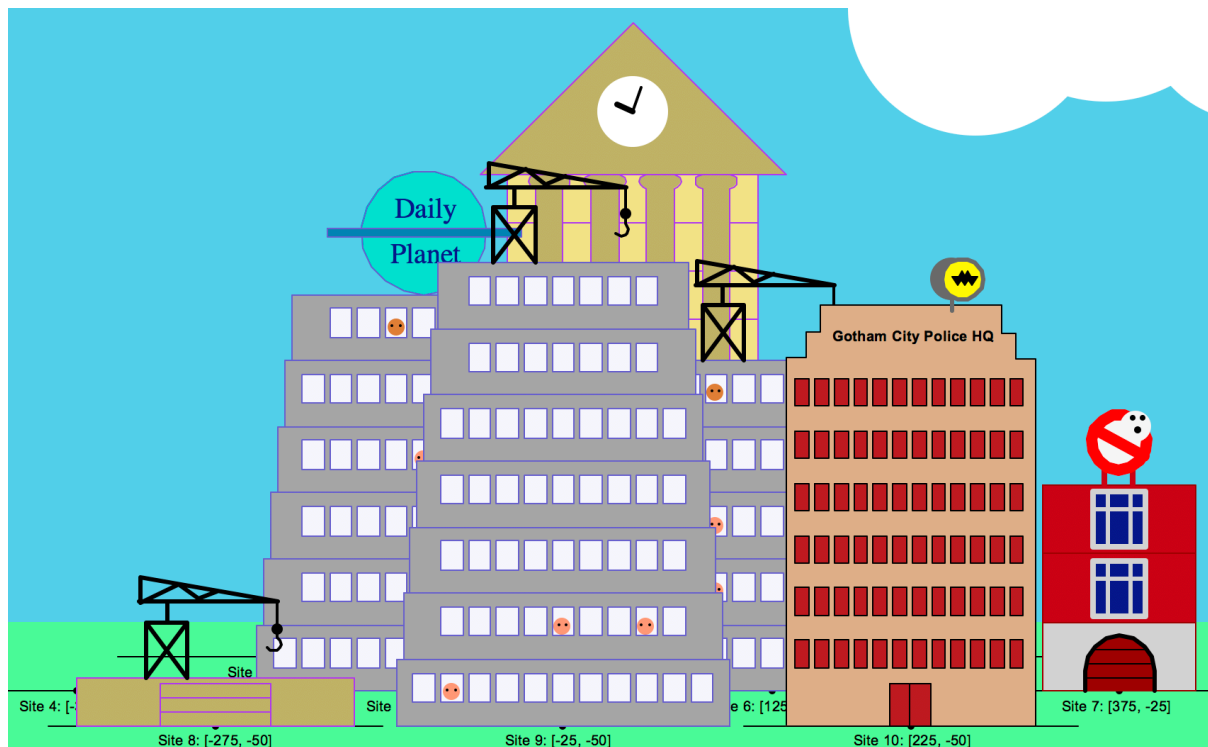
Requirements and marking guide

To complete this task you are required to extend your Part A `cityscapes.py` file by modifying the code so that building instrcutions that end with an 'X' result in the corresponding building's roof structure being replaced by some image indicating construction work in progress.

Your submitted solution for both Parts A and B will consist of a *single Python file*. Your Part B extension must satisfy the following criteria. Marks available are as shown.

1. **An image representing construction work can be drawn (2%)**. Your program must be capable of drawing a non-trivial image which is clearly evocative of building work in progress. The image must be drawn using basic Turtle graphics moves and must be clearly distinct from the rest of the building. Some suggestions for such an image are listed above. You are free to draw a different "construction image" for each style of building, but this is not necessary.

2. **Buildings marked with an 'X' are shown as under construction (3%)**. Whenever your `build_city` function is given instructions for a building ending with an 'X' it must draw your chosen "construction image" instead of the building's usual roof structure. (All other floors of the building are unaffected.) The image must be drawn at an appropriate position and height to clearly be associated with the building in question. All other buildings should be drawn in their usual "completed" form.

You must complete the assignment using basic Turtle graphics and maths functions only. You may not import any additional modules or files into your program other than those already

included in the given `cityscapes.py` template. In particular, you may not import any image files for use in creating your drawing.

Development hints

- It should be possible to complete this task merely by *adding* code to your existing solution, with little or no change to the code you have already completed.

- If you are unable to complete the whole task, just submit whatever part you can get working. You will receive *partial marks for incomplete solutions*. Try to ensure that your program runs when submitted, even if it is incomplete.

Deliverable

You must develop your solution by completing and submitting the provided Python 3 file `cityscapes.py` as follows. **Do not submit any other files! Do not submit a compressed archive ('zip' or 'rar') containing multiple files!**

1. Complete the "statement" at the beginning of the Python file to confirm that this is your own individual work by inserting your name and student number in the places indicated. *We will assume that submissions without a completed statement are <u>not</u> your own work.*

2. Complete your solution by developing Python code to replace the dummy `build_city` function. You should complete your solution using only the standard Python 3 modules already imported by the provided template. In particular, you must *not* use any Python modules that must be downloaded and installed separately because the markers may not have these modules installed. Furthermore, you may *not* import any image files into your solution; the entire image must be drawn using Turtle graphics drawing primitives.

3. Submit *a single Python file* containing your solution for marking. Do *not* submit multiple files. Only a single file will be accepted, so you cannot accompany your solution with other files or pre-defined images. **Do not submit any other files! Submit only a single Python 3 file!**

Apart from working correctly your program code must be well-presented and easy to understand, thanks to (sparse) commenting that explains the *purpose* of significant code segments and *helpful* choices of variable and function names. *Professional presentation* of your code will be taken into account when marking this assignment.

If you are unable to solve the whole problem, submit whatever parts you can get working. You will receive *partial marks for incomplete solutions*.

Plagiarism

This is an individual assessment item. All files submitted will be subjected to software plagiarism analysis using the MoSS system (http://theory.stanford.edu/~aiken/moss/). Serious violations of the university's policies regarding plagiarism will be forwarded to the Science and Engineering Faculty's Academic Misconduct Committee for formal prosecution.

How to submit your solution

A link is available on Blackboard under *Assessment* for uploading your solution file before the deadline (11:59pm Sunday, April 22nd, end of Week 7). You can submit as many drafts of your solution as you like. You are strongly encouraged to *submit draft solutions* before the deadline as insurance against computer and network failures. If you are unsure whether or not you have successfully uploaded your file, upload it again!

Students who encounter problems uploading their Python files to Blackboard should contact the *IT Helpdesk* (ithelpdesk@qut.edu.au; 3138 4000) for assistance and advice. Teaching staff will *not* answer email queries on the weekend the assignment is due, so ensure that you have successfully uploaded at least one solution by close-of-business on Friday, April 20th.