**Joseph Cryer**

**Planetary Simulation**

**Project**

**Candidate Number: 2115**

**Centre Number: 15121**

Table of Contents

[**Analysis**](#_6i7y8ye7vih8) **2**

[Description](#_h6gdxb2sgvfa) 2

[Users and Clients](#_ywuei3ldm3u8) 3

[Use Cases:](#_m2i9ftsrlb0j) 3

[Project Background:](#_rov6isyvns1e) 14

[Objectives](#_ssha8sj5nmum) 15

# Analysis

## Description

The program requested is a 3D simulation environment capable of displaying various planetary phenomena clearly. It will visualise a variety of different things (e.g. our solar system, a binary star system, comet orbits etc.).

Currently, a 2D online flash web application is used to show this, however it has many flaws, such as not being able to add custom objects, or modify many of the experimental values of the objects. Teachers attempt to supplement this tool with animated videos on the topic, however this is also very rigid, as any question by a student cannot immediately be followed up with a visual tool that helps to explain the answer.

## Research

## Users and Clients

### Use Cases:

Use Case 1:

**User:** Teacher

**Activity:** Open and view a preset simulation

**What happens:**

* Load up the application
* Click on the “Run Simulation” option on the menu
* Click on the “Load From Presets” option
* Select a preset from the list of simulations
* Click “Confirm”
* Click “Run” at the top of the control panel
* View the preset simulation, rendered in 3D

**New parts to the system:**

* Main menu
* Simulation menu
* Preset menu
* Control menu
* Object definitions set up to store information about the simulation
* Way to load simulations from local storage as presets
* Methods to calculate the relative positions and velocities of all the objects in the simulation over a period of time
* Way to render objects in these positions and velocities in 3D, and have them move over a period of time

**How this will be tested:**

* The application is started
  + Expected result: The main menu appears
* The “Run Simulation” option on the menu is clicked
  + Expected result: The sim menu appears
* The “Load From Presets” option on the menu is clicked
  + Expected result: The preset menu appears
  + A list of selectable presets saved within the database should load in
* A preset is selected
  + Expected result: The preset selected is now highlighted, and the “Confirm” button may be clicked.
* The “Confirm” button is clicked
  + Expected result: The control menu appears, with information about the simulation.
* The “Run” button is clicked
  + Expected result: A window opens with the simulation inside, rendering moving 3D spheres representing objects in the simulation.

Use Case 2:

**User:** Teacher

**Activity:** Create a new custom simulation of the Sun and the Earth

**What happens:**

* Load up the application
* Click on the “Run Simulation” option on the menu
* Click on the “New Custom Simulation” option
* Click on the “Add” option on the control menu
* Click on the “Copy from Existing” option, and select the Sun
* Click the “Confirm” button, and then the “Save” button
* Click on the “Add” option on the control menu
* Click on the “Copy from Existing” option, and select the Earth
* Click the “Confirm” button, and then the “Save” button
* Click “Run” at the top of the control panel
* View the custom simulation, rendered in 3D

**New parts to the system:**

* Object edit menu
* List of standard solar objects menu
* Way to add new objects to a simulation
* Way to modify the values of a solar object
* Way to get a list of the Sun and the planets in the solar system, and accurate values for them (Mass, initial velocity, initial position, radius)

**How this will be tested:**

* The application is started
  + Expected result: The main menu appears
* The “Run Simulation” option on the menu is clicked
  + Expected result: The sim menu appears
* The “New Custom Simulation” option on the menu is clicked
  + Expected result: The control menu appears
* The “Add” option on the menu is clicked
  + Expected result: A form appears with all of the editable values of a planetary body.
* The “Copy from Existing” option is clicked
  + Expected result: A selectable list of the Sun and the planets appears
* The Sun is selected, and “Confirm” is pressed
  + Expected result: The selectable list closes. The form has its values filled in with accurate values for the Sun
* The “Save” option is clicked
  + The form closes, and the Sun is added to a list of objects in the simulation on the control form.
* The “Add” option on the menu is clicked
  + Expected result: A form appears with all of the editable values of a planetary body.
* The “Copy from Existing” option is clicked
  + Expected result: A selectable list of the Sun and the planets appears
* The Earth is selected, and “Confirm” is pressed
  + Expected result: The selectable list closes. The form has its values filled in with accurate values for the Earth
* The “Save” option is clicked
  + The form closes, and the Earth is added to a list of objects in the simulation on the control form.
* The “Run” button is clicked
  + Expected result: A window opens with the simulation inside, rendering moving 3D spheres representing the Sun and the Earth in the simulation.

Use Case 3:

**User:** Teacher

**Activity:** Save a new custom simulation of the Sun and the Earth as a new preset

**What happens:**

* Load up the application
* Click on the “Run Simulation” option on the menu
* Click on the “New Custom Simulation” option
* Click on the “Add” option on the control menu
* Click on the “Copy from Existing” option, and select the Sun
* Click the “Confirm” button, and then the “Save” button
* Click on the “Add” option on the control menu
* Click on the “Copy from Existing” option, and select the Earth
* Click the “Confirm” button, and then the “Save” button
* Click the “Save as Preset” option on the control menu
* Give the new preset simulation a name and a description
* Click the “Confirm” button

**New parts to the system:**

* Way to save simulations to local storage as new presets
* New preset menu

**How this will be tested:**

* The application is started
  + Expected result: The main menu appears
* The “Run Simulation” option on the menu is clicked
  + Expected result: The sim menu appears
* The “New Custom Simulation” option on the menu is clicked
  + Expected result: The control menu appears
* The “Add” option on the menu is clicked
  + Expected result: A form appears with all of the editable values of a planetary body.
* The “Copy from Existing” option is clicked
  + Expected result: A selectable list of the Sun and the planets appears
* The Sun is selected, and “Confirm” is pressed
  + Expected result: The selectable list closes. The form has its values filled in with accurate values for the Sun
* The “Save” option is clicked
  + The form closes, and the Sun is added to a list of objects in the simulation on the control form.
* The “Add” option on the menu is clicked
  + Expected result: A form appears with all of the editable values of a planetary body.
* The “Copy from Existing” option is clicked
  + Expected result: A selectable list of the Sun and the planets appears
* The Earth is selected, and “Confirm” is pressed
  + Expected result: The selectable list closes. The form has its values filled in with accurate values for the Earth
* The “Save” option is clicked
  + The form closes, and the Earth is added to a list of objects in the simulation on the control form.
* The “Run” button is clicked
  + Expected result: A window opens with the simulation inside, rendering moving 3D spheres representing the Sun and the Earth in the simulation.

Use Case 4:

**User:** Teacher

**Activity:** Modify values for an object inside a custom simulation of the Sun and the Earth, as the simulation is running

**What happens:**

* Load up the application
* Click on the “Run Simulation” option on the menu
* Click on the “New Custom Simulation” option
* Click on the “Add” option on the control menu
* Click on the “Copy from Existing” option, and select the Sun
* Click the “Confirm” button, and then the “Save” button
* Click on the “Add” option on the control menu
* Click on the “Copy from Existing” option, and select the Earth
* Click the “Confirm” button, and then the “Save” button
* Click the “Confirm” button
* Click “Run” at the top of the control panel
* View the custom simulation, rendered in 3D
* Click on the object labelled “Sun” on the control form, and then click the “Edit” button
* Change the colour of the Sun to be yellow, and then increase the mass of the Sun by 10x
* Click the “Save” button
* See the Sun change colour, and the Earth collide with the Sun

**New parts to the system:**

* Way to recalculate the positions and velocities of the objects in the simulation after changing values

**How this will be tested:**

* The application is started
  + Expected result: The main menu appears
* The “Run Simulation” option on the menu is clicked
  + Expected result: The sim menu appears
* The “New Custom Simulation” option on the menu is clicked
  + Expected result: The control menu appears
* The “Add” option on the menu is clicked
  + Expected result: A form appears with all of the editable values of a planetary body.
* The “Copy from Existing” option is clicked
  + Expected result: A selectable list of the Sun and the planets appears
* The Sun is selected, and “Confirm” is pressed
  + Expected result: The selectable list closes. The form has its values filled in with accurate values for the Sun
* The “Save” option is clicked
  + The form closes, and the Sun is added to a list of objects in the simulation on the control form.
* The “Add” option on the menu is clicked
  + Expected result: A form appears with all of the editable values of a planetary body.
* The “Copy from Existing” option is clicked
  + Expected result: A selectable list of the Sun and the planets appears
* The Earth is selected, and “Confirm” is pressed
  + Expected result: The selectable list closes. The form has its values filled in with accurate values for the Earth
* The “Save” option is clicked
  + The form closes, and the Earth is added to a list of objects in the simulation on the control form.
* The “Run” button is clicked
  + Expected result: A window opens with the simulation inside, rendering moving 3D spheres representing the Sun and the Earth in the simulation.
* The object labelled “Sun” on the control form is clicked, and the “Edit” button is then clicked
  + Expected result: A form appears with all of the editable values for the Sun
* The “Object Colour” button is pressed
  + Expected result: A form appears to select a colour for the object
* A yellow colour is selected, and the “OK” button is pressed
  + Expected result: The form closes, and the larger object form’s colour indicator changes to the new colour
* The mass of the Sun is increased by ten times, and the “Save” button is clicked
  + Expected result: The simulation will continue from where it was previously, with the new values set. The Sun should change colour, and the Earth should collide with it.

Use Case 5:

**User:** Teacher

**Activity:** Delete an object inside a custom simulation of the Sun and the Earth, as the simulation is running

**What happens:**

* Load up the application
* Click on the “Run Simulation” option on the menu
* Click on the “New Custom Simulation” option
* Click on the “Add” option on the control menu
* Click on the “Copy from Existing” option, and select the Sun
* Click the “Confirm” button, and then the “Save” button
* Click on the “Add” option on the control menu
* Click on the “Copy from Existing” option, and select the Earth
* Click the “Confirm” button, and then the “Save” button
* Click the “Confirm” button
* Click “Run” at the top of the control panel
* View the custom simulation, rendered in 3D
* Click on the object labelled “Sun” on the control form, and then click the “Delete” button
* See the Sun disappear, and the Earth travel in a straight line

**New parts to the system:**

* Way to recalculate the positions and velocities of the objects in the simulation after removing an object

**How this will be tested:**

* The application is started
  + Expected result: The main menu appears
* The “Run Simulation” option on the menu is clicked
  + Expected result: The sim menu appears
* The “New Custom Simulation” option on the menu is clicked
  + Expected result: The control menu appears
* The “Add” option on the menu is clicked
  + Expected result: A form appears with all of the editable values of a planetary body.
* The “Copy from Existing” option is clicked
  + Expected result: A selectable list of the Sun and the planets appears
* The Sun is selected, and “Confirm” is pressed
  + Expected result: The selectable list closes. The form has its values filled in with accurate values for the Sun
* The “Save” option is clicked
  + The form closes, and the Sun is added to a list of objects in the simulation on the control form.
* The “Add” option on the menu is clicked
  + Expected result: A form appears with all of the editable values of a planetary body.
* The “Copy from Existing” option is clicked
  + Expected result: A selectable list of the Sun and the planets appears
* The Earth is selected, and “Confirm” is pressed
  + Expected result: The selectable list closes. The form has its values filled in with accurate values for the Earth
* The “Save” option is clicked
  + The form closes, and the Earth is added to a list of objects in the simulation on the control form.
* The “Run” button is clicked
  + Expected result: A window opens with the simulation inside, rendering moving 3D spheres representing the Sun and the Earth in the simulation.
* The object labelled “Sun” on the control form is clicked, and the “Delete” button is then clicked
  + Expected result: The “Sun” object disappears from the control form list, and the simulation updates itself to remove the Sun. The Earth travels onwards in a straight line.

Use Case 6:

**User:** Teacher

**Activity:** Save a new custom simulation of the Sun and the Earth to a file

**What happens:**

* Load up the application
* Click on the “Run Simulation” option on the menu
* Click on the “New Custom Simulation” option
* Click on the “Add” option on the control menu
* Click on the “Copy from Existing” option, and select the Sun
* Click the “Confirm” button, and then the “Save” button
* Click on the “Add” option on the control menu
* Click on the “Copy from Existing” option, and select the Earth
* Click the “Confirm” button, and then the “Save” button
* Click the “Export” button on the control menu
* Select a location to save the exported simulation, and a file name
* Click “Save”

**New parts to the system:**

* Way to save simulations to a file

**How this will be tested:**

* The application is started
  + Expected result: The main menu appears
* The “Run Simulation” option on the menu is clicked
  + Expected result: The sim menu appears
* The “New Custom Simulation” option on the menu is clicked
  + Expected result: The control menu appears
* The “Add” option on the menu is clicked
  + Expected result: A form appears with all of the editable values of a planetary body.
* The “Copy from Existing” option is clicked
  + Expected result: A selectable list of the Sun and the planets appears
* The Sun is selected, and “Confirm” is pressed
  + Expected result: The selectable list closes. The form has its values filled in with accurate values for the Sun
* The “Save” option is clicked
  + The form closes, and the Sun is added to a list of objects in the simulation on the control form.
* The “Add” option on the menu is clicked
  + Expected result: A form appears with all of the editable values of a planetary body.
* The “Copy from Existing” option is clicked
  + Expected result: A selectable list of the Sun and the planets appears
* The Earth is selected, and “Confirm” is pressed
  + Expected result: The selectable list closes. The form has its values filled in with accurate values for the Earth
* The “Save” option is clicked
  + The form closes, and the Earth is added to a list of objects in the simulation on the control form.
* The “Export” option is clicked
  + Expected result: A dialog appears to allow the user to select a file name for the exported file, and the location to save the file
* The “Save” button is clicked
  + Expected result: The simulation is exported to a file in that location with that name.

Use Case 7:

**User:** Teacher

**Activity:** Load a custom simulation of the Sun and the Earth from a file

**What happens:**

* Load up the application
* Click on the “Run Simulation” option on the menu
* Click on the “Load from File” option
* Select a file with a file dialog (the one made in Use Case 6), and then press the “OK” button
* Click the “Run” button on the control form

**New parts to the system:**

* Way to load simulations from a file

**How this will be tested:**

* The application is started
  + Expected result: The main menu appears
* The “Run Simulation” option on the menu is clicked
  + Expected result: The sim menu appears
* The “Load from File” option on the menu is clicked
  + Expected result: A file selection dialog appears
* Select a previously exported simulation file (from Use Case 6), and then click the “OK” button
  + Expected result: The control menu appears with all of the simulation information contained within it
* The “Run” button is clicked
  + Expected result: A window opens with the simulation inside, rendering moving 3D spheres representing the Sun and the Earth in the simulation.

## Project Background:

Currently, the Bedford Modern School Physics department does not have a suitable visual aid to help in teaching the astrophysics sections of their A-Level or GCSE courses, or the years seven to nine introduction lessons.

A-Level:

For the OCR Physics course chosen by the school, there are three chapters dedicated to astrophysics and cosmology. These are named “Gravitational Fields”, “Stars” and “Cosmology”. There are a few key things that the Physics department want to be able to do with a visual aid, these being:

* To show how every object in a system affects every other object in the system,
* To be able to modify objects in a simulation as it is running
  + In order to show how small changes affect the simulation massively
* To be able to introduce a new object (for example, a comet, or another planet) into the simulation and see how much it affects the other objects in the simulation
* To be able to have preset simulations showing:
  + Comets
  + Black holes
  + Binary star systems
  + Our solar system

GCSE:

For the Edexcel Physics course chosen by the school, the astrophysics sections are minor. The emphasis in the course is therefore instead to give students a basic understanding of the solar system and the planetary bodies within it, without needing to go into the same level of detail as the A-Level simulations.

Years 7 - 9:

Here, the main request was that the sheer scale of the solar system can be displayed in a way that lets younger students comprehend it well.

## Objectives

1. The system must use the gravitational field equations taught in the A-Level Physics course, in order to approximate the movement of celestial objects within the solar system
2. It must be suitable to be displayed on a projector in a classroom
3. The system must be easily navigable and intuitive to use
4. The system must be able to run on the Windows 10 laptops used by the Science Department staff
5. There must be a way to distinguish between the different objects (e.g. using colours)
6. There must be some way to show the path that the object has taken through space
7. The system must show an animated three-dimensional representation of the simulation being ran
8. The speed of the simulation should be variable
9. There must be a way of easily navigating around the three-dimensional simulation environment
10. There must be a way to view the simulation from the perspective of any object in the simulation (centering around that object)
11. The system should come with a preset showing the Earth and Moon
12. The system should come with a preset showing the Earth and Sun
13. The system should come with a preset showing the Earth, Moon and Sun
14. The system should come with a preset showing the entire Solar System
15. The system should come with a preset showing a Binary Star system
16. There must be a way of quickly and efficiently loading up preset simulations
17. There must be a way of easily setting up and saving new permanent presets
18. There must be a way of creating a new custom simulation
19. There must be a way of adding, removing and editing celestial objects to the simulation as requested
20. There must be a way to modify the Initial Position for each celestial object in a simulation
21. There must be a way to modify the Initial Velocity for each celestial object in a simulation
22. There must be a way to modify the Mass for each celestial object in a simulation
23. There must be a way to modify the Radius for each celestial object in a simulation
24. There must be a way of saving a custom simulation to a portable file
25. There must be a way of loading a custom simulation from a portable file
26. The system must allow users to save custom simulations as preset simulations

# Design

## Packages used

- OpenTK (OpenGL)

- SQLite Lib

- Telnet Lib

- Metro UI Framework

- Newtonsoft.Json Lib

## Class Inheritance Diagram

## User Interface

## Database Data Dictionary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table** | **PK/FK** | **Column** | **Data Type** | **References** | **Description** |
| Object | PK | ObjectID | Integer |  | Unique number ID of object |
| Object |  | Name | Text |  | Name of object |
| Object |  | Mass | Real |  | Mass of object |
| Object |  | Radius | Real |  | Radius of object |
| Object |  | Obliquity | Real |  | Obliquity of object |
| Object |  | OrbitalSpeed | Real |  | Orbital speed of object |
| InitialValues | FK | ObjectID | Integer | Object | References unique object |
| InitialValues | FK | PlanetarySystemID | Integer | PlanetarySystem | References unique planetary system |
| InitialValues |  | PositionX | Real |  | Initial X co-ordinate of object position in a specific planetary system |
| InitialValues |  | PositionY | Real |  | Initial Y co-ordinate of object position in a specific planetary system |
| InitialValues |  | PositionZ | Real |  | Initial Z co-ordinate of object position in a specific planetary system |
| InitialValues |  | VelocityX | Real |  | Initial X vector of object velocity in a specific planetary system |
| InitialValues |  | VelocityY | Real |  | Initial Y vector of object velocity in a specific planetary system |
| InitialValues |  | VelocityZ | Real |  | Initial Z vector of object velocity in a specific planetary system |
| ObjectView | FK | ObjectID | Integer | Object | References unique object |
| ObjectView | FK | SimulationID | Integer | Simulation | References unique simulation |
| ObjectView |  | TrailActive | Integer |  | Boolean value stored as integer (1: true, 0: false)  States whether the trail is active for a specific object in a specific simulation |
| ObjectView |  | TrailLength | Integer |  | Length of trail for a specific object in a specific simulation |
| ObjectView |  | TrailColour | Text |  | Hex string of colour of a specific object’s trail in a specific simulation |
| ObjectView |  | ObjectColour | Text |  | Hex string of colour of a specific object in a specific simulation |
| PlanetarySystem | PK | PlanetarySystemID | Integer |  | Unique number ID of planetary system |
| PlanetarySystem |  | Name | Text |  | Name of planetary system |
| PlanetarySystem |  | Description | Text |  | Description for planetary system |
| Simulation | PK | SimulationID | Integer |  | Unique number ID of simulation |
| Simulation | FK | PlanetarySystemID | Integer | PlanetarySystem | References unique planetary system |
| Simulation |  | Zoom | Real |  | Zoom value for a specific simulation |
| Simulation |  | ZoomModifier | Real |  | Zoom modifier for a specific simulation |
| Simulation |  | Focus | Integer |  | Focus value for a specific simulation – references object ID |
| Simulation |  | Fixed | Integer |  | Boolean value stored as integer (1: true, 0: false)  States whether the camera position is fixed to a specific set of co-ordinates |
| Simulation |  | Speed | Integer |  | Speed value for a specific simulation |
| Simulation |  | SpeedModifier | Integer |  | Speed modifier for a specific simulation |
| Simulation |  | Scale | Integer |  | Scale value for a specific simulation |

## Database Normalisation

Requirements for First Normal Form:

* The values in each column of the table are atomic

Requirements for Second Normal Form:

* Must be in First Normal Form
* Every non-key attribute must depend upon the whole key

Requirements for Third Normal Form:

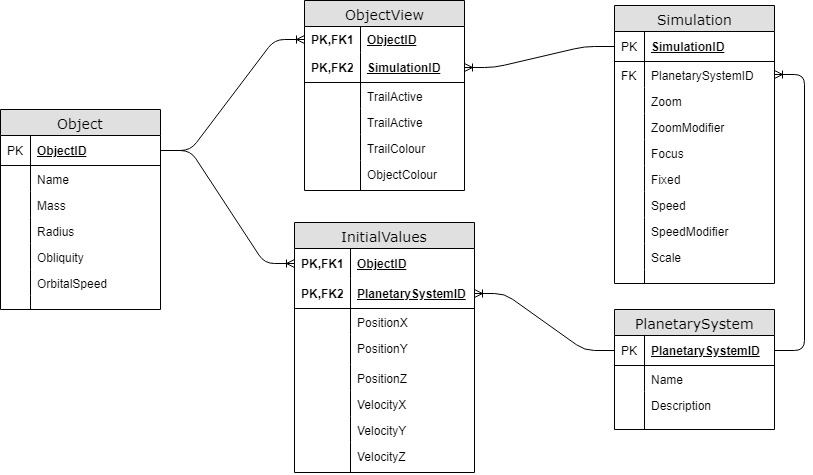
* Must be in Second Normal Form
* Must contain only columns that are non-transitively dependent on the primary key

Every value in my database is atomic (i.e. cannot be broken down into smaller parts). Therefore, it satisfies the requirement for it to be in First Normal Form.

There are three primary keys throughout my database – ObjectID for the Object table, PlanetarySystemID for the PlanetarySystem table and SimulationID for the Simulation table – and for each of these tables, every non-key attribute depends entirely on the key. In the ObjectView linking table a compound key is used, combining the foreign keys ObjectID and SimulationID. Likewise, in the InitialValues linking table a compound key is used, combining the foreign keys ObjectID and PlanetarySystemID. For these tables, every non-key attribute also depends entirely on the compound key.

No tables within my database contain columns that are transitively dependent on each other, and all tables within my database are comprised entirely of columns that depend on the primary key or compound key (as stated in the previous paragraph). Therefore, my database is in third normal form, as all of the above requirements have been satisfied.

## Database E-R Diagram



TrailLength

## Database Queries

## Algorithms

# Testing

## White Box Testing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test No.** | **Relevant Objective (s)** | **View** | **Type** | **Purpose** | **Action** | **Expected Result** | **Outcome** |
| 1 | 2, 3 | MainForm  SimForm | Normal | Test navigation | Click “Run Simulation” button | SimForm is opened, current form is closed. | Pass |
| 2 | 3 | MainForm | Normal | Test navigation | Click “Quit” button | Application is closed. | Pass |
| 3 | 3, 11, 12, 13, 14, 15, 16 | SimForm  PresetForm | Normal | Test navigation | Click “Load From Presets” button | PresetForm is opened, current form is closed. | Pass |
| 4 | 3, 26 | SimForm  ControlForm | Normal | Test navigation  Test file parsing | Click “Load From File” button.  Select valid simulation file.  Click “OK” button | File selection dialog is opened.  File is selected.  File is read, and ControlForm is opened with the simulation loaded from the file.  Current form is closed. | Pass |
| 5 | 3, 18 | SimForm  ControlForm | Normal | Test navigation | Click “New Custom Simulation” button | ControlForm is opened with a new blank simulation loaded, current form is closed. | Pass |
| 6 | 3 | SimForm  MainForm | Normal | Test navigation | Click “Back” button | MainForm is opened, current form is closed. | Pass |
| 7 | 3, 14, 16 | PresetForm  ControlForm | Normal | Test navigation  Test database methods | Select the “Solar System” preset, and then click “Confirm” button | ControlForm is opened with the “Solar System” simulation loaded, current form is closed. | Pass |
| 8 | 3 | PresetForm | Normal | Test navigation | Click “Back” button | SimForm is opened, current form is closed. | Pass |
| 9 | 3, 19 | ControlForm | Normal | Test navigation | Click “Add” button | ObjectForm is opened, fields are blank. | Pass |
| 10 | 3, 19 | ObjectForm | Erroneous | Test input fields & validation | Enter nothing for the name of the object and click Save. | Error message displays:  ’Name’ field must not be empty | Pass |
| 11 | 3, 19 | ObjectForm | Boundary | Test input fields & validation | Enter the name “Name that won't fit...........” for the object | Error message displays:  ‘Name’ field must be less than 30 chars. | Pass |
| 12 | 3, 19 | ObjectForm | Boundary | Test input fields & validation | Enter the name “Name that will fit...........” for the object | Value accepted. | Pass |
| 13 | 3, 19 | ObjectForm | Normal | Test input fields & validation | Enter the name “Moon” for the object and click Save. | Value accepted.  Other error messages for other fields appear. | Pass |
| 14 | 3, 19, 20 | ObjectForm | Erroneous | Test input fields & validation | Enter nothing for any one of the three position  co-ordinate textboxes | Error message displays:  ’Position’ fields must not be empty | Pass |
| 15 | 3, 19, 20 | ObjectForm | Erroneous | Test input fields & validation | Enter any combination of characters in any one of the three position  co-ordinate textboxes. | Field validation doesn’t allow illegal characters into the fields. | Fail |
| 16 | 3, 19, 20 | ObjectForm | Normal | Test input fields & validation | Enter position co-ordinates (x, y, z) (“100”, “-1000”, “3E+20”) | Value accepted. | Pass |
| 17 | 3, 19, 21 | ObjectForm | Erroneous | Test input fields & validation | Enter nothing for any one of the three velocity  vector textboxes | Error message displays:  ‘Velocity’ fields must not be empty. | Pass |
| 18 | 3, 19, 21 | ObjectForm | Erroneous | Test input fields & validation | Enter any combination of characters in any one of the three velocity vector textboxes. | Field validation doesn’t allow illegal characters into the fields. | Pass |
| 19 | 3, 19, 21 | ObjectForm | Normal | Test input fields & validation | Enter velocity vector (x, y, z)  (“-10”, “6E-2”, “0”) | Value accepted. | Pass |
| 20 | 3, 19, 22 | ObjectForm | Erroneous | Test input fields & validation | Enter nothing for the mass | Error message displays:  ‘Mass’ field must not be empty. | Pass |
| 21 | 3, 19, 22 | ObjectForm | Erroneous | Test input fields & validation | Enter mass of  “-10” | Error message displays:  ‘Mass’ field must be greater than 0. | Pass |
| 22 | 3, 19, 22 | ObjectForm | Boundary | Test input fields & validation | Enter mass of  “0” | Error message displays:  ‘Mass’ field must be greater than 0. | Fail |
| 23 | 3, 19, 22 | ObjectForm | Erroneous | Test input fields & validation | Enter any combination of characters for the mass. | Field validation doesn’t allow illegal characters into the field. | Pass |
| 24 | 3, 19, 22 | ObjectForm | Normal | Test input fields & validation | Enter mass of “1E+25” | Value accepted. | Pass |
| 25 | 3, 19, 23 | ObjectForm | Erroneous | Test input fields & validation | Enter nothing for the radius | Error message displays:  ‘Radius field must not be empty. | Pass |
| 26 | 3, 19, 23 | ObjectForm | Erroneous | Test input fields & validation | Enter radius of  “-1000” | Error message displays:  ‘Radius field must be greater than 0. | Pass |
| 27 | 3, 19, 23 | ObjectForm | Boundary | Test input fields & validation | Enter radius of  “0” | Error message displays:  ‘Radius field must be greater than 0. | Pass |
| 28 | 3, 19, 23 | ObjectForm | Erroneous | Test input fields & validation | Enter any combination of characters for the radius. | Field validation doesn’t allow illegal characters into the field. | Pass |
| 29 | 3, 19, 23 | ObjectForm | Normal | Test input fields & validation | Enter radius of “1000” | Value accepted. | Pass |
| 30 | 3, 19 | ObjectForm | Normal | Test input fields & validation | Enter nothing for the orbital speed | Value accepted, defaults to 0. | Pass |
| 31 | 3, 19 | ObjectForm | Erroneous | Test input fields & validation | Enter any combination of characters for the orbital speed. | Field validation doesn’t allow illegal characters into the field. | Pass |
| 32 | 3, 19 | ObjectForm | Normal | Test input fields & validation | Enter orbital speed of “100” | Value accepted. | Pass |
| 33 | 3, 19 | ObjectForm | Normal | Test input fields & validation | Enter nothing for the obliquity. | Value accepted, defaults to 0. | Pass |
| 34 | 3, 19 | ObjectForm | Erroneous | Test input fields & validation | Enter any combination of characters for the obliquity. | Field validation doesn’t allow illegal characters into the field. | Pass |
| 35 | 3, 19 | ObjectForm | Normal | Test input fields & validation | Enter obliquity of 10 degrees. | Value accepted. | Pass |
| 36 | 3, 6, 19 | ObjectForm | Normal | Test input fields & validation | Check the “Trails Active?” checkbox. | Checkbox is ticked, Trail Length and Trail Colour fields become editable. | Pass |
| 37 | 3, 6, 19 | ObjectForm | Normal | Test input fields & validation | Uncheck the “Trails Active?” checkbox. | Checkbox is unticked, Trail Length and Trail Colour fields are no longer editable. | Pass |
| 38 | 3, 6, 19 | ObjectForm | Normal | Test input fields & validation | Enter nothing for the trail length. | Value accepted, defaults to 1000. | Pass |
| 39 | 3, 6, 19 | ObjectForm | Normal | Test input fields & validation | Enter “-1” for the trail length. | Field validation doesn’t allow illegal characters into the field. | Pass |
| 40 | 3, 6, 19 | ObjectForm | Erroneous | Test input fields & validation | Enter any combination of characters for the trail length. | Field validation doesn’t allow illegal characters into the field. | Pass |
| 41 | 3, 6, 19 | ObjectForm | Normal | Test input fields & validation | Enter Trail Length of “1000” | Value accepted. | Pass |
| 42 | 3, 5, 6, 19 | ObjectForm | Normal | Test input fields & validation | Click “Trail Colour” button, set the colour to light blue and click ok. | Value accepted, trail colour box displays the colour chosen. | Pass |
| 43 | 3, 5, 6, 19 | ObjectForm | Normal | Test input fields & validation | Click “Object Colour” button, set the colour to light blue and click ok. | Value accepted, object colour box displays the colour chosen. | Pass |
| 44 | 3, 19, 20, 21, 22, 23 | ObjectForm  Existing  Objects  Form | Normal | Test input fields & validation  Test database methods  Test navigation | Click “Copy from Existing” button.  Select “Sun”  Click “Confirm”. button. | System pulls a list of objects from database, displays the names in a list.  When confirm button clicked, the data is copied into the object form fields. | Pass |
| 45 | 3, 19, 20 | ObjectForm | Normal | Test input fields & validation  Test database methods  Test navigation | Click “Presets” button next to “Initial Position” fields.  Select “Mercury”  Click “Confirm”. button. | System pulls a list of objects from database, displays the names in a list.  When confirm button clicked, the data is copied into the position X, Y and Z fields. | Pass |
| 46 | 3, 19, 21 | ObjectForm | Normal | Test input fields & validation  Test database methods  Test navigation | Click “Presets” button next to “Initial Velocity” fields.  Select “Mars”  Click “Confirm”. button. | System pulls a list of objects from database, displays the names in a list.  When confirm button clicked, the data is copied into the velocity X, Y and Z fields. | Pass |
| 47 | 3, 6, 7, 19, 20, 21, 22, 23 | ControlForm  ObjectForm | Normal | Test navigation | Click “Add” button  Fill in form with valid values.  Click “Confirm”. button. | ObjectForm opens, valid values are filled in, and ObjectForm is closed. New object appears in list on ControlForm. | Pass |
| 48 | 3, 6, 7, 19, 20, 21, 22, 23 | ControlForm  ObjectForm | Normal | Test navigation | Select an object from the list.  Click “Edit” button  Fill in form with valid values.  Click “Confirm”. button. | ObjectForm opens, values are changed (to still be within valid values), and ObjectForm is closed. Object’s data has changed. If the name of the object was changed, name changes in list on ControlForm. | Pass |
| 49 | 3, 19 | ControlForm | Normal | Test navigation | Select an object from the list.  Click “Remove” button | Selected object is removed from list on ControlForm. | Pass |
| 50 | 3, 19, 26 | ControlForm  SaveAs  Form | Normal | Test navigation | Create new simulation, add objects to it.  Click “Save as Preset” button.  Enter “Preset-Test-1” and click “Save” button. | SaveAsForm opens. Name is entered, and simulation is saved.  Saved simulation can now be opened from the PresetForm. | Pass |
| 51 | 3, 19, 24 | ControlForm | Normal | Test navigation | Create new simulation, add objects to it.  Click “Export to File” button.  Choose a location and enter “Export-Test-1” and click “Save” button. | File saving dialog is opened. Name is entered, and file is saved. | Pass |
| 52 | 3, 7, 9, 16 | Main  Window  ControlForm | Normal | Test simulation rendering  Test simulation view controls | Load the “Solar System” preset simulation.  Left click and drag with the mouse over the view window to change the camera view. | OpenGL view window opens.  3D representations of the major bodies in our Solar System are displayed and are moving in orbits.  Camera view changes work seamlessly. | Pass |
| 53 | 3, 7, 16 | Main  Window  ControlForm | Normal | Test simulation view controls  Test simulation rendering | Load the “Solar System” preset simulation.  Click “Pause” button.  Click “Play” button. | OpenGL view window opens.  3D representations of the major bodies in our Solar System are displayed and are moving in orbits. Simulation pauses and plays when told to. | Pass |
| 54 | 3, 7, 16 | Main  Window  ControlForm | Normal | Test simulation view controls  Test simulation rendering | Load the “Solar System” preset simulation.  Press spacebar.  Press spacebar again. | OpenGL view window opens.  Simulation displays.  Simulation pauses and plays on spacebar press. | Pass |
| 55 | 3, 7, 8, 9, 16 | Main  Window  ControlForm | Normal | Test simulation view controls  Test simulation rendering | Load the “Solar System” preset simulation.  Move the “Speed Control” slider to the right. | OpenGL view window opens.  Simulation displays.  Speed of simulation increases. | Pass |
| 56 | 3, 7, 8, 9, 16 | Main  Window  ControlForm | Normal | Test simulation view controls  Test simulation rendering | Load the “Solar System” preset simulation.  Move the “Speed Control” slider to the right.  Move the “Speed Control” slider to the left. | OpenGL view window opens.  Simulation displays.  Speed of simulation increases.  Simulation plays in reverse, to the beginning of the simulation where it pauses. | Pass |
| 57 | 3, 7, 16 | Main  Window  ControlForm | Normal | Test simulation view controls  Test simulation rendering | Load the “Solar System” preset simulation.  Press “Restart” button. | OpenGL view window opens.  Simulation displays.  Simulation returns to the beginning and starts again. | Pass |
| 58 | 3, 7, 16 | Main  Window  ControlForm | Normal | Test simulation view controls  Test simulation rendering | Load the “Solar System” preset simulation.  Press the “R” key. | OpenGL view window opens.  Simulation displays.  Simulation returns to the beginning and starts again. | Pass |
| 59 | 3, 6, 7, 9, 16 | Main  Window  ControlForm  ObjectForm | Normal | Test simulation rendering  Test object control form | Load the “Solar System” preset simulation.  View the trails behind the objects. | OpenGL view window opens.  Simulation displays.  Trails should display in the path of the objects. | Pass |
| 60 | 3, 6, 7, 9, 16, 19 | Main  Window  ControlForm  ObjectForm | Normal | Test simulation rendering  Test object control form | Load the “Solar System” preset simulation.  Disable trail for “Earth”. | OpenGL view window opens.  Simulation displays.  Simulation updates and no longer renders a trail behind the Earth. | Pass |
| 61 | 3, 5, 6, 7, 9, 16, 19 | Main  Window  ControlForm  ObjectForm | Normal | Test simulation Test simulation rendering  Test object control form | Load the “Solar System” preset simulation.  Change the colour of the trail for “Mercury” to red. | OpenGL view window opens.  Simulation displays.  Simulation updates and the colour of Mercury’s trail changes to red. | Pass |
| 62 | 3, 6, 7, 9, 16, 19 | Main  Window  ControlForm  ObjectForm | Normal | Test simulation rendering  Test object control form | Load the “Solar System” preset simulation.  Reduce trail length of Earth’s trail to “1500” | OpenGL view window opens.  Simulation displays.  Simulation updates and trail behind the Earth is half as long. | Pass |
| 63 | 3, 7, 9, 16 | Main  Window  ControlForm | Normal | Test simulation rendering  Test simulation view controls | Load the “Solar System” preset simulation.  Use the mouse scroll wheel to zoom in and out of the simulation. | OpenGL view window opens. Simulation displays. Simulation zooms in and out as told to. | Pass |
| 64 | 3, 7, 9, 10, 16 | Main  Window  ControlForm | Normal | Test simulation rendering  Test simulation view controls | Load the “Solar System” preset simulation.  Use the ControlForm menu to select the current focus of the simulation. | OpenGL view window opens. Simulation displays. Simulation focus changes as requested. | Pass |

## White Box Testing Evidence

Below are screenshots from each test, along with a description of what happened, and what was changed if the test failed.

Where relevant, videos have been recorded of the test taking place and uploaded to YouTube. The video will be linked with the individual test, however will also be listed here:

**Test 52:** https://youtu.be/lmW73UAqAjo

**Test 53**: https://youtu.be/\_XleaTKUlBY

**Test 54:** https://youtu.be/oUzT\_Hdp-Io

**Tests 55 & 56:** https://youtu.be/XNnRGANL4Lk

**Test 57:** https://youtu.be/dCxbOyNGdic

**Test 58:** https://youtu.be/5GI8sxKO1YY

**Tests 59 & 60:** https://youtu.be/Q4JtV-ypBdI

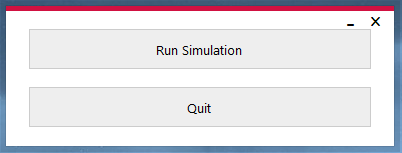
**Test 61:** https://youtu.be/oz-B3ddN1vg

**Test 62:** https://youtu.be/W6yv8mwcdIQ

**Test 63:** https://youtu.be/NpFVJgtORhI

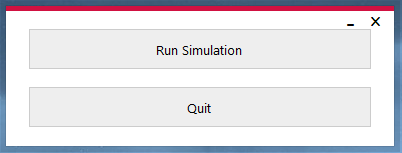
These recordings show the simulation itself being ran, as well as the interface for the simulation (ControlForm) being used in conjunction with it. Within the test evidence itself, screenshots will also be provided, as well as a basic description of what happened. All tests were also carried out on one of the Physics Department’s laptops, satisfying both objectives 2 and 4.

Test 1:

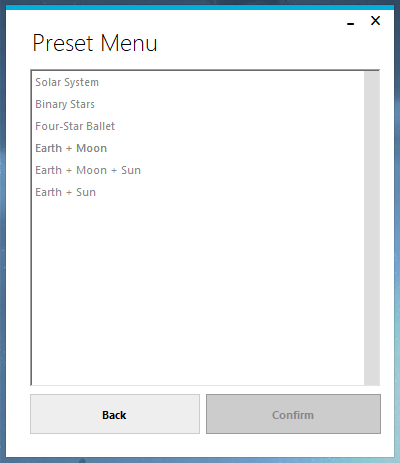


The “Run Simulation” button was pressed, SimForm opened and MainForm closed.

Test 2:

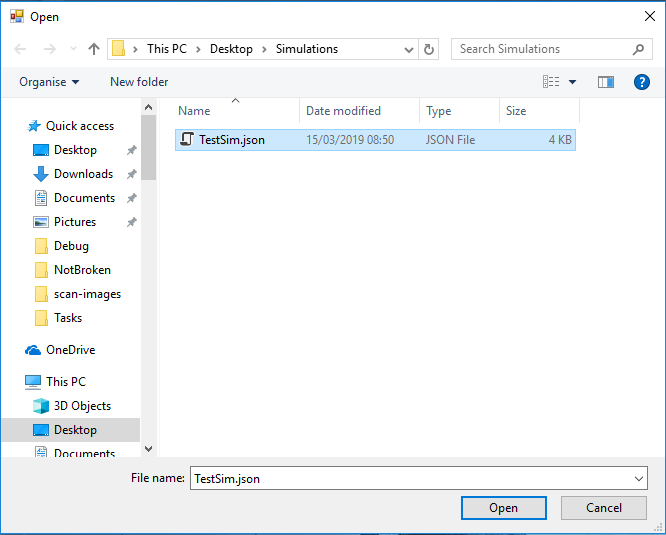


The “Quit” button was pressed, and the application exited.

Test 3:



The “Load from Presets” button was pressed, PresetForm opened and MainForm closed.

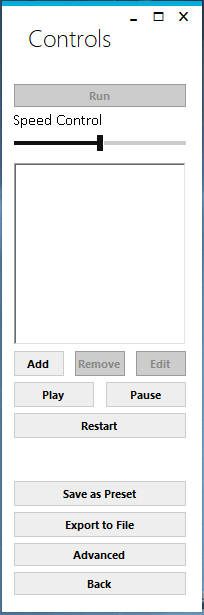
Test 4:





The “Load From File” button was pressed, and a file selection dialog opened.

“TestSim.json” was selected, and the “Open” button was pressed. ControlForm opened (with the information from the file loaded), and SimForm closed.

Test 5:

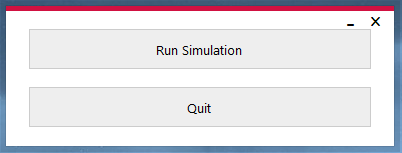


The “New Custom Simulation” button was pressed.

ControlForm opened, and SimForm closed.

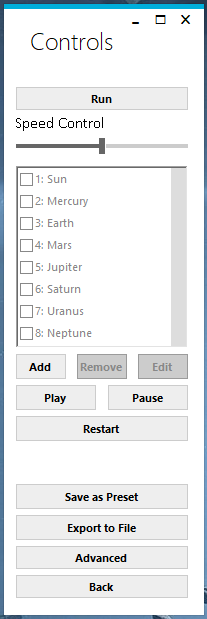
Test 6:

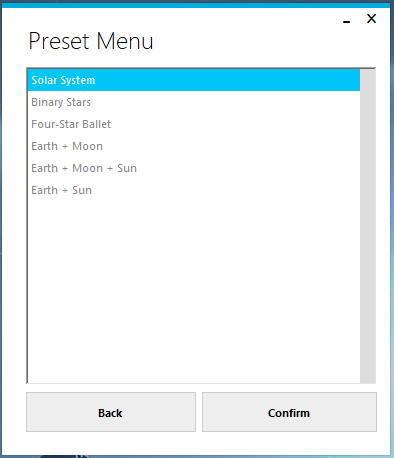




The “Back” button was pressed.

MainForm opened, and SimForm closed.

Test 7:

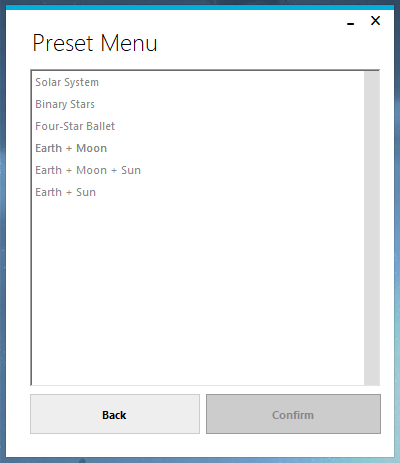


The “Solar System” preset is selected, and the “Confirm” button

is pressed. PresetForm closes, and ControlForm opens with the

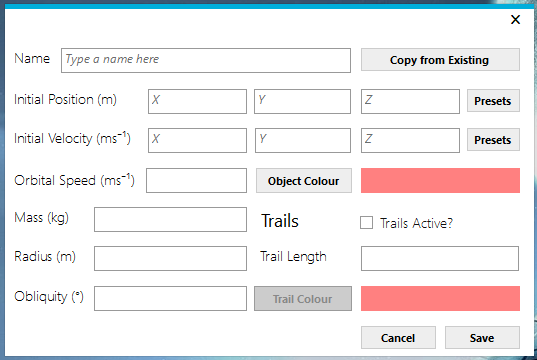
“Solar System” preset loaded.

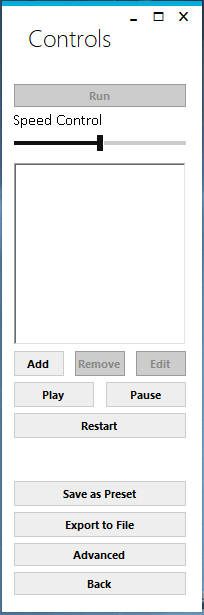
Test 8:





The “Back” button was pressed. SimForm opened, and PresetForm closed.

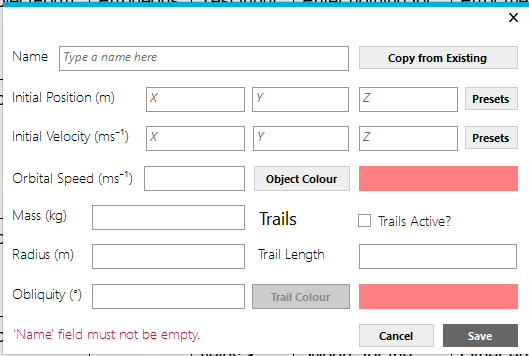
Test 9:



The “Add” button is pressed on the ControlForm, in a new custom

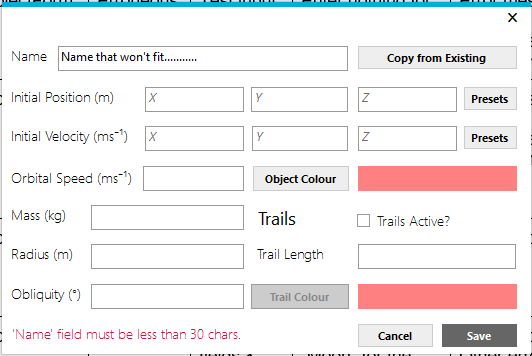
simulation. ObjectForm opens, with blank editable fields.

Test 10:



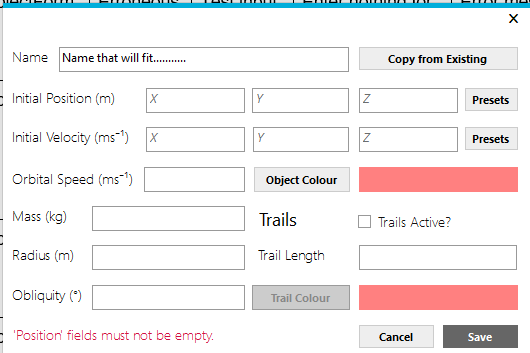
No name was entered into the “Name” textbox. When the “Save” button is pressed, an error appears for the “Name” textbox stating that the name must not be blank.

Test 11:



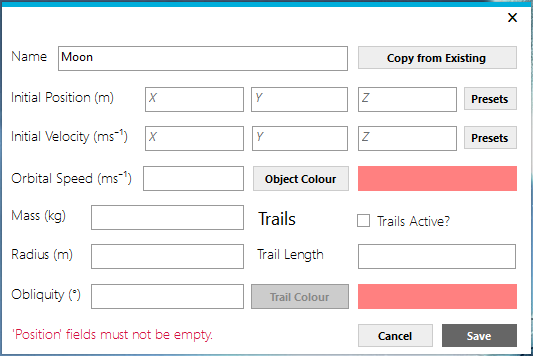
The name “Name that won't fit...........” was entered into the “Name” textbox. When the “Save” button is pressed, an error appears for the “Name” textbox stating that the name is too long.

Test 12:



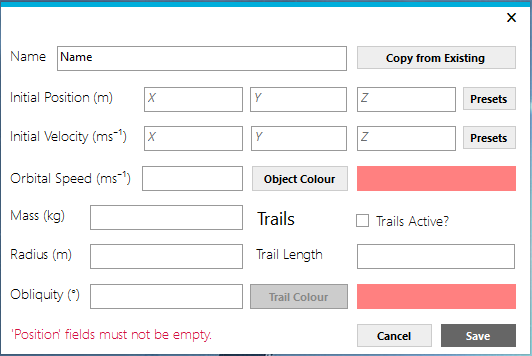
The name “Name that will fit...........” was entered into the “Name” textbox. When the “Save” button is pressed, no error appears for the “Name” textbox.

Test 13:

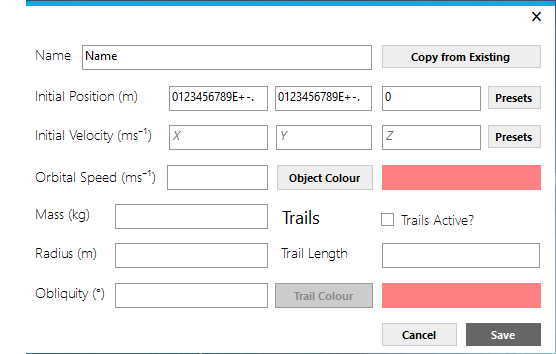


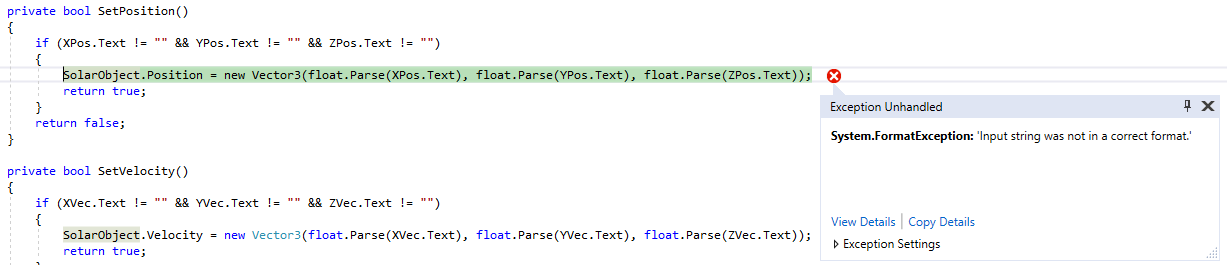
The name “Moon” was entered into the “Name” textbox. When the “Save” button is pressed, no error appears for the “Name” textbox.

Test 14:

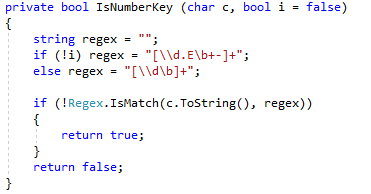


No values were entered into any of the position textboxes. When the “Save” button was pressed, an error appeared stating that values must be entered.

Test 15 (FAIL):

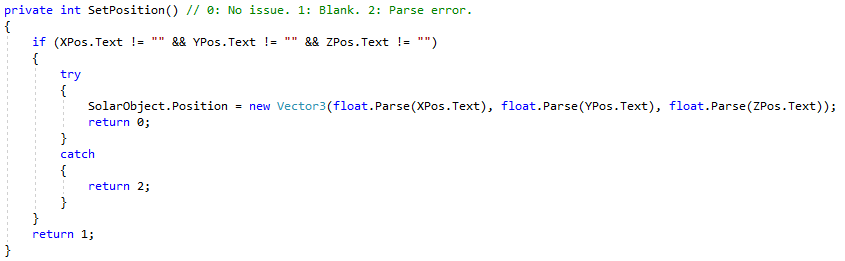


While the field validation written for the position fields worked (i.e. Only a specific character set could be entered into the text boxes, comprising of the numbers 0-9, E, full stop, + and -), there was no final validation to check whether or not the text could actually be converted into a valid number.

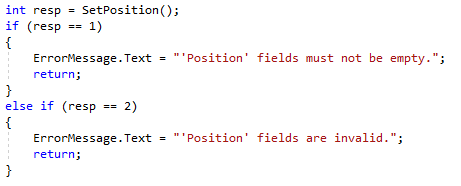


To fix this, I changed the SetPosition method to return different values depending on the error returned. I also surrounded the code block in a try/catch statement, in order to handle the potential exception gracefully.

I therefore also had to update the code inside the “Save” button click event to return a different error message depending on the error that occurred.

New Code:

New code:

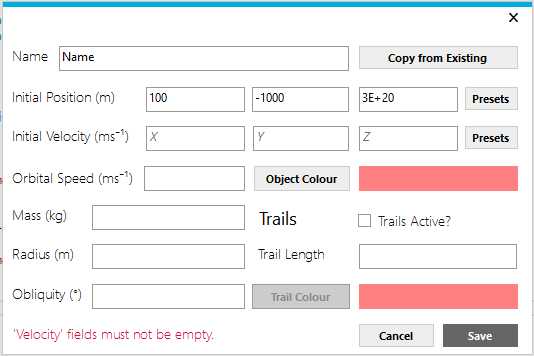


I re-did Test 15, and it passed:



While fixing this error, I realised that it the same situation would apply to all fields in this form, and so I modified the code to fix all of these issues at the same time.

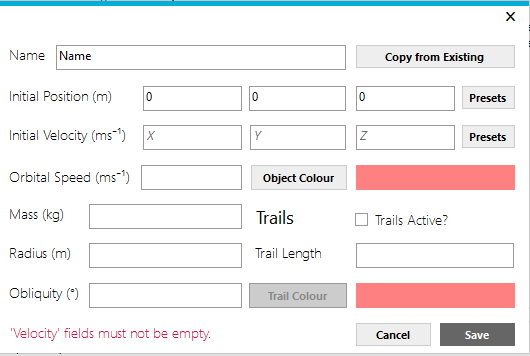
Test 16:



The values “100”, “-1000”, “3E+20” were entered into the relevant position textboxes.

When the “Save” button was pressed, no error appeared for the “Position” textboxes.

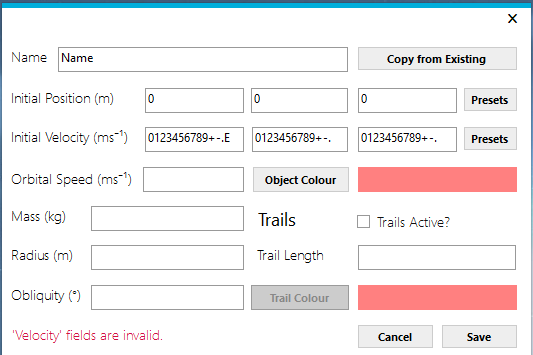
Test 17:



No values were entered into any of the velocity textboxes.

When the “Save” button was pressed, an error message appeared stating that the fields must not be empty.

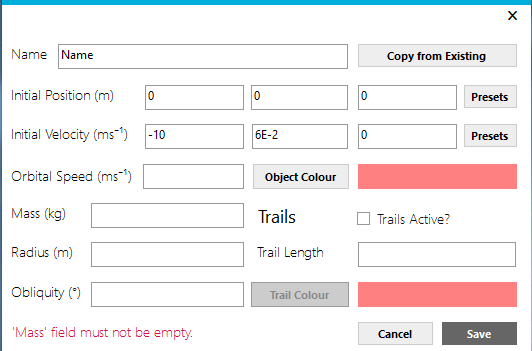
Test 18:



Only a specific set of characters were allowed into the fields, and this functioned correctly.

However, there was not initially any check to see whether the value entered could actually be converted properly. This bug was fixed during Test 15, and is documented there. Therefore, this test passed, and when the “Save” button was pressed, an error appeared stating that the fields were invalid.

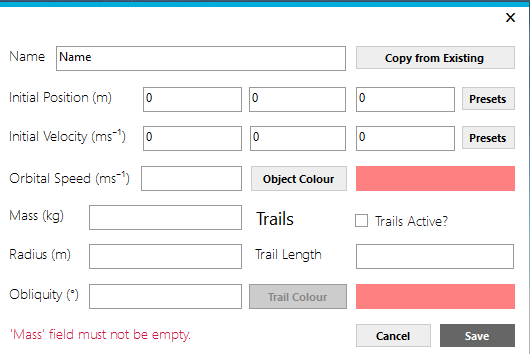
Test 19:



The values “-10”, “6E-2”, “0” were entered into the relevant velocity textboxes.

When the “Save” button was pressed, no error appeared for the “Velocity” textboxes.

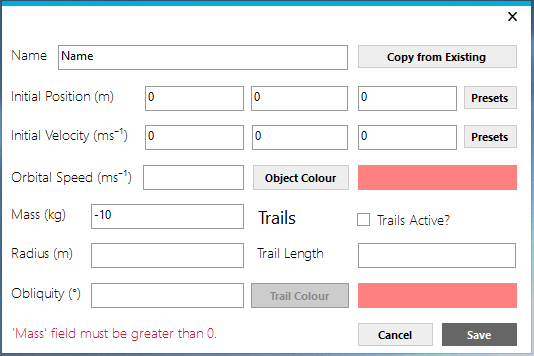
Test 20:



No value was entered into the mass textbox.

When the “Save” button was pressed, an error message appeared stating that the field must not be empty.

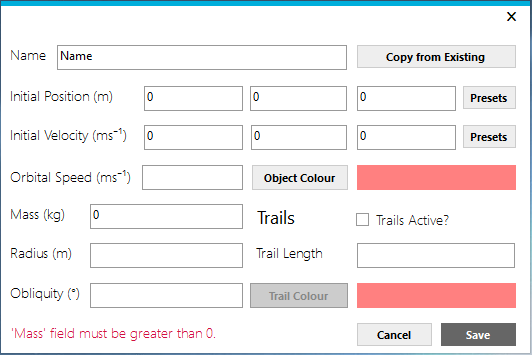
Test 21:



“-10” was entered into the mass textbox.

When the “Save” button was pressed, an error message appeared stating that the value must be greater than 0.

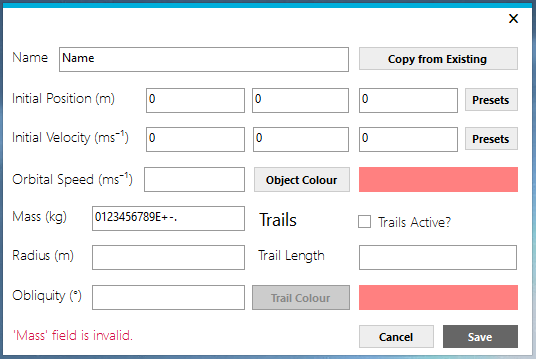
Test 22:



“0” was entered into the mass textbox.

When the “Save” button was pressed, an error message appeared stating that the value must be greater than 0.

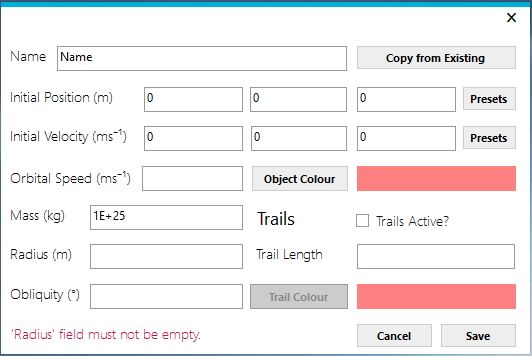
Test 23:



Only a specific set of characters was allowed into the field, and this functioned correctly.

However, there was not initially any check to see whether the value entered could actually be converted properly. This bug was fixed during Test 15, and is documented there. Therefore, this test passed, and when the “Save” button was pressed, an error appeared stating that the field was invalid.

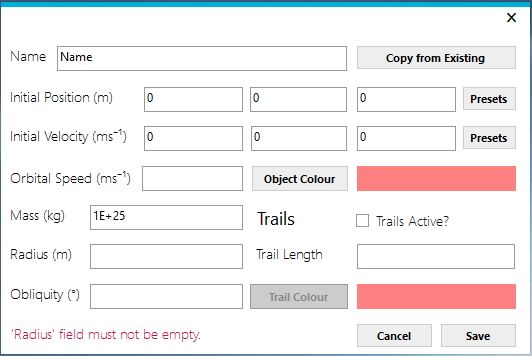
Test 24:



The value 1E+25” was entered into the mass textbox.

When the “Save” button was pressed, no error appeared for the “Mass” textbox.

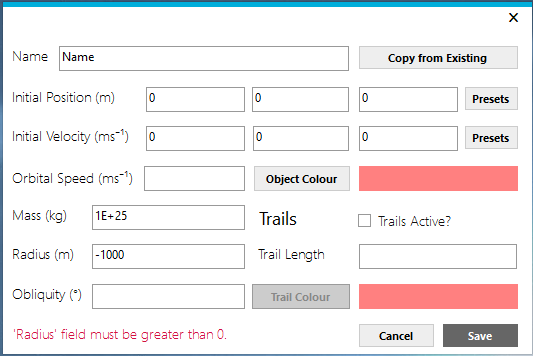
Test 25:



No value was entered into the radius textbox.

When the “Save” button was pressed, an error message appeared stating that the field must not be empty.

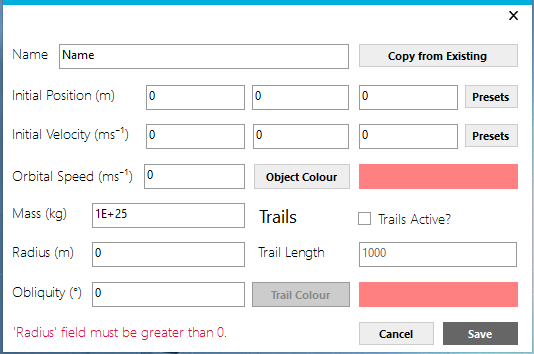
Test 26:



“-1000” was entered into the radius textbox.

When the “Save” button was pressed, an error message appeared stating that the value must be greater than 0.

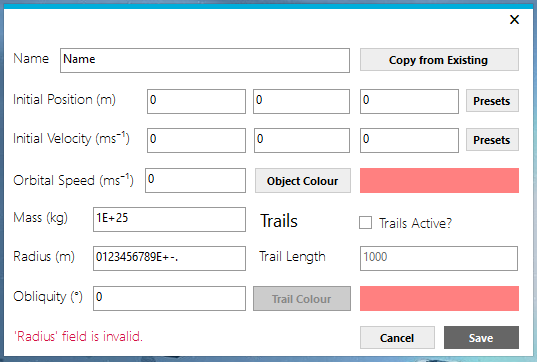
Test 27:



“0” was entered into the radius textbox.

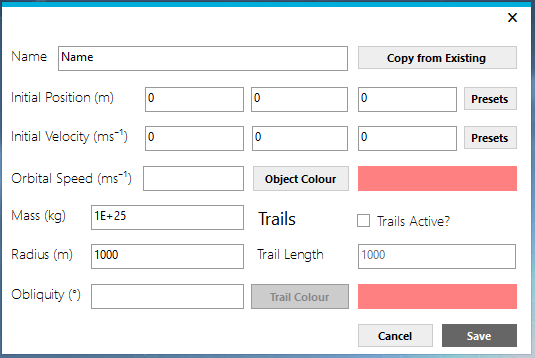
When the “Save” button was pressed, an error message appeared stating that the value must be greater than 0.

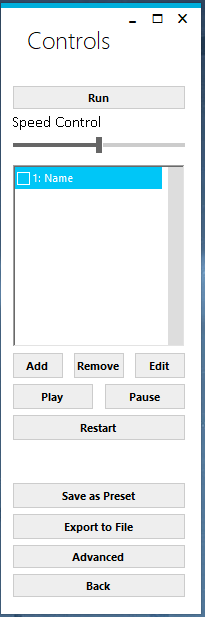
Test 28:



Only a specific set of characters was allowed into the field, and this functioned correctly.

However, there was not initially any check to see whether the value entered could actually be converted properly. This bug was fixed during Test 15, and is documented there. Therefore, this test passed, and when the “Save” button was pressed, an error appeared stating that the field was invalid.

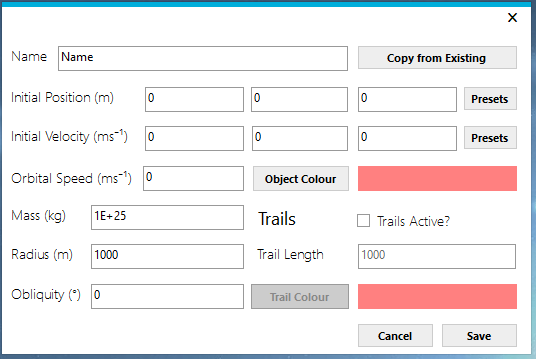
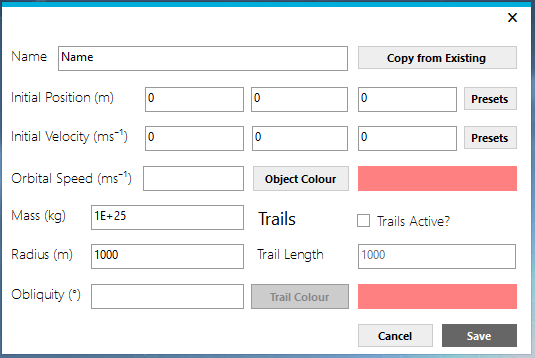
Test 29:



“1000” was entered into the radius textbox.

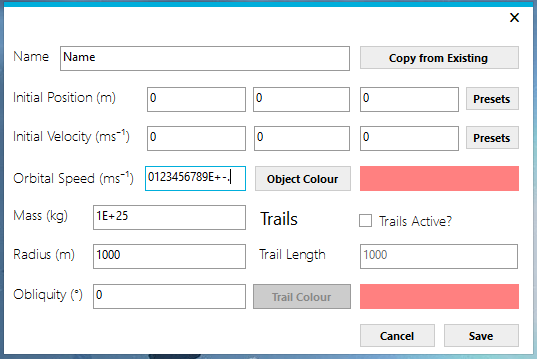
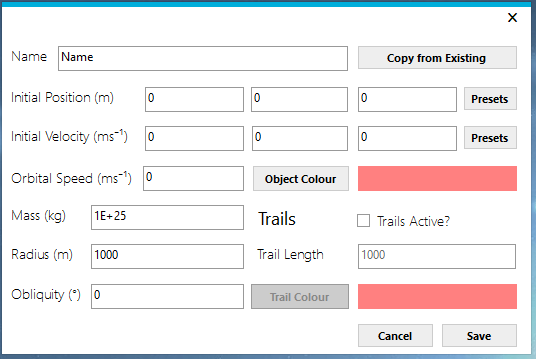
When the “Save” button was pressed, no errors appeared as in this case all required fields were filled in. Therefore, the object was saved, and ObjectForm exited out to the ControlForm, which had the object in the list.

Test 30:



As stated in Test 29, the ObjectForm exited out to the ControlForm despite the orbital speed not being filled in. When I went to edit the object again, the orbital speed’s field had the value “0” in it. This shows that the speed defaulted to “0”, as expected.

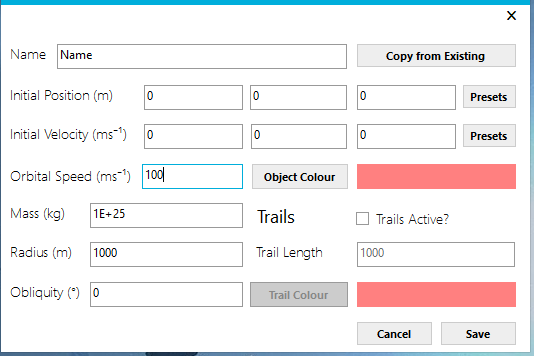
Test 31:



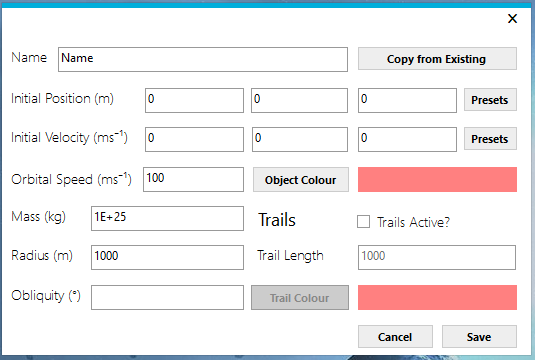
Only a specific set of characters was allowed into the field, and this functioned correctly.

However, there was not initially any check to see whether the value entered could actually be converted properly. This bug was fixed during Test 15, and is documented there. Therefore, this test passed. When the “Save” button is pressed, the orbital speed defaulted to “0” (as it is not a required field), and the ObjectForm exited out to the ControlForm. When I went to edit the object again, the orbital speed had the value “0”, showing that the speed did indeed default to “0” as expected.

Test 32:

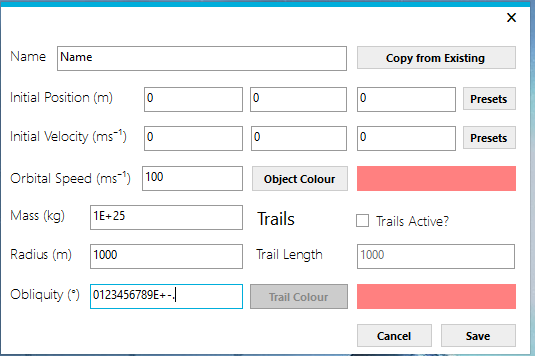


The value was accepted, and the form closed. When the object was opened back up, the value remained.

Test 33:

As stated in Test 29, the ObjectForm exited out to the ControlForm despite the obliquity not being filled in. When I went to edit the object again, the obliquity’s field had the value “0” in it. This shows that the obliquity defaulted to “0”, as expected.

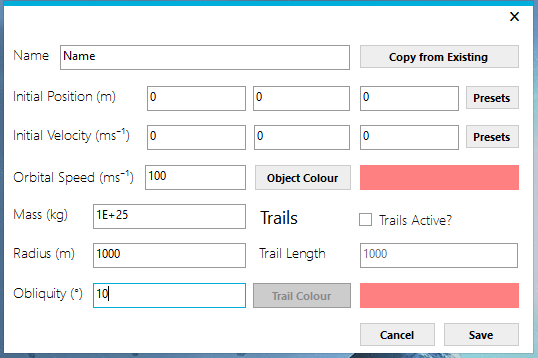
Test 34:



Only a specific set of characters was allowed into the field, and this functioned correctly.

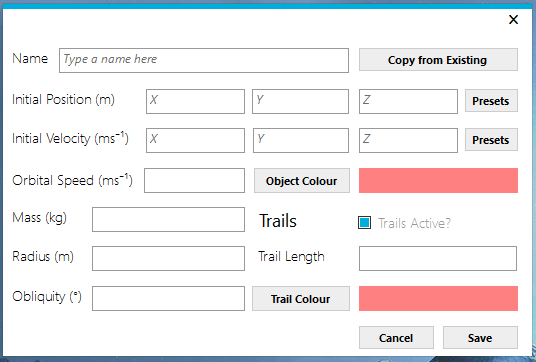
However, there was not initially any check to see whether the value entered could actually be converted properly. This bug was fixed during Test 15, and is documented there. Therefore, this test passed. When the “Save” button is pressed, the obliquity defaulted to “0” (as it is not a required field), and the ObjectForm exited out to the ControlForm. When I went to edit the object again, the obliquity had the value “0”, showing that it did indeed default to “0” as expected.

Test 35:



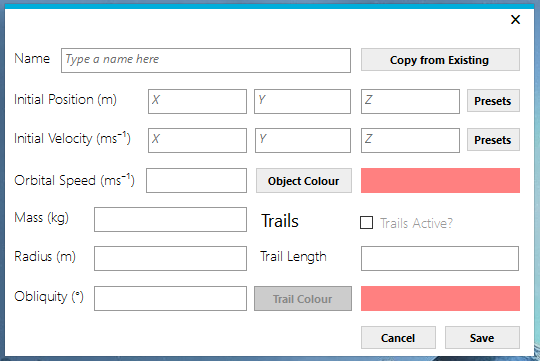
The value was accepted, and the form closed. When the object was opened back up, the value remained.

Test 36:



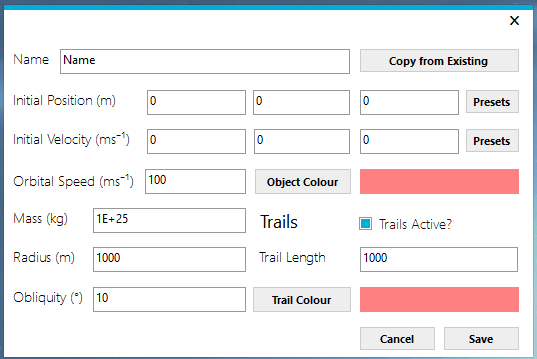
Tickbox checked, other fields become activated.

Test 37:

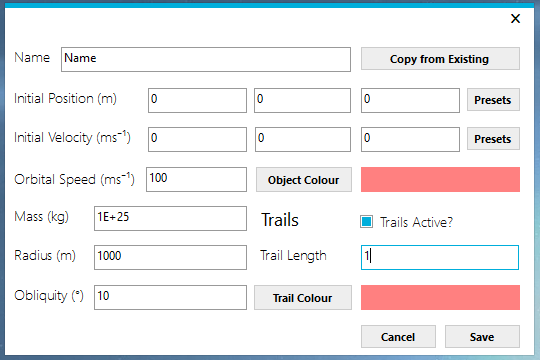


Tickbox unchecked, other fields deactivate.

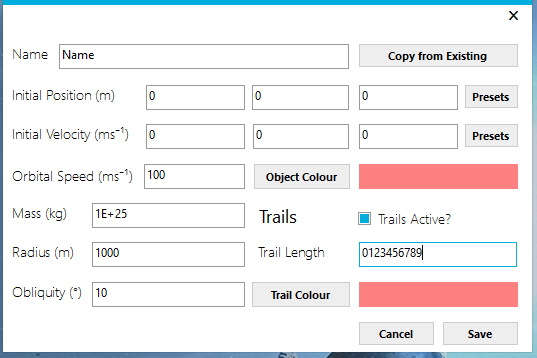
Test 38:



As stated in Test 29, the ObjectForm exited out to the ControlForm despite the trail length not being filled in. When I went to edit the object again, the trail length’s field had the value “1000” in it. This shows that the trail length defaulted to “1000”, as expected.

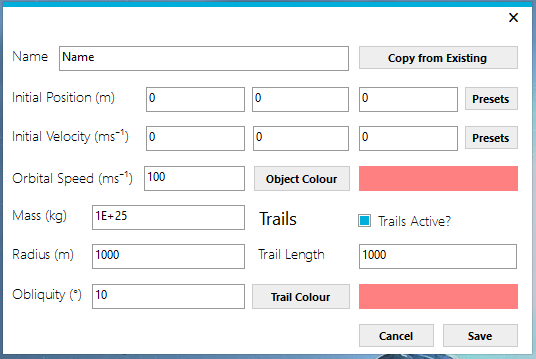
Test 39:

When attempting to type in “-1” for the trail length, I was not allowed to type in the “-“ sign. Therefore, the trail length is 1.

Test 40:

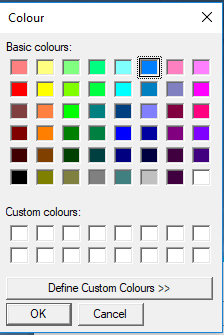
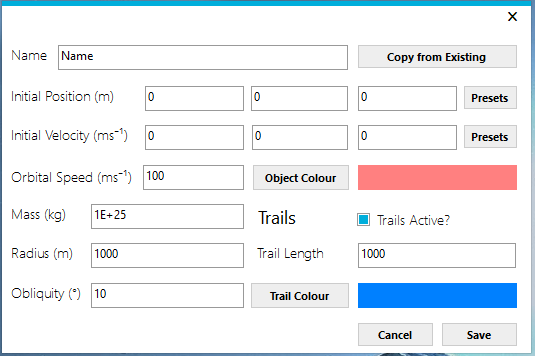
When attempting to type in “ABC0123456789” for the trail length, I was not allowed to type in the letters. Therefore, the trail length is 123456789.

Test 41:



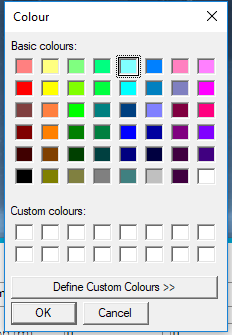
The value was accepted, and the form closed. When the object was opened back up, the value remained.

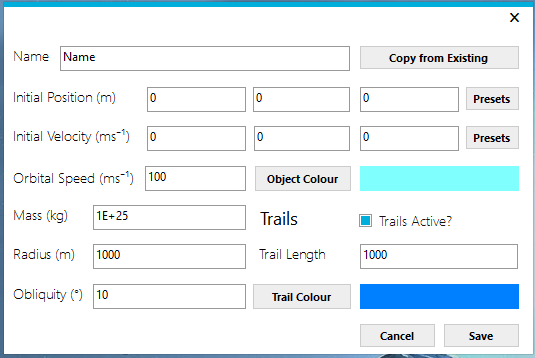
Test 42:



The colour was chosen, and the trail colour box displayed the colour chosen.

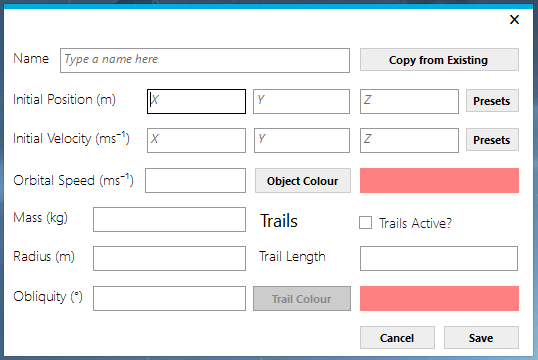
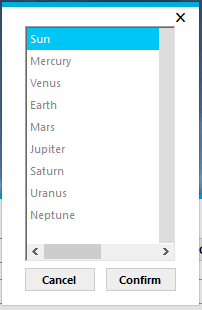
Test 43:

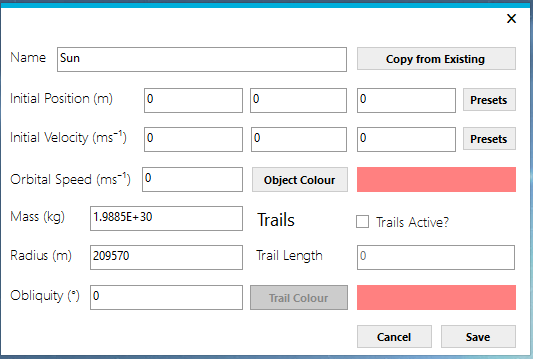




The colour was chosen, and the object colour box displayed the colour chosen.

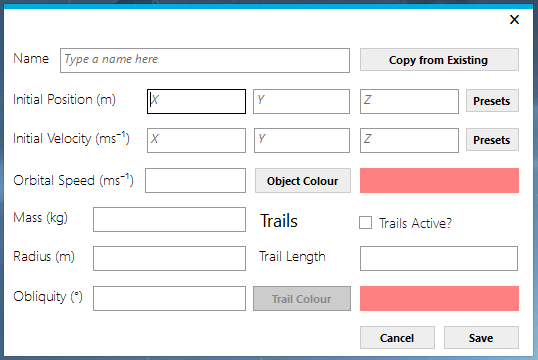
Test 44:



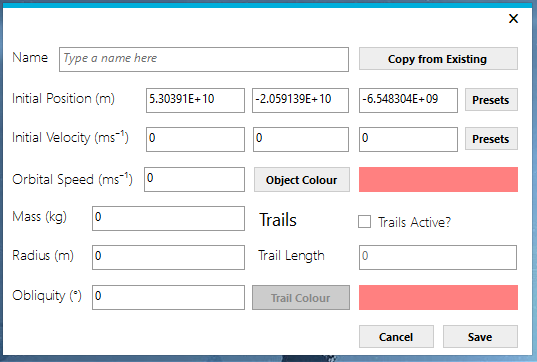


The “Copy from Existing” button was pressed, and ExistingObjectsForm opened. “Sun” was selected, and “Confirm” was clicked. The data for the “Sun” object was copied into the object form successfully.

Test 45:

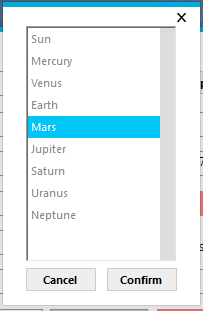


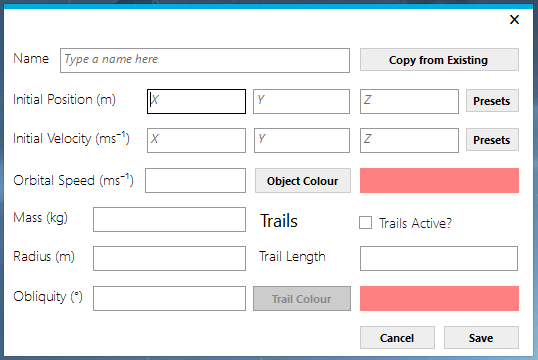


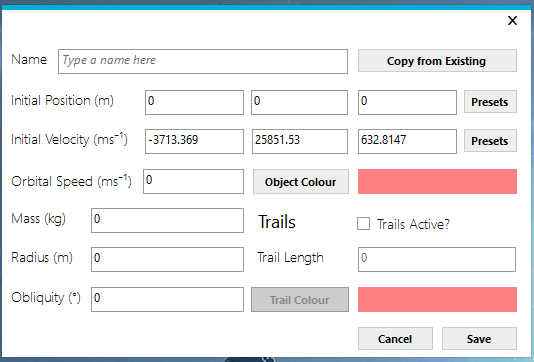


The Position “Preset” button was pressed, and ExistingObjectsForm opened. “Mercury” was selected, and “Confirm” was clicked. The data for the “Mercury” object was copied into the position fields successfully.

Test 46:

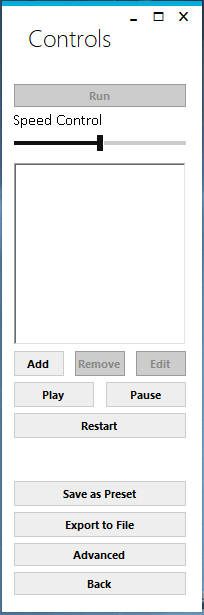
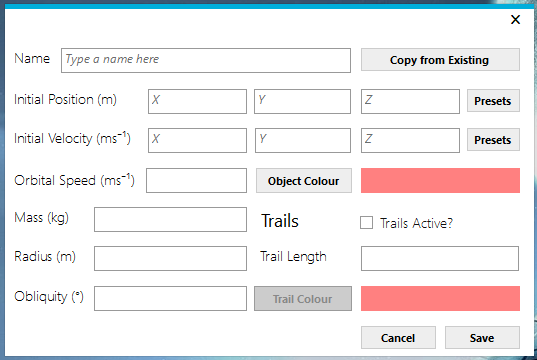
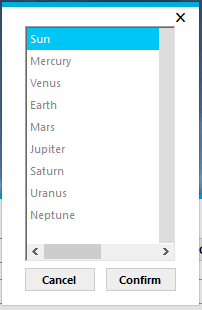


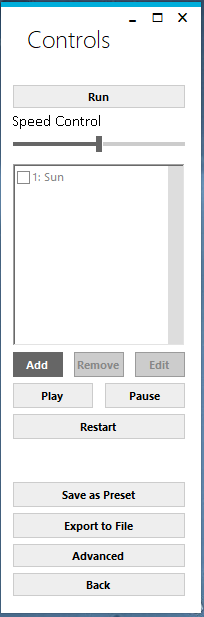


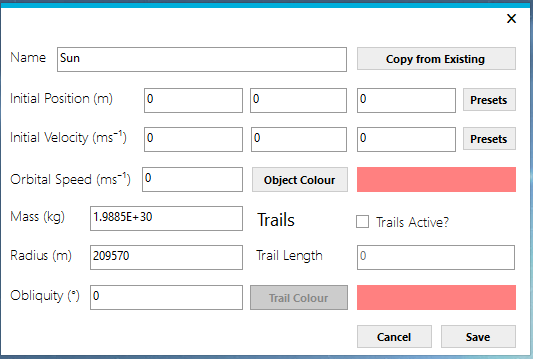


The velocity “Preset” button was pressed, and ExistingObjectsForm opened. “Mars” was selected, and “Confirm” was clicked. The data for the “Mercury” object was copied into the velocity fields successfully.

Test 47:

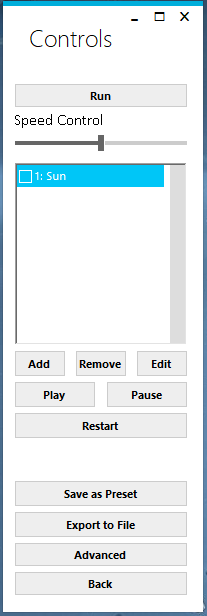


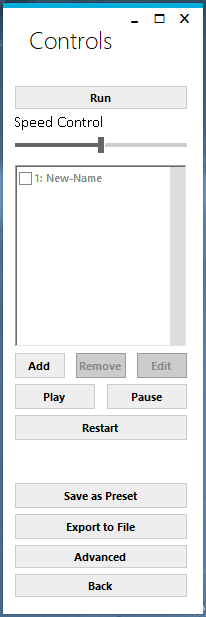


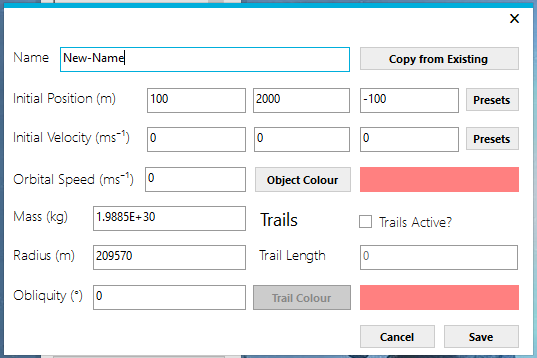


The “Add” button was pressed, and ObjectForm opened. The “Copy from Existing” button was pressed, and ExistingObjectsForm opened. “Sun” was selected, and “Confirm” was clicked. The data for the “Sun” object was copied into the velocity fields successfully. The “Save” button was pressed, and the form closed. The “Sun” object appears in the ControlForm list of objects successfully.

Test 48:

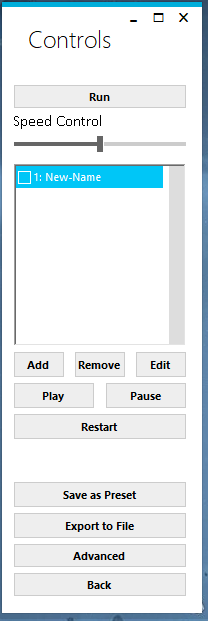


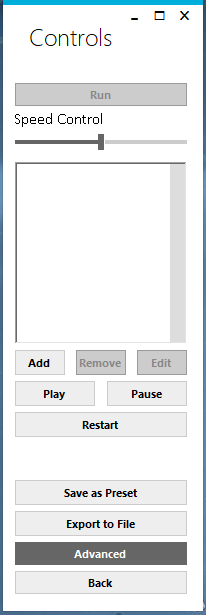




The “Edit” button was pressed when the “Sun” object was selected. ObjectForm opened up, and values were changed (“Sun” was changed to “New-Name”, and the Position values were changed). The “Save” button was pressed, and ObjectForm closed. The object’s name changed in the object list on ControlForm, from “Sun” to “New-Name”.

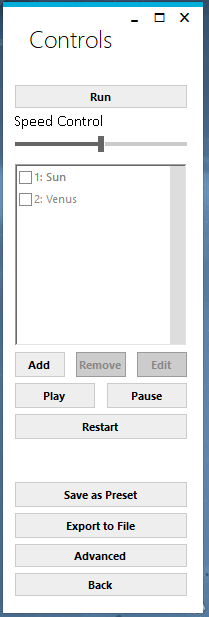
Test 49:

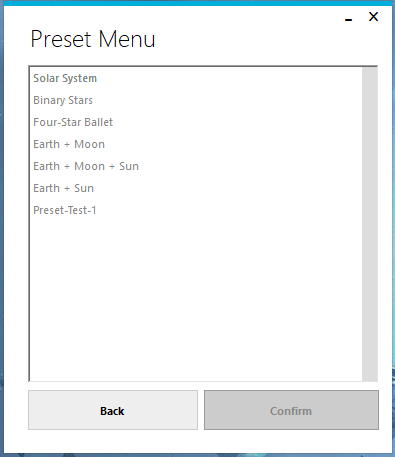


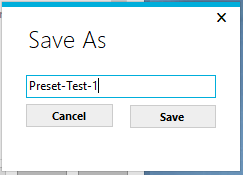


The “New-Name” object was selected in the object list, and the “Delete” button was pressed. The “New-Name” object was removed from the object list successfully.

Test 50:

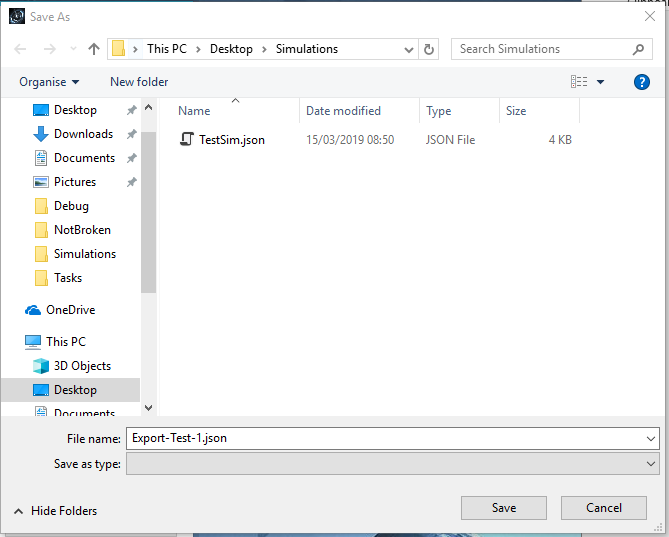


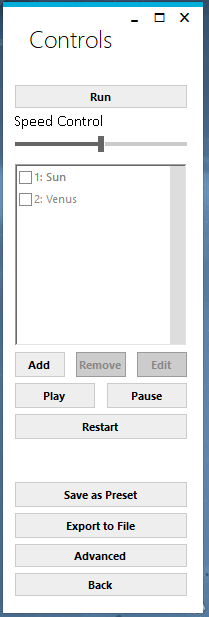


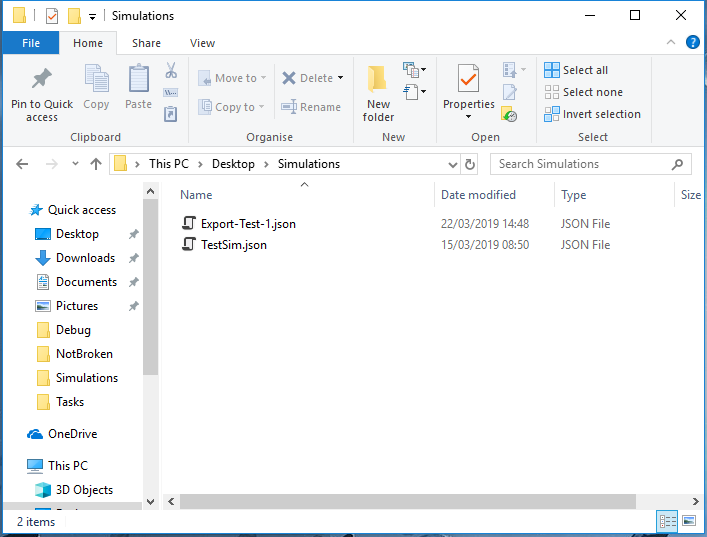


Objects were added to a new custom simulation, and the “Save as Preset” button was pressed. The name “Preset-Test-1” was typed into SaveAsForm, and the “Save” button was pressed. SaveAsForm closed itself, and returned to ControlForm.

When checked, the “Preset-Test-1” simulation was in the PresetForm list of simulations.

Test 51:

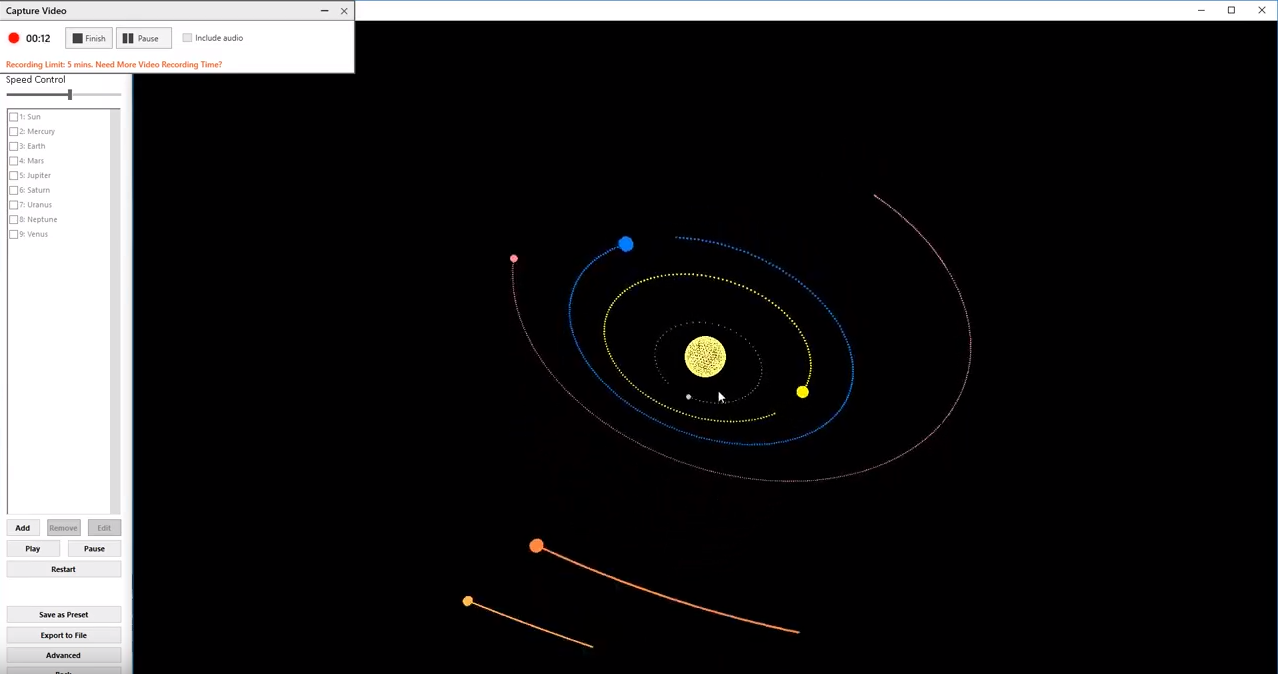


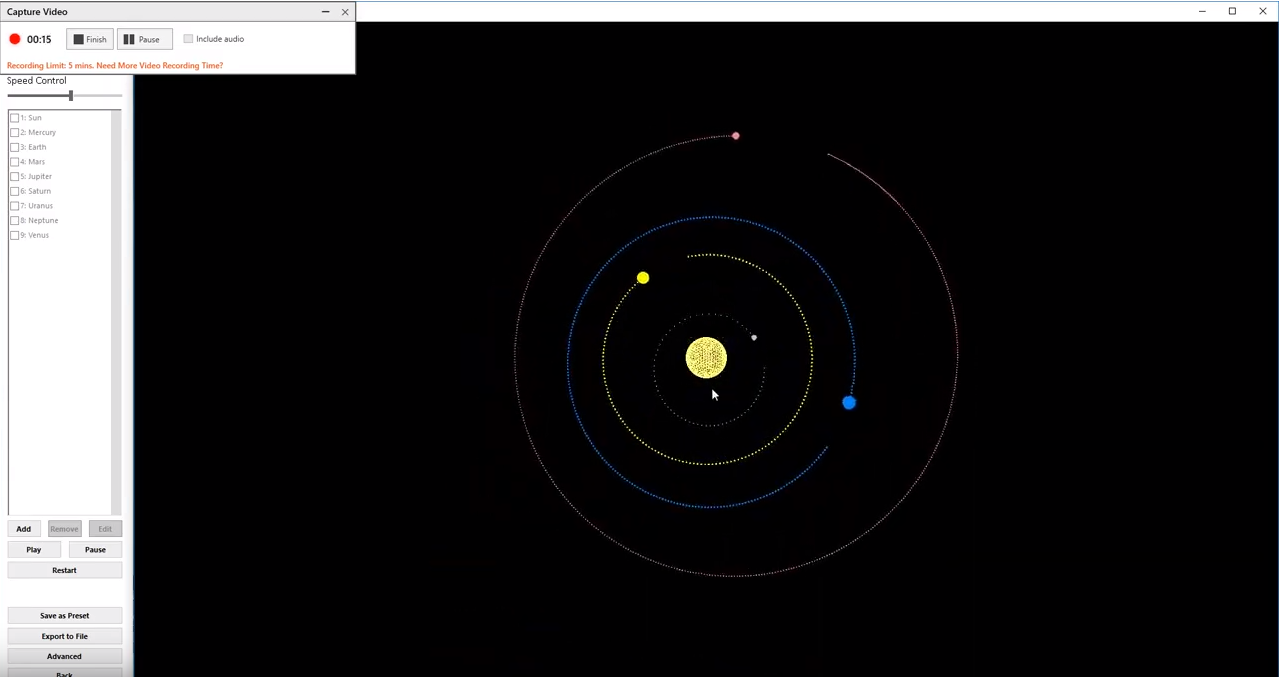


Objects were added to a new custom simulation, and the “Export to File” button was pressed. The name “Export-Test-1” was typed into the file save dialog, and the “Save” button was pressed.

When checked, the “Export-Test-1” simulation was found in the location it had been saved.

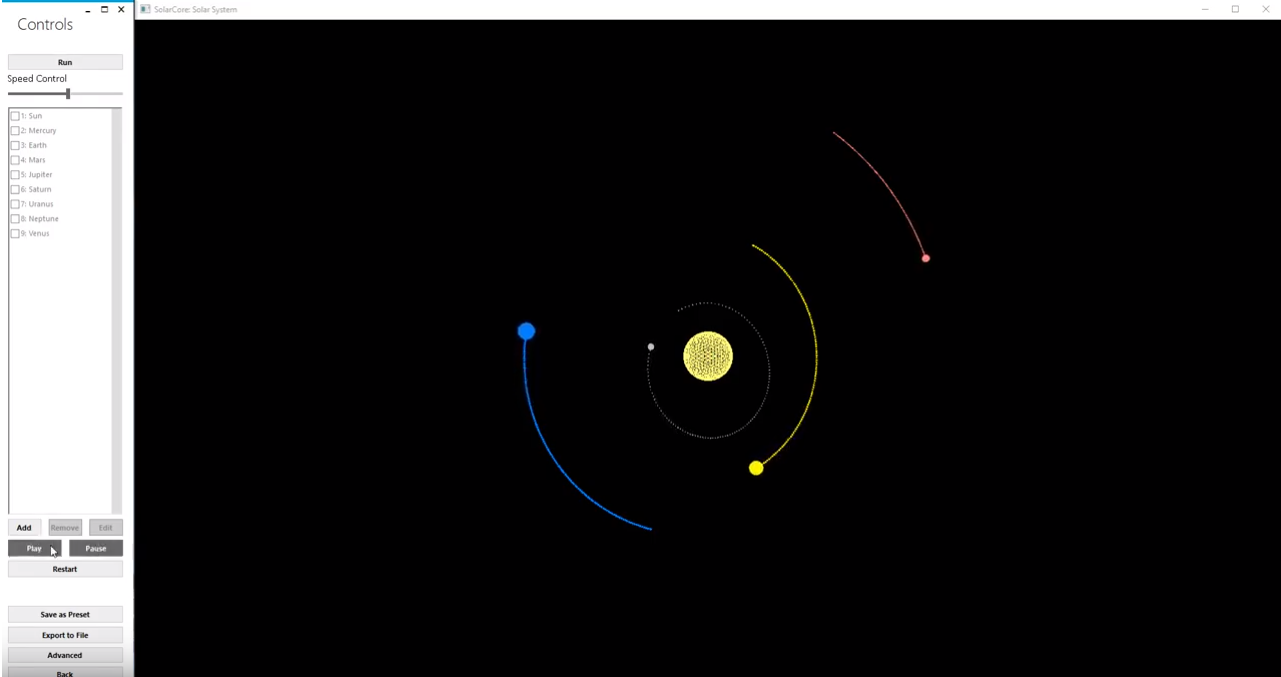
Test 52 (https://youtu.be/lmW73UAqAjo):





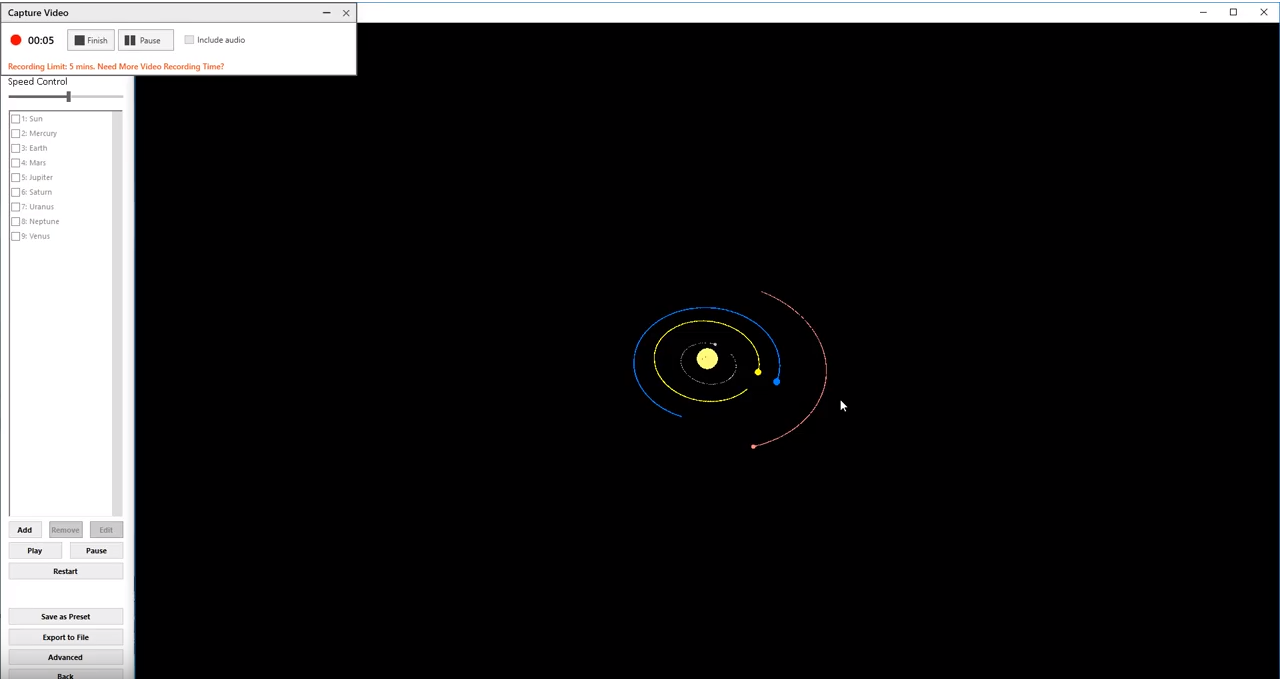
The simulation successfully loaded, and the mouse was used to change the view within the simulation.

Test 53 (https://youtu.be/XleaTKUlBY):



The simulation successfully loaded, and the mouse was used to press the “Pause” and “Play” buttons on the Control Form.

Test 54 (https://youtu.be/oUzTHdp-Io):



The simulation successfully loaded, and the spacebar was used to toggle pause/play for the simulation.

Tests 55 & 56 (https://youtu.be/XNnRGANL4Lk):



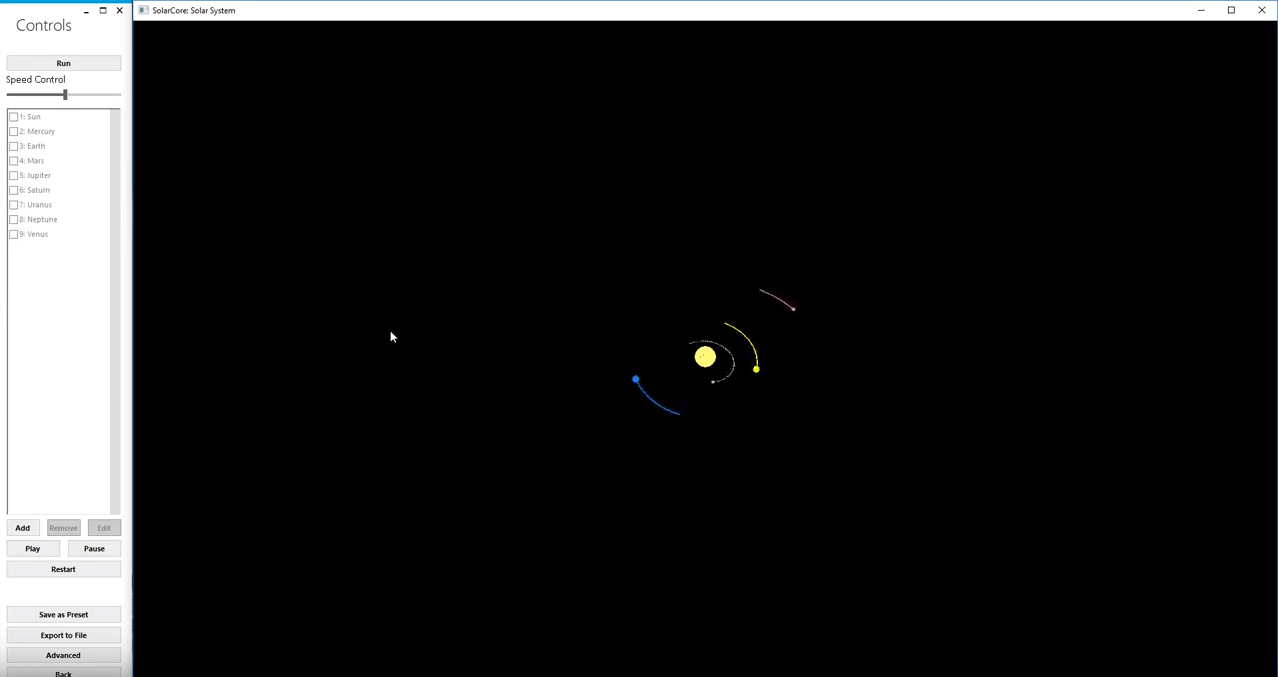
The simulation successfully loaded, and the mouse was used to drag the speed control slider to both the left and right, affecting the simulation depending on where the slider was.

Test 57 (https://youtu.be/dCxbOyNGdic):



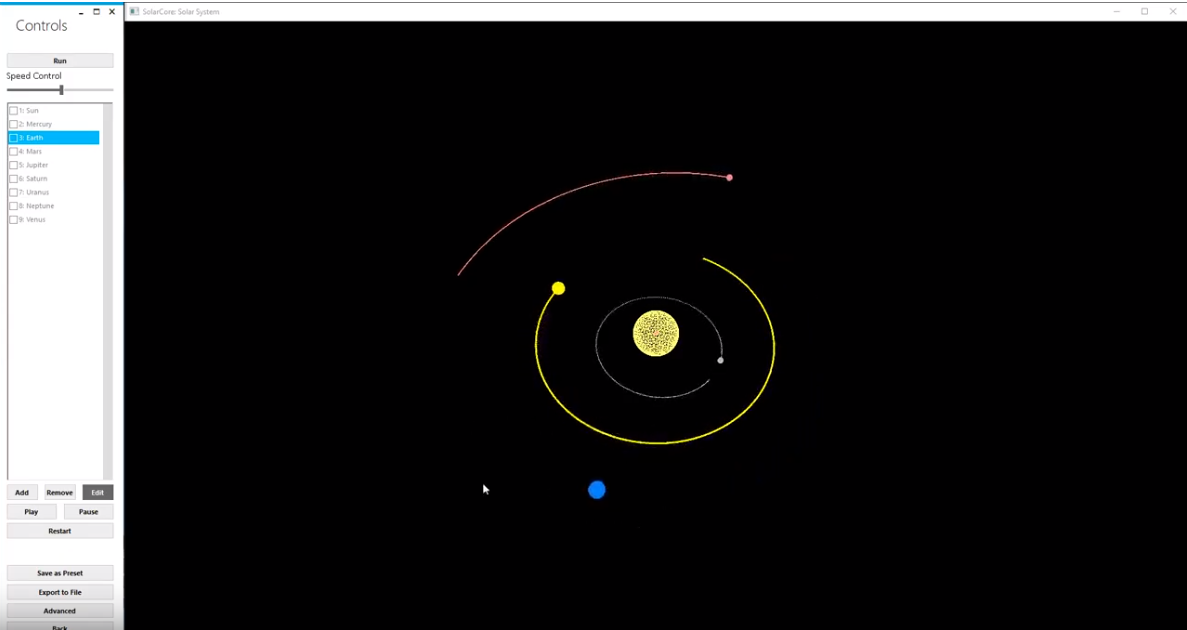
The simulation successfully loaded, and the mouse was used to press the “Restart” button, to restart the simulation.

Test 58 (https://youtu.be/5GI8sxKO1YY):



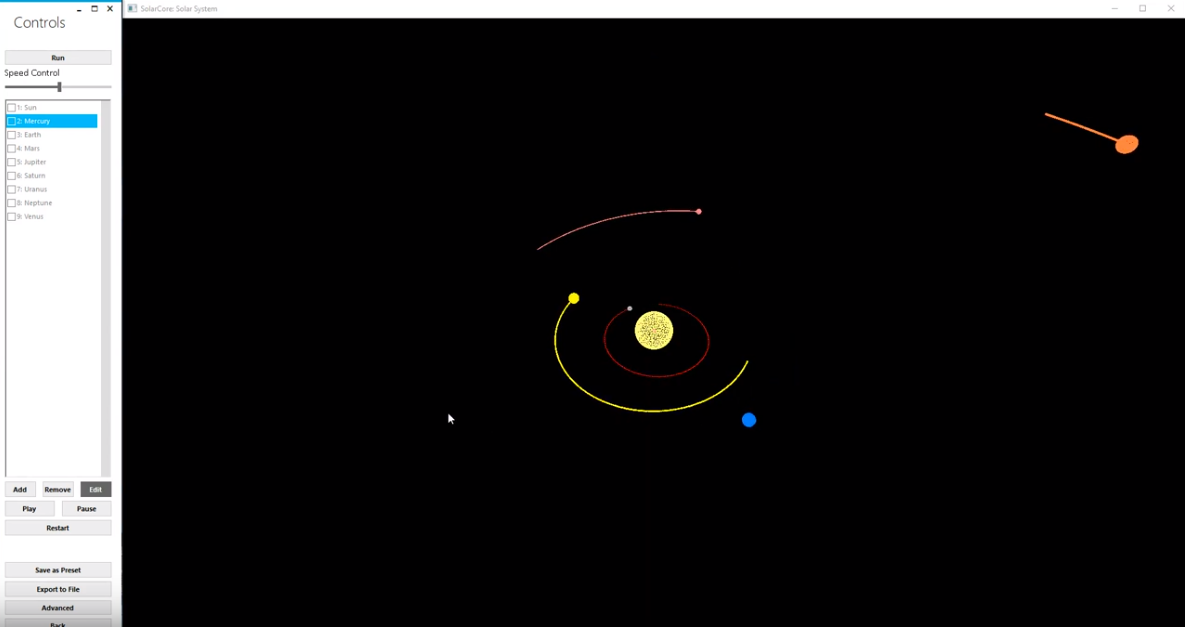
The simulation successfully loaded, and the “R” key was used to restart the simulation.

Tests 59 & 60 (https://youtu.be/Q4JtV-ypBdI):



The simulation successfully loaded, and the trails behind the planets displayed. The Earth’s trail was disabled in the ObjectForm menu for that object, and it no longer displayed the Earth’s trail.

Test 61 (https://youtu.be/oz-B3ddN1vg):



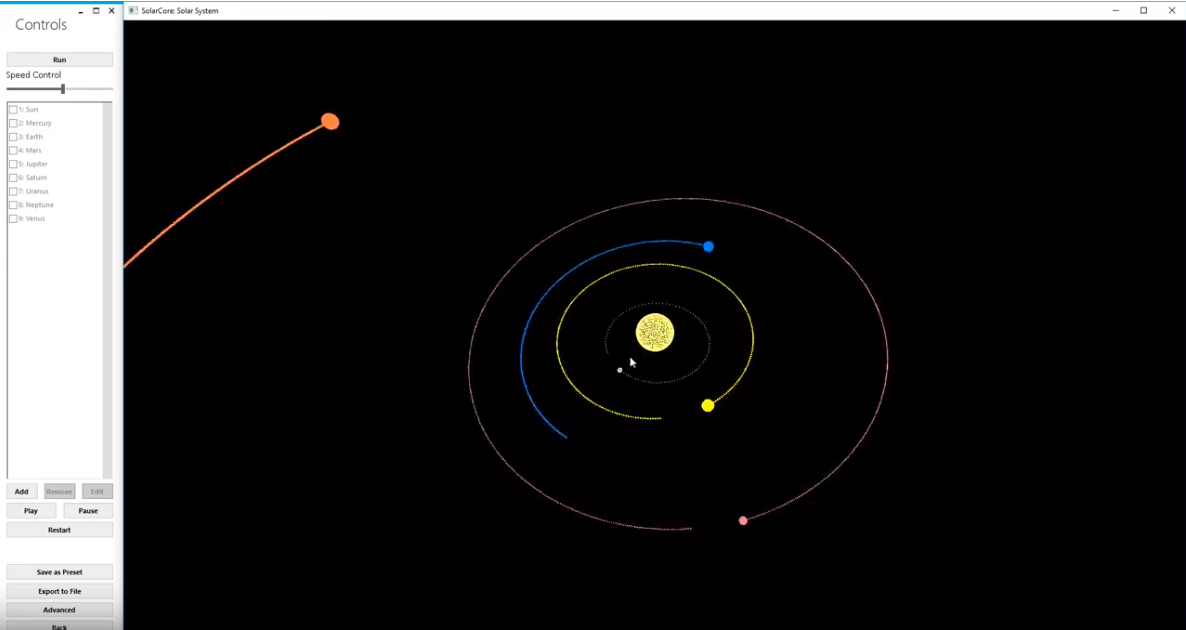
The simulation successfully loaded, and the trails behind the planets displayed. Mercury’s trail colour was changed in the ObjectForm menu for that object, and the colour changed to red in the display window successfully.

Test 62 (https://youtu.be/W6yv8mwcdIQ):



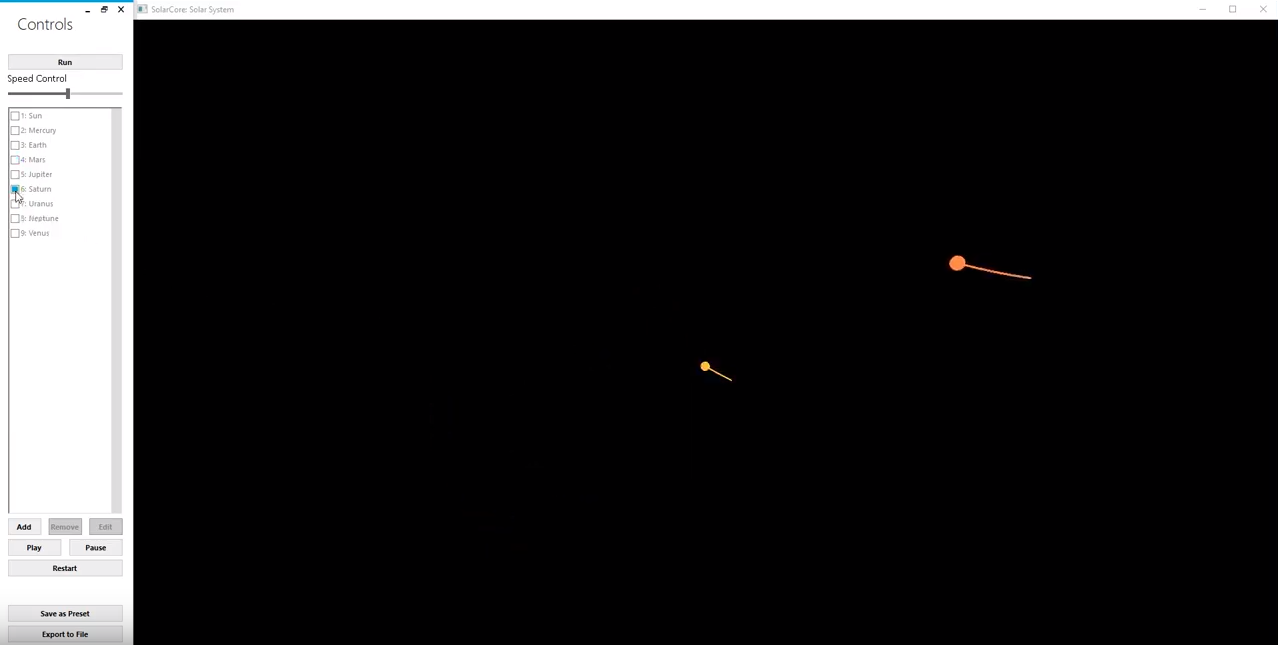
The simulation successfully loaded, and the trails behind the planets displayed. The Earth’s trail length was changed in the ObjectForm menu for that object, and the trail then only swept out half of its orbit around the Sun, as expected.

Test 63 (https://youtu.be/NpFVJgtORhI):



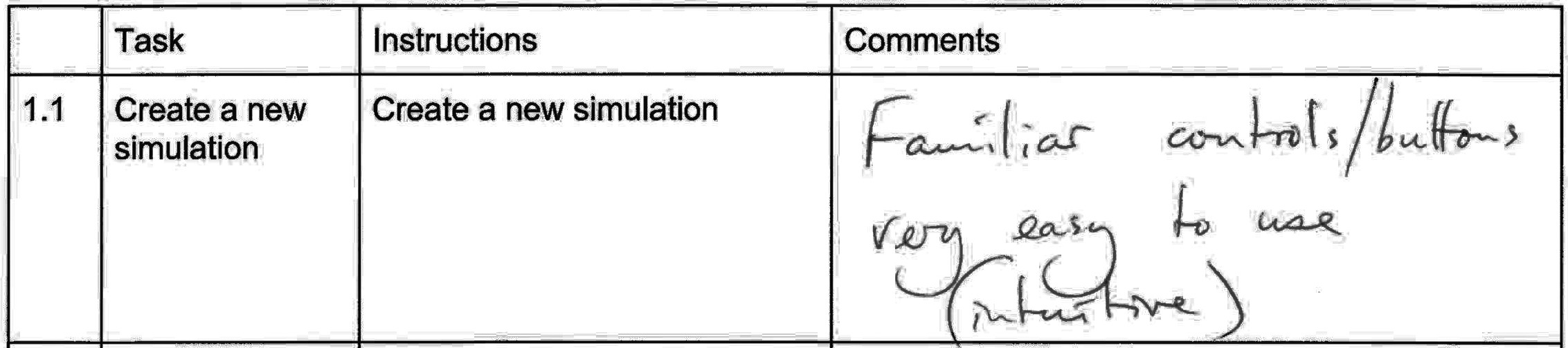
The simulation successfully loaded. The mouse scroll wheel was used, and the simulation zoomed in and out with the focus on the Sun as expected.

Test 64 (https://youtu.be/QhCfP8wVGtY):



The simulation successfully loaded. The ControlForm menu was used to change the focus of the simulation, centering it around different objects within the simulation.

## Black Box Testing

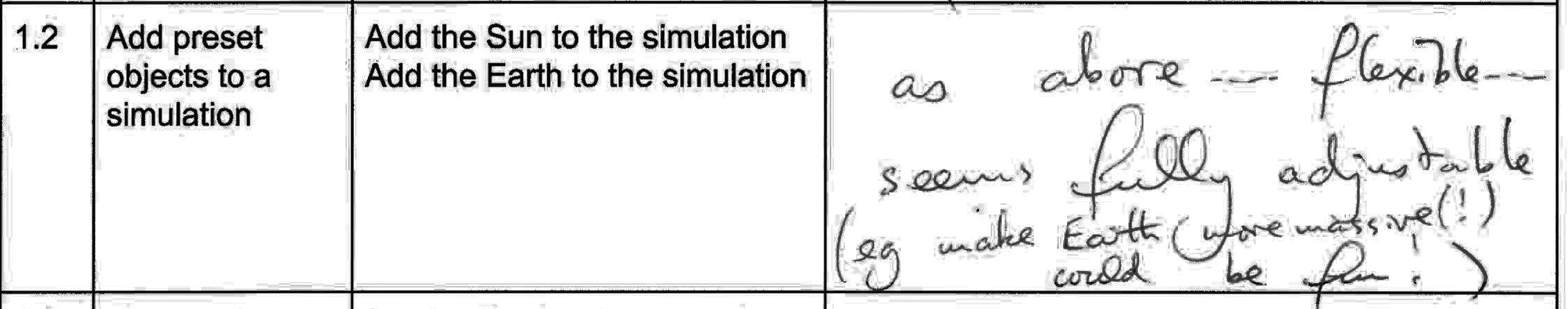


**Comment transcript:**

Familiar controls/buttons.

Very easy to use (intuitive).

**What has been changed as a result:** N/A



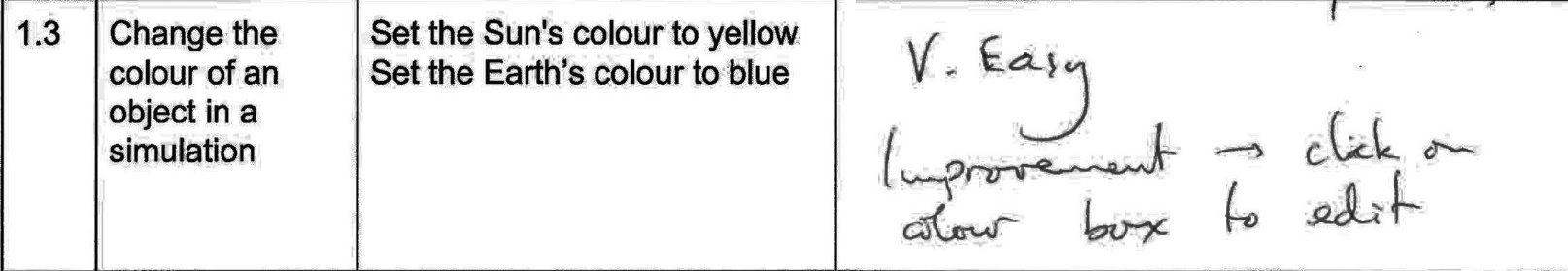
**Comment transcript:**

As above…

Flexible…

Seems fully adjustable (e.g. make Earth more massive(!) could be fun!)

**What has been changed as a result:** N/A

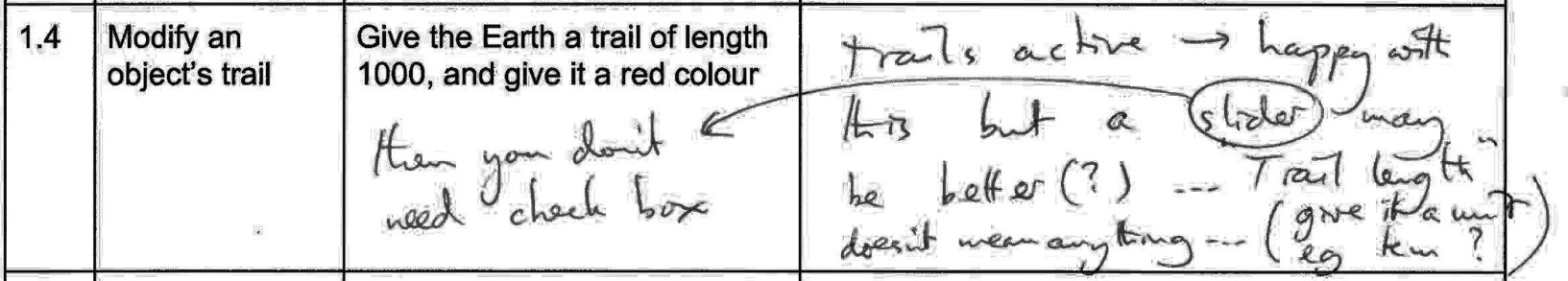


**Comment transcript:**

V. Easy.

Improvement – Click on colour box to edit.

**What has been changed as a result:**



**Comment transcript:**

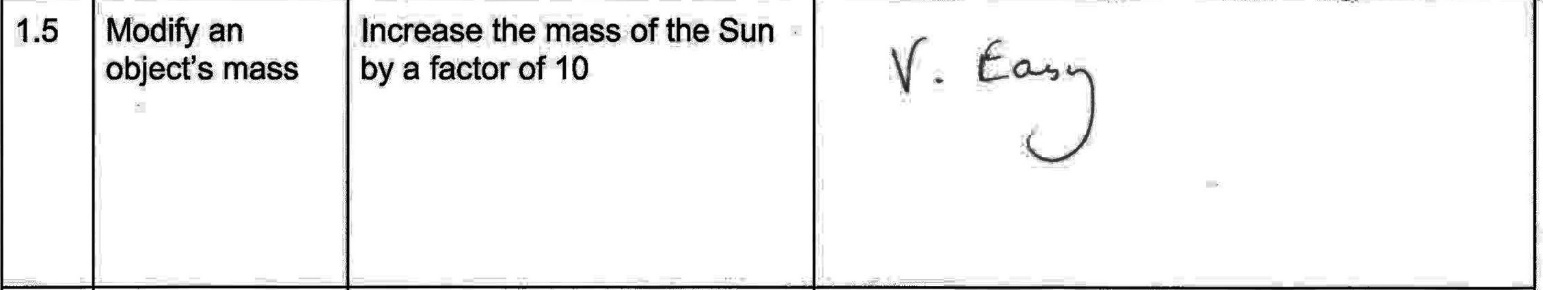
Trails Active:

* Happy with this but a slider may be better(?)
* Then you don’t need a check box.

Trail Length:

* Doesn’t mean anything
* Give it a unit e.g. KM?

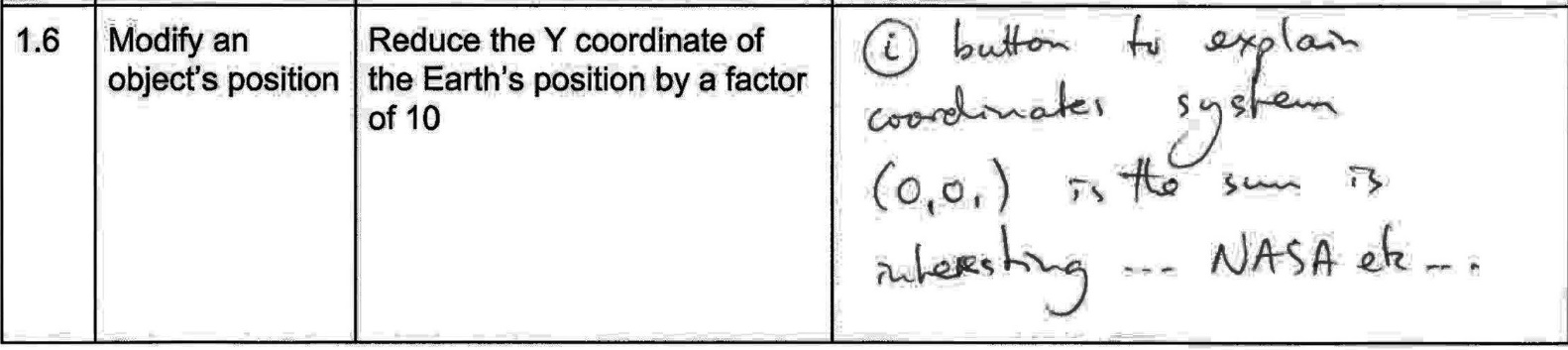
**What has been changed as a result:**



**Comment transcript:**

V. Easy.

**What has been changed as a result:** N/A

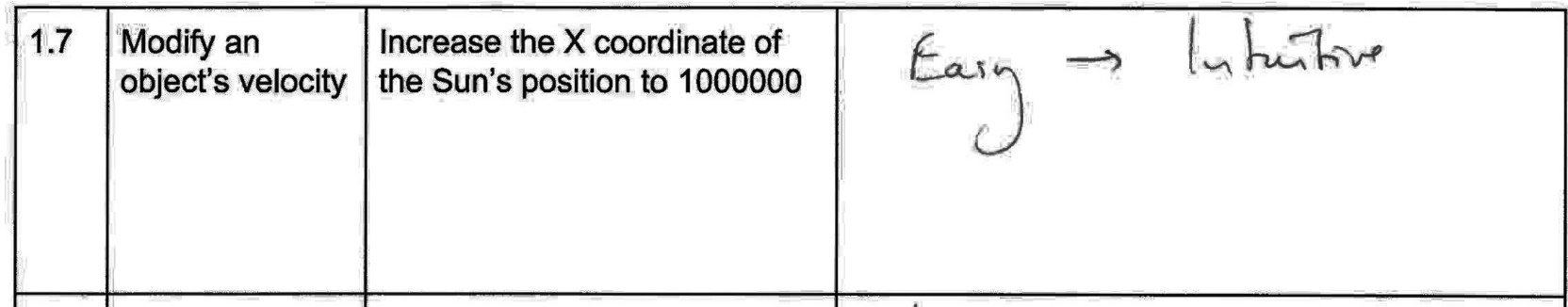


**Comment transcript:**

i button to explain the co-ordinates system

* E.g. Co-ordinates relative to the Sun, which is at (0,0,0)
* Talk about NASA etc.

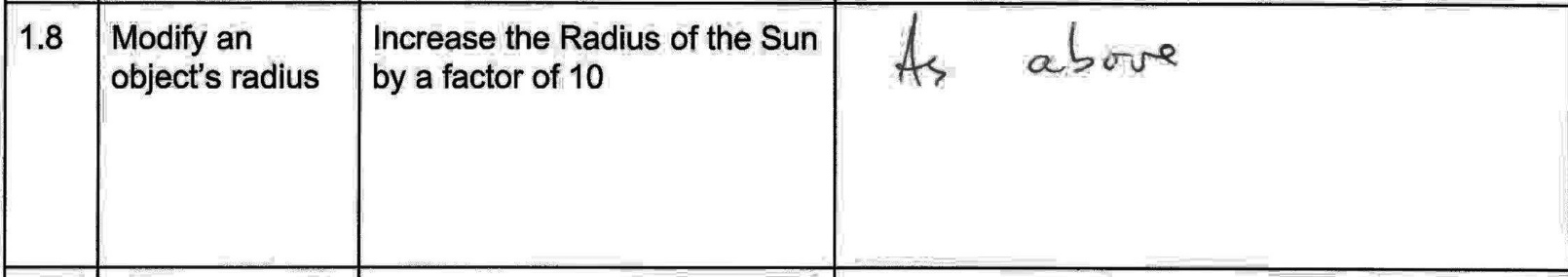
**What has been changed as a result:**



**Comment transcript:**

Easy - Intuitive

**What has been changed as a result:** N/A



**Comment transcript:**

As above

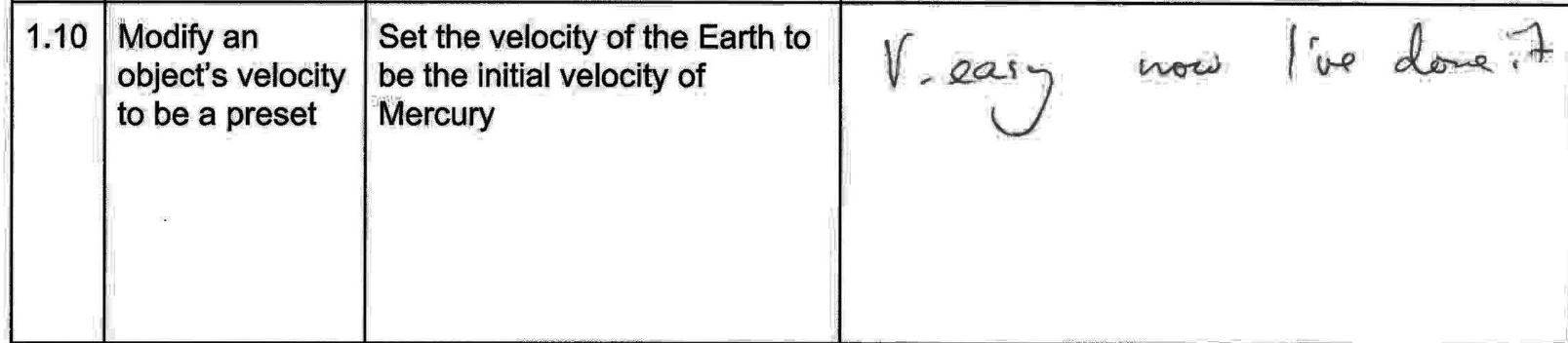
**What has been changed as a result:** N/A



**Comment transcript:**

i button to explain this feature

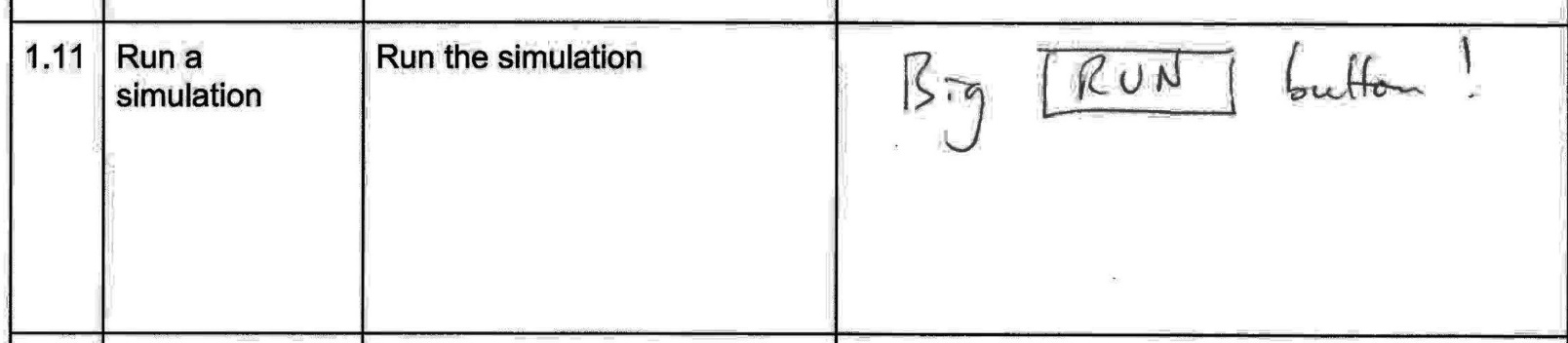
**What has been changed as a result:**



**Comment transcript:**

V. Easy now I’ve done it

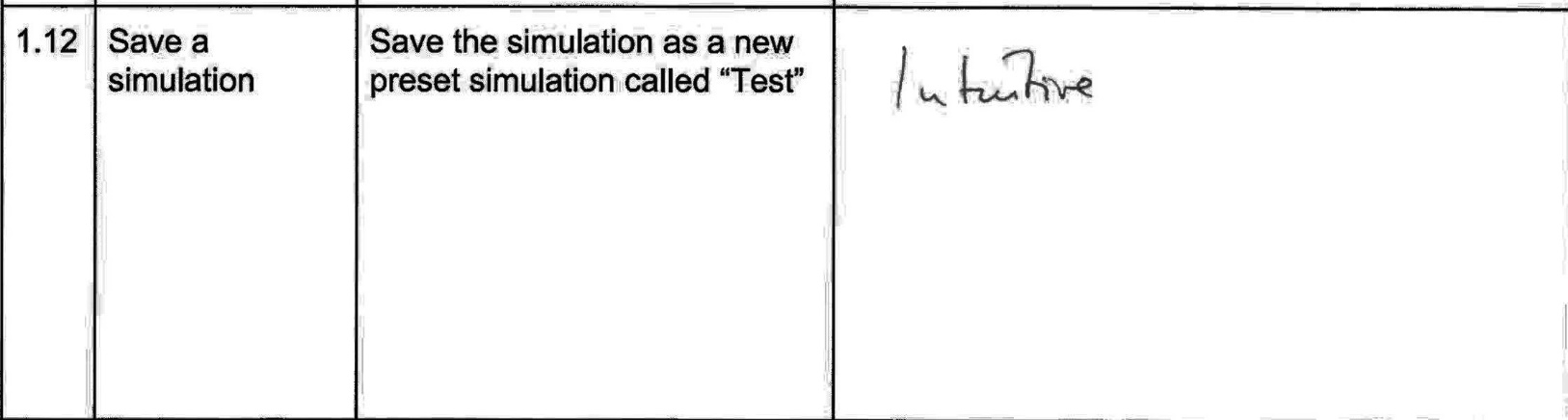
**What has been changed as a result:** N/A



**Comment transcript:**

Big RUN Button!

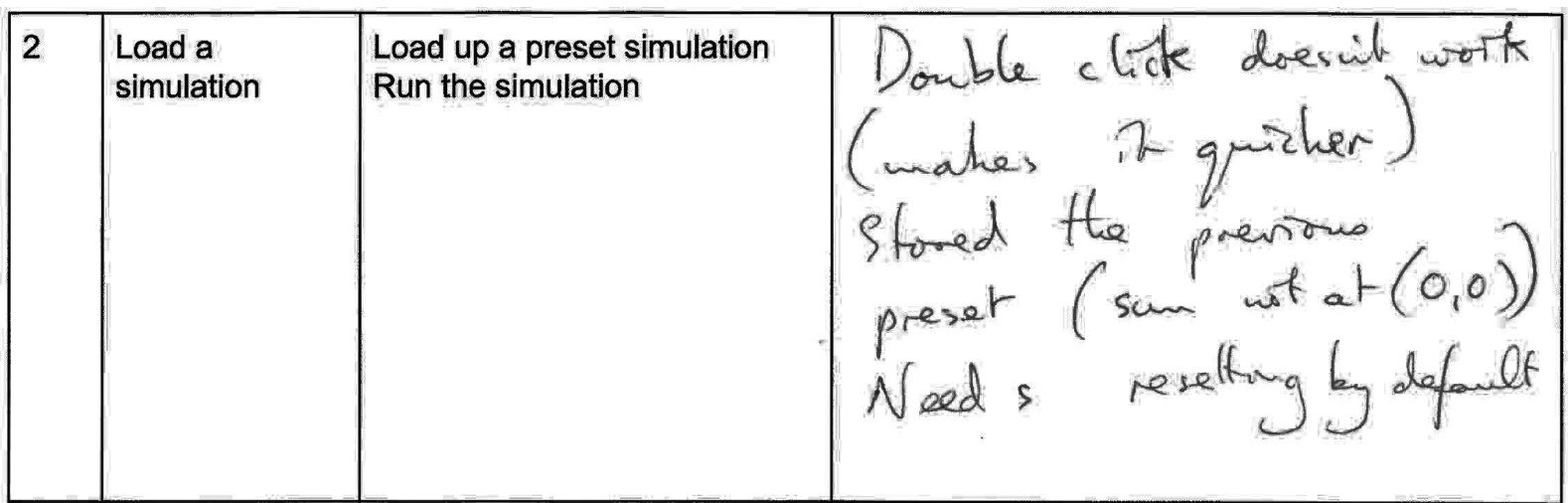
**What has been changed as a result:** N/A



**Comment transcript:**

Intuitive.

**What has been changed as a result:**

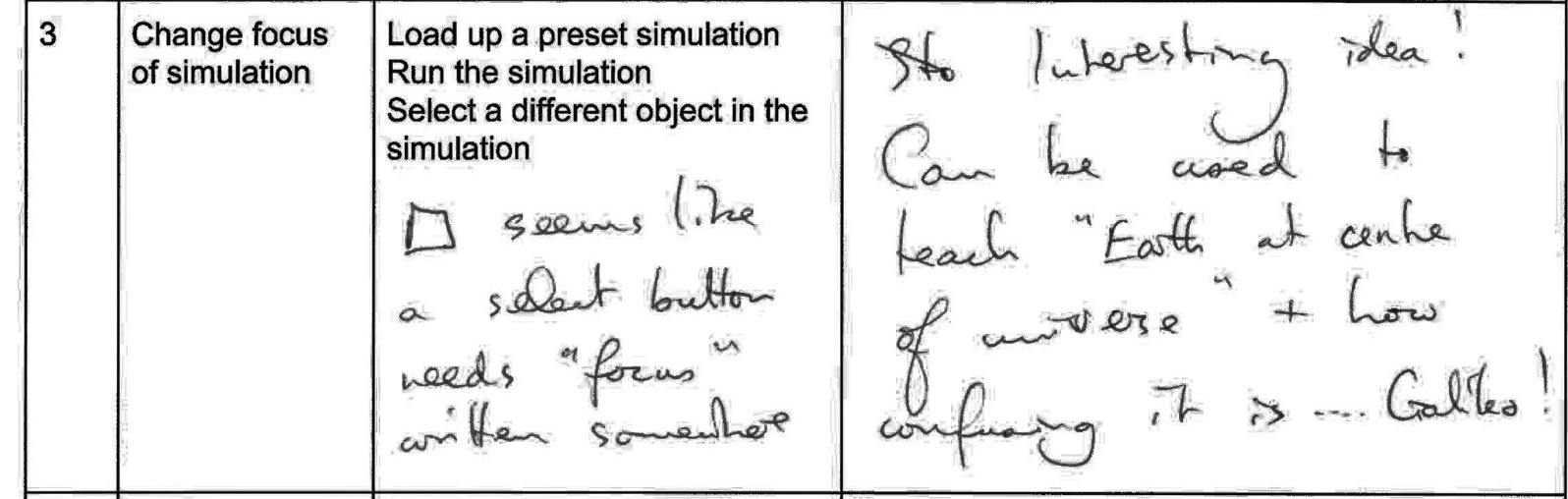


**Comment transcript:**

Double click doesn’t work (makes it quicker)

Stored the previous preset (Sun not at (0,0,0)) – Needs resetting by default

**What has been changed as a result:**



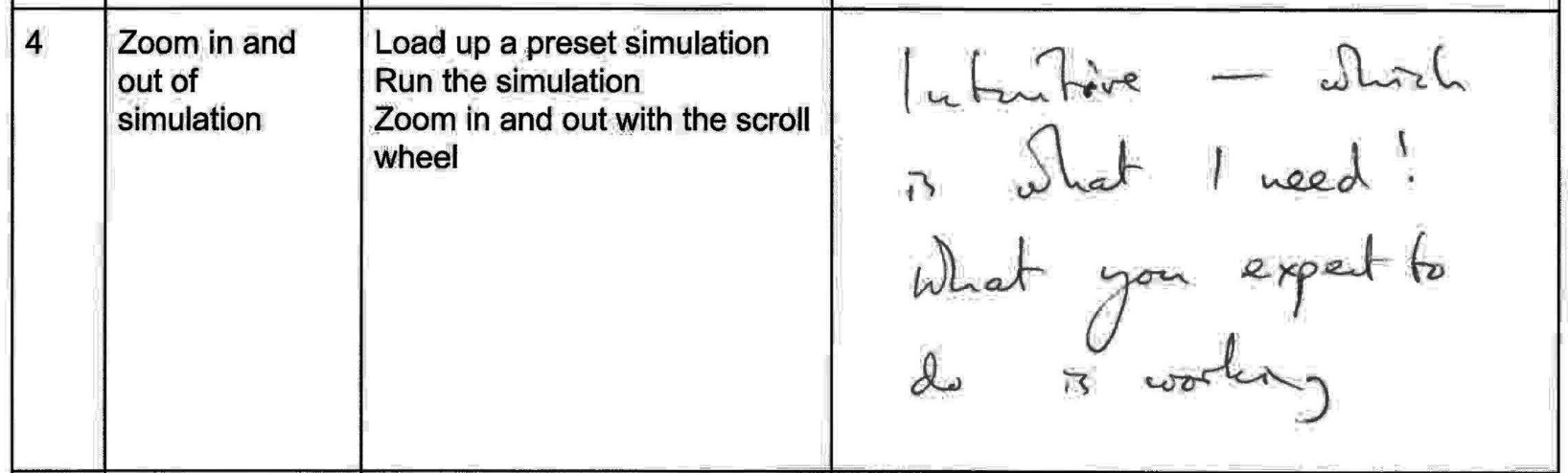
**Comment transcript:**

Interesting idea!

Can be used to teach “Earth at centre of universe” and how confusing it is – Galileo!

Seems like a select button. Needs “focus” written somewhere.

**What has been changed as a result:**

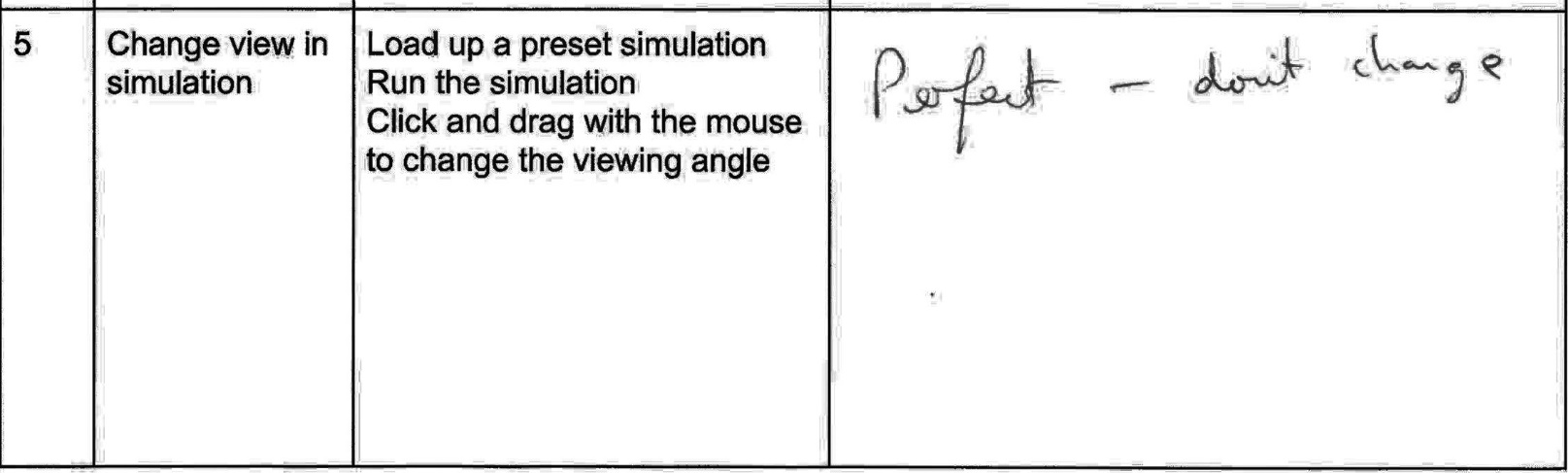


**Comment transcript:**

Intuitive – Which is what I need!

What you expect to do is working.

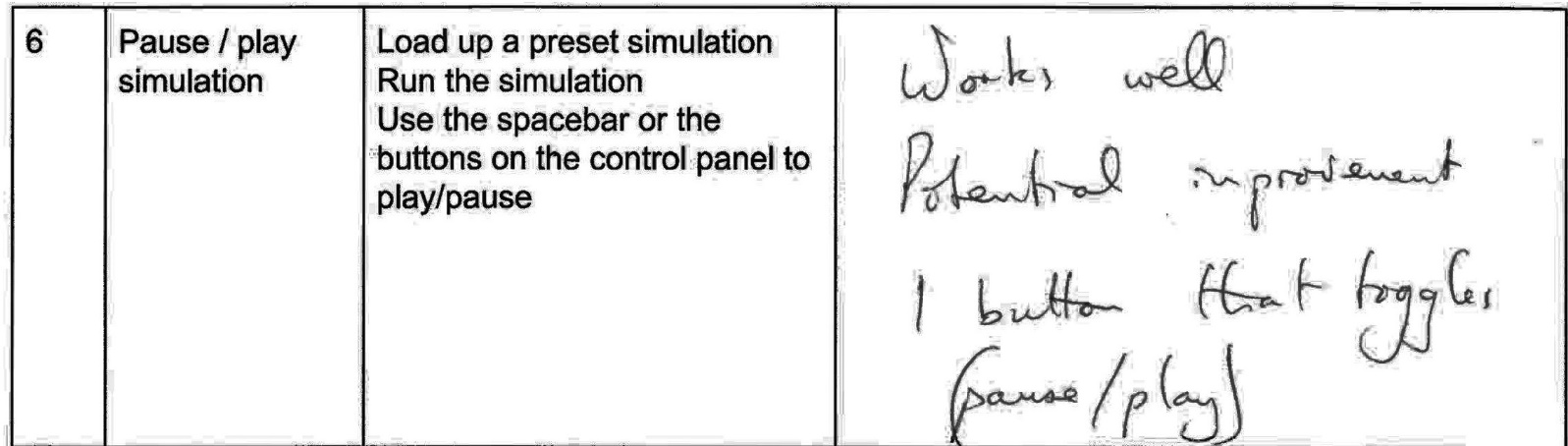
**What has been changed as a result:** N/A



**Comment transcript:**

Perfect – Don’t change

**What has been changed as a result:** N/A

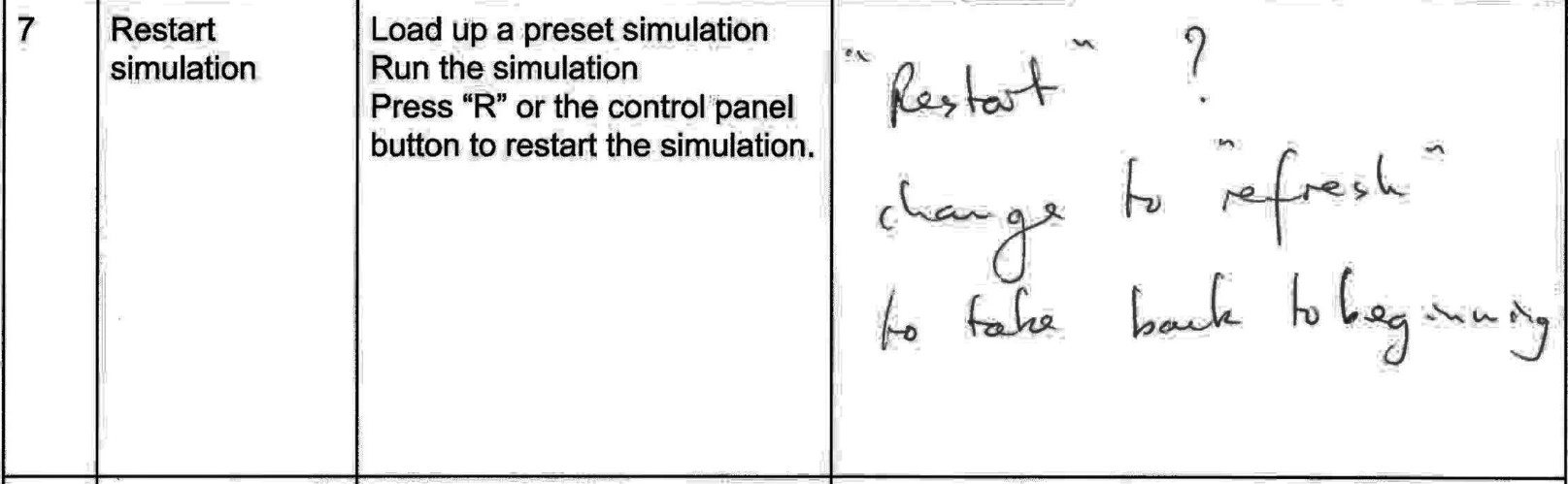


**Comment transcript:**

Works well

Potential improvement: One button that toggles between pause and play

**What has been changed as a result:**

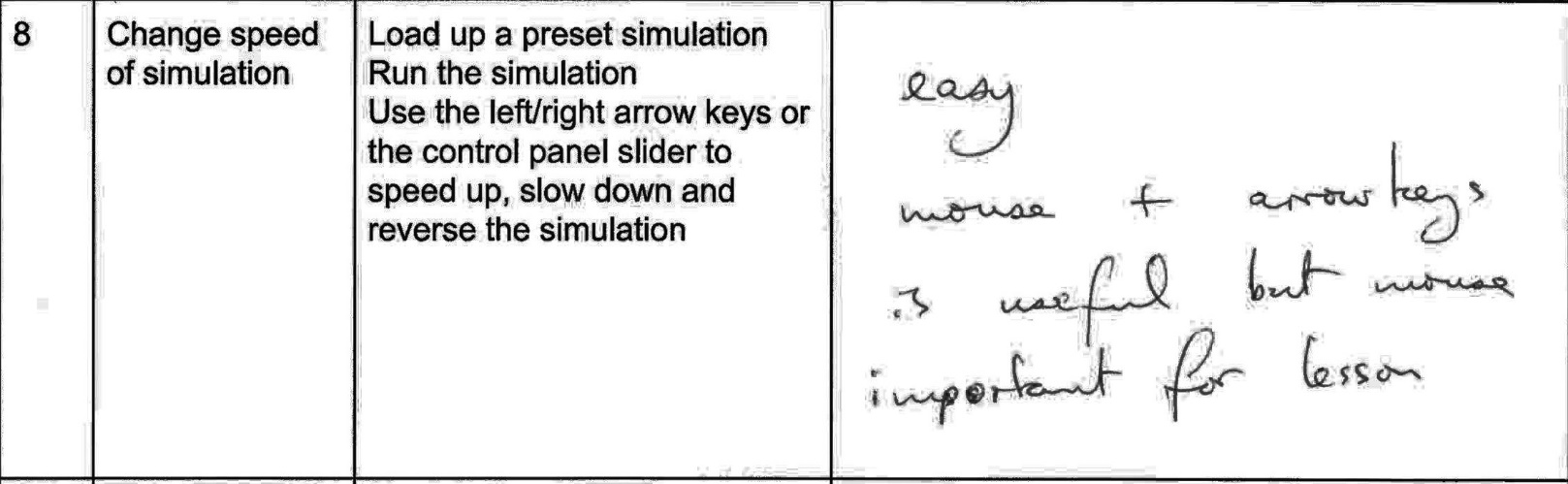


**Comment transcript:**

“Restart”?

Change to “Refresh” to take back to beginning

**What has been changed as a result:**

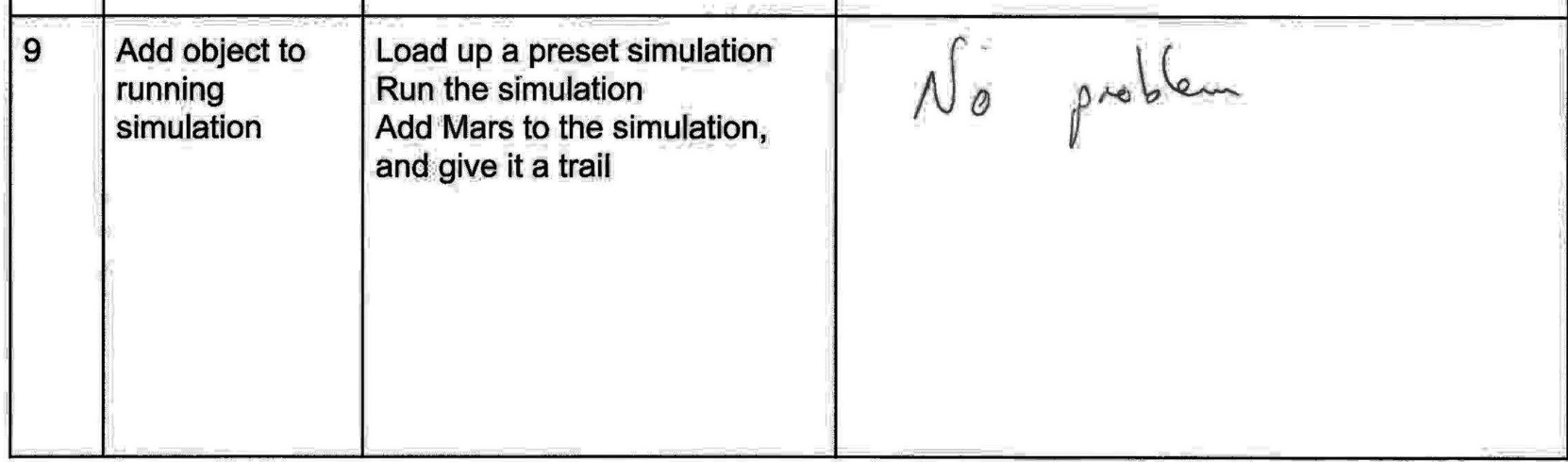


**Comment transcript:**

Easy.

Mouse & Arrow keys is useful but mouse important for lesson.

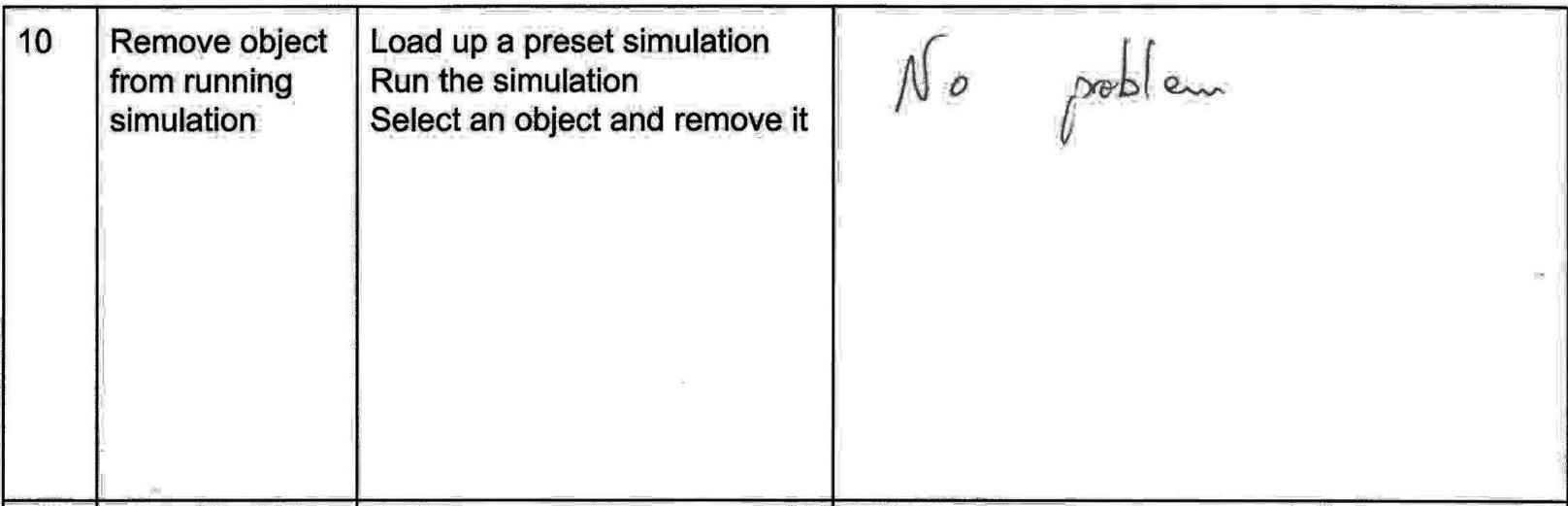
**What has been changed as a result:** N/A



**Comment transcript:**

No problem.

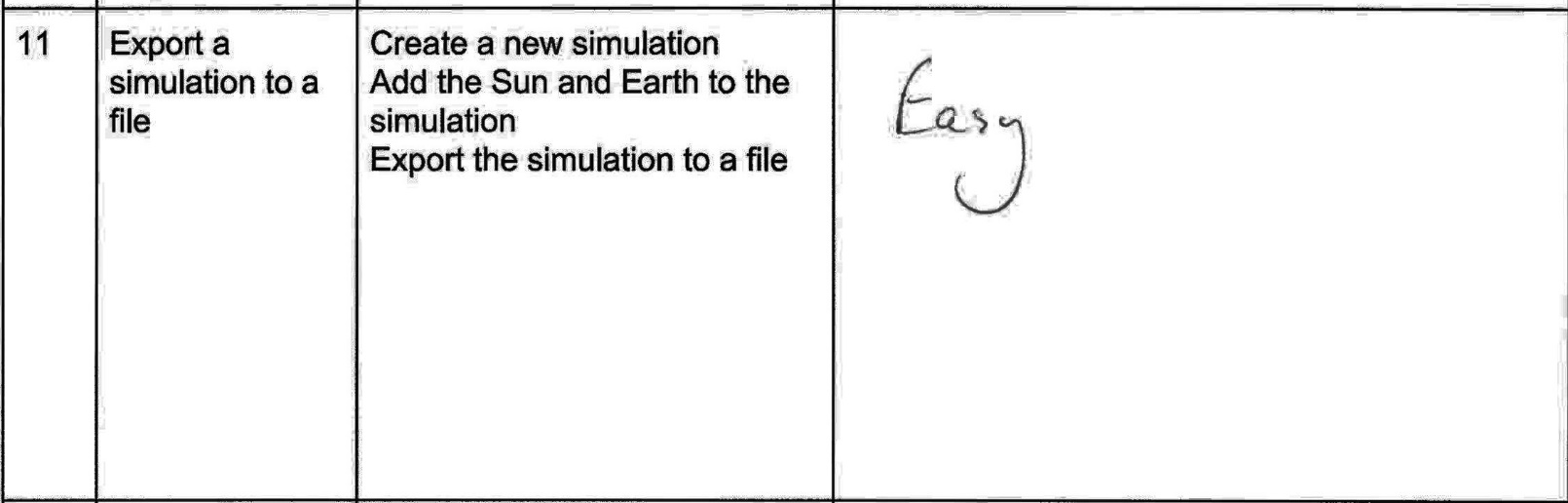
**What has been changed as a result:** N/A



**Comment transcript:**

No problem.

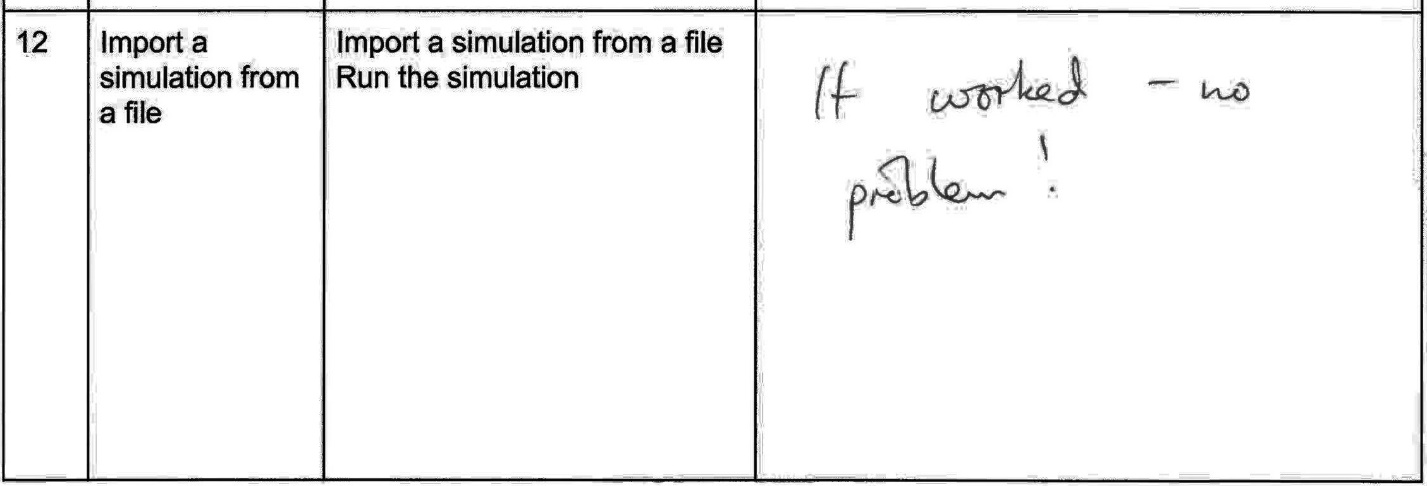
**What has been changed as a result:** N/A



**Comment transcript:**

Easy

**What has been changed as a result:** N/A



**Comment transcript:**

It worked – No problem!

**What has been changed as a result:** N/A

## Questionnaire

