Flat Earth Space Program

User's Manual

Gage Phillips, Jacob Marshall, Zachary Atkins, Thomas Gartman

Manual written by Jacob Marshall and edited by Zachary Atkins

Table of Contents

Introduction

Flat Earth Space Program is a 2-dimensional physics-based sandbox game in which the player can build a rocket using a variety of different components and fly it between different planets and moons. The game has no concrete goals or destinations, but instead allows the player to experiment how they please amongst the vast space environment.

Setup and Installation

Python must be installed in order to run Flat Earth Space Program. Python can be downloaded at https://www.python.org/downloads/. Once python is downloaded, PyMunk and PyGame, 2 python libraries, must be installed as well. In order to install these, simply type:

```
pip install pymunk
```

and

```
pip install pygame
```

into your command line of choice (a command line can be opened by searching for "command prompt" on windows or "terminal" on Mac/Linux). Once these are installed, navigate to the program's directory and type

```
python main.py
```

into the command line. This will launch the game.

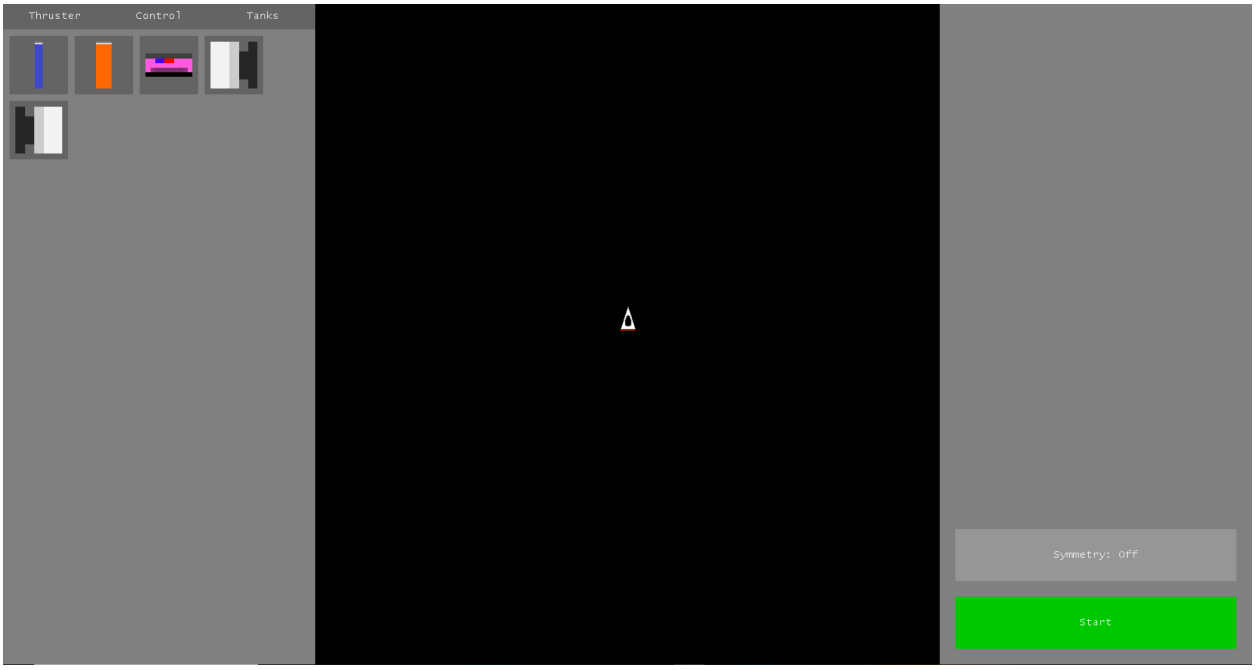
Main Menu

Options

- **Start Demo**: Starts the game with a pre-built rocket (See Spaceflight)
- **Rocket Builder**: Allows the player to build their own rocket prior to starting the game (See Rocket Builder)
- **Test Suite**: Prints the results of the game's test suite to the console
- **Exit to Desktop**: Terminates the program

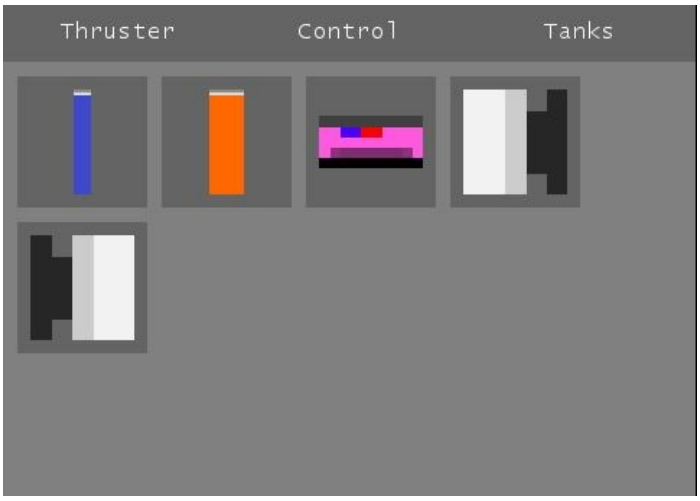
Rocket Builder

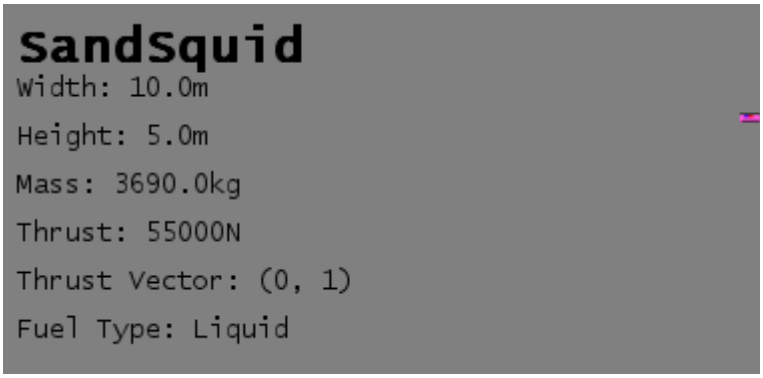
The Rocket Builder allows players to build their own rocket using combinations of different rocket components. Upon entering the rocket builder, players will see a screen similar to the one pictured above.



The player's rocket is initialized with a command module, which can not be moved or deleted. In order to add more components to the rocket, the player can drag and drop components from the different drawers on the left-hand side of the screen. When the mouse is released, if the component being dragged borders/intersects another component, the component will be added to the rocket. If the component does not border the existing rocket then the placement is considered illegal and the component will not be placed. Players may also interact with the symmetry toggle on the bottom-left, which will place a component on the opposite side of the rocket equidistant from the center axis relative to where the player places a component.

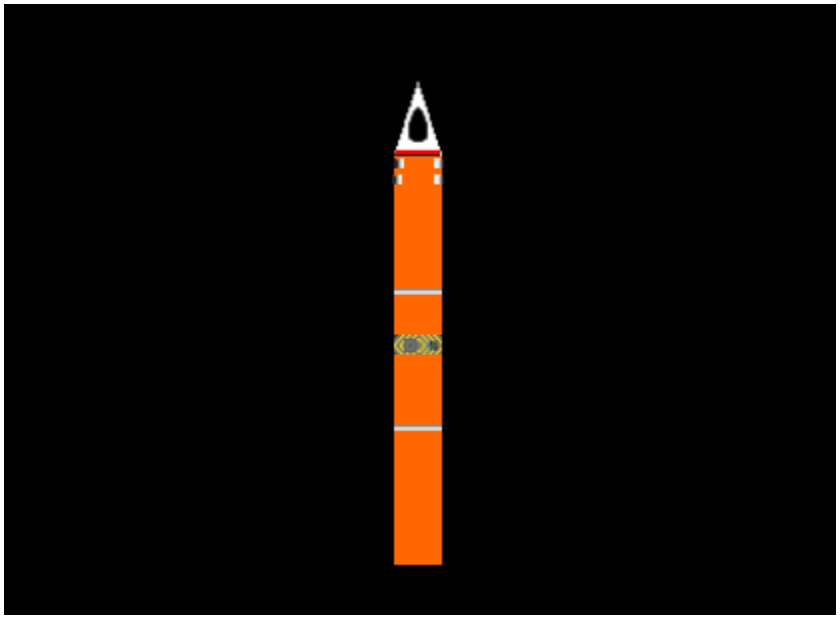
Players may select components from 3 different drawers in the left pane: Thrusters, Control, and Tanks (pictured right).





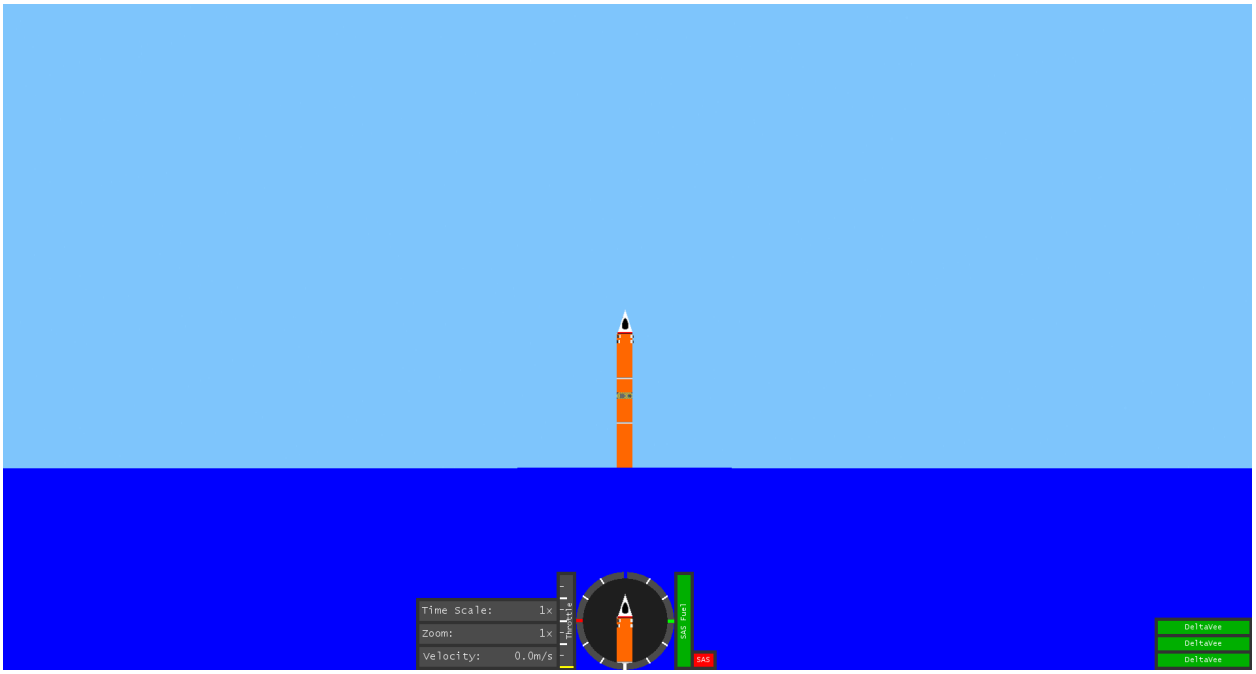
When dragging a component, additional information about the component will be displayed in the right pane (pictured left).

Below is an example of a completed rocket. When the player is done constructing their rocket, they may click the "Start" button which will start spaceflight with the rocket that they have built.



Spaceflight

When spaceflight begins, the player's rocket is placed on the surface of a planet. At this point the player has full control over the behavior of the rocket.

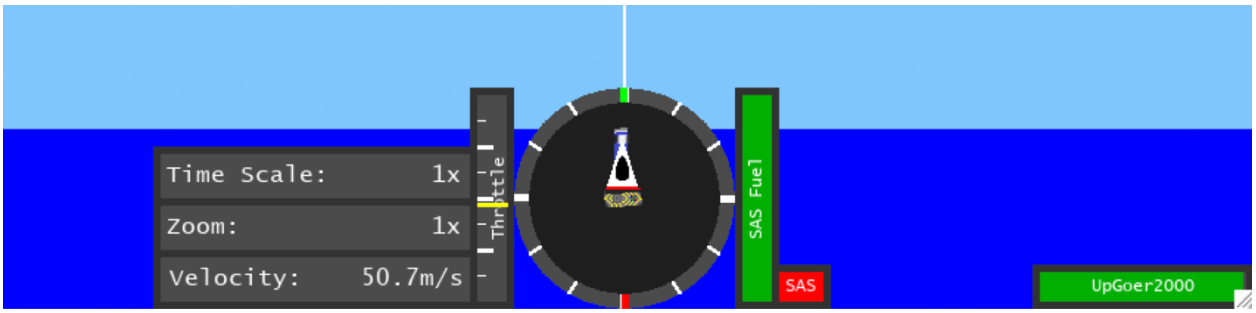


Controls

- Left SHIFT - Increase Thrust Throttle
- Left CTRL - Decrease Thrust Throttle
- Z - Full Throttle
- X - Cut Thrust
- A - Rotate Counter-Clockwise
- D - Rotate Clockwise
- V - Enable SAS, locking rocket to the current angle of rotation.
- = - Increases Timescale (speeds up time)
- - - Decreases Timescale (slows down time)

The HUD

A variety of information is displayed to the player in the in-game HUD, pictured in detail below, including Time Scale, Zoom, Velocity, Throttle, as well as a meter showing how much fuel is left in each thruster. Non-SAS Thrusters are shown in the lower-left corner, and a Navigation Circle is shown in the center, which can be useful for viewing the orientation of the rocket when zoomed out to the point where the rocket is no longer visible.



Trajectory Lines

These lines show the future path of the rocket (assuming no other forces act upon it) and will become visible as soon as a non zero velocity is achieved. These can be useful for navigating over large distances and timescales.