

***Modelling the Solar System (Assessment Task 7.3)***  
***in Fundamentals of Computing***  
**MSc Computer Science (online)**  
**Sheffield-Hallam University**

Gary Crosby  
SHU# c5058354  
16 October 2025

## Assignment Specification

From the module's *Assignment Brief*:

*"Create a program that displays information about planets in our solar system. For each planet your program should hold its:*

- *Name.*
- *Mass.*
- *Distance from the Sun.*
- *A list of the planet's moons (be sensible, don't include all of the moons for the large planets).*
- *All data should be held using appropriate data types. Use Wikipedia to help with values.*

*A user should be able to query your data by asking questions such as:*

- *Tell me everything about Saturn?*
- *How massive is Neptune?*
- *Is Pluto in the list of planets?*
- *How many moons does Earth have?*

*You must write a test plan that covers key parts of your solution. You should include this as a separate file within your solution.*

*You should bring all of your programming knowledge to bear on this task.*

- *You must use classes throughout your program.*
- *You should include code to properly validate inputs from the user.*
- *The raw data may be held in a file.*
- *You may build a simple menu system.*
- *You might create a GUI using Tkinter.*
- *You may want to use a unit testing framework to implement some, or all, of your test plan."*

## GitHub Repository

Completed files for this project are located in a public GitHub repository:

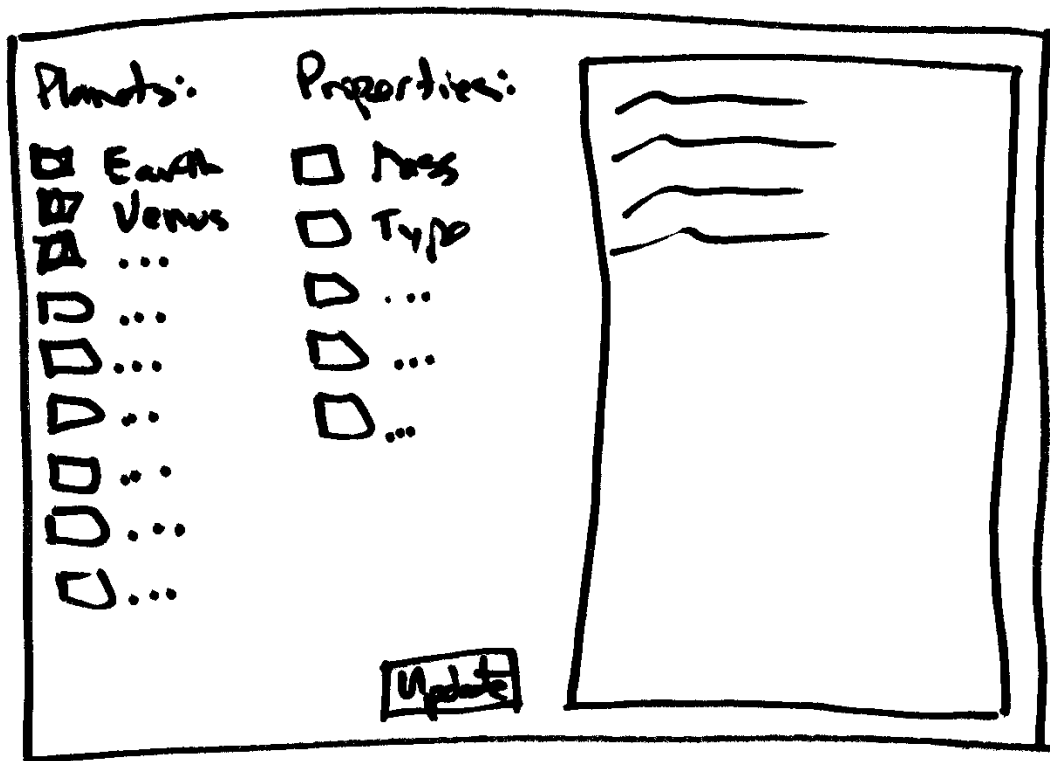
<https://github.com/gary-crosby/modelling-solar-system>

## Concept

Program reads planet data from a JSON file. No planet data is hardcoded in Python code.

GUI has one column of checkboxes for **Planets**, a second column of planet **Properties**, and **Update** button, and a large text area for output. (Refer to sketch below.)

User selects one or more planet(s) and one or more property(s), clicks the **Update** button, and the program displays the relevant information in the text area.



## User Input Validation

Because user input is limited to selecting checkboxes, user input validation is simple and straightforward:

- **Update** button is disabled until user selects at least one planet and at least one characteristic.
- After **Update** is selected and information displayed, the **Update** button will be disabled until user changes their selections.

## Testing Plan

The following testing was completed:

- Unit testing for class instantiation and call of class method (see file ***unit\_tests.py***).
- Black box testing was conducted to ensure that only valid planet and property selections results in an enabled **Update** button.

## AI Transparency Declaration Statement

For this assessment, I used AI in two contexts:

1. I used Google Gemini Pro 2.5 to extract data from 4 websites and create a **solar\_system\_data.json** containing the data. I manually audited the JSON for 3 planets so obtain reasonable assurance that the data are correct. In Visual Studio Code I made changes to the JSON file: renamed JSON keys, deleted some keys and values of data I decided not to use, and added keys and values for data sources. Because the focus on this assessment is programming rather than structuring data in files and because I felt compelled (by my OCD) to include the names of all permanently-named moons in the project, this approach seemed an acceptable use of AI.
2. Visual Studio Code (VSC) AI-based **code completion**. When enabled, VSC's code completion suggests short code snippets based on the code already in the file and the line of code the developer is currently typing. In VSC Code, this feature is enabled by default. Some snippets are relevant whereas others are not. However, in any case, the developer must review every snippet to determine whether it is relevant and whether to accept (or reject) it. Additionally, in my experience few of the snippets can be used as-is. Rather, most need to be further refined to make them suitable to task, and to conform to the developer's style throughout the rest of the program. (Note that VSC has a related feature, **next edit suggestions**, which takes code completions to a higher level. For this assessment, I disabled next edit suggestions.)

Except for AI uses described about, I did not use any other AI agents or queries to complete this assessment.