**Kepler’s Laws Simulator**

Samuel Barr

Joshua Shields

**Overview**

Kepler’s laws define how bodies in our solar system move, and can be used to find and predict their positions. This is the subject of a laboratory project in ASUs astronomy department. They are currently using a very dated simulator that the lab director would like updated. It has to be run through DOSBox, and has no mouse support. The current interface requires constant restarting, and back tracking through menus. Working with the current problem adds extra time to the lab, and requires the instructor to spend more time explain how to use the simulator than discussing the labs subject. We are proposing an updated replacement that retains all of the needed functionality, while streamlining operation, and updating the GUI and graphics.

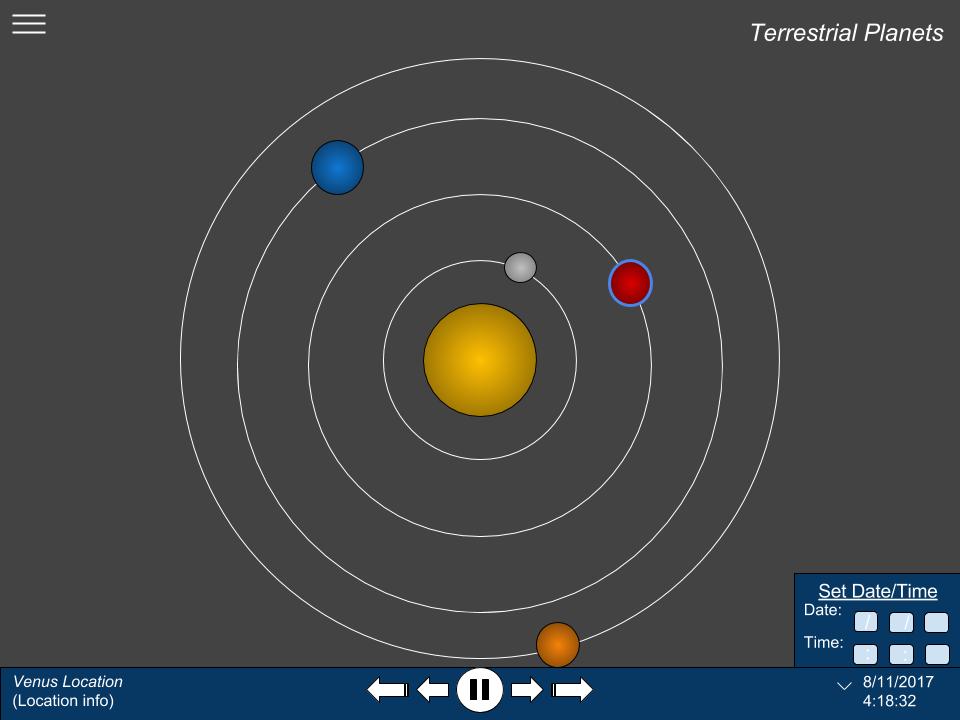


Figure 1

Concept image of the simulator focused on Venus.

We also want to add a modern menu and options to make the simulator more intuitive for users. This should also make it unnecessary to require back tracking into and out of the simulation.

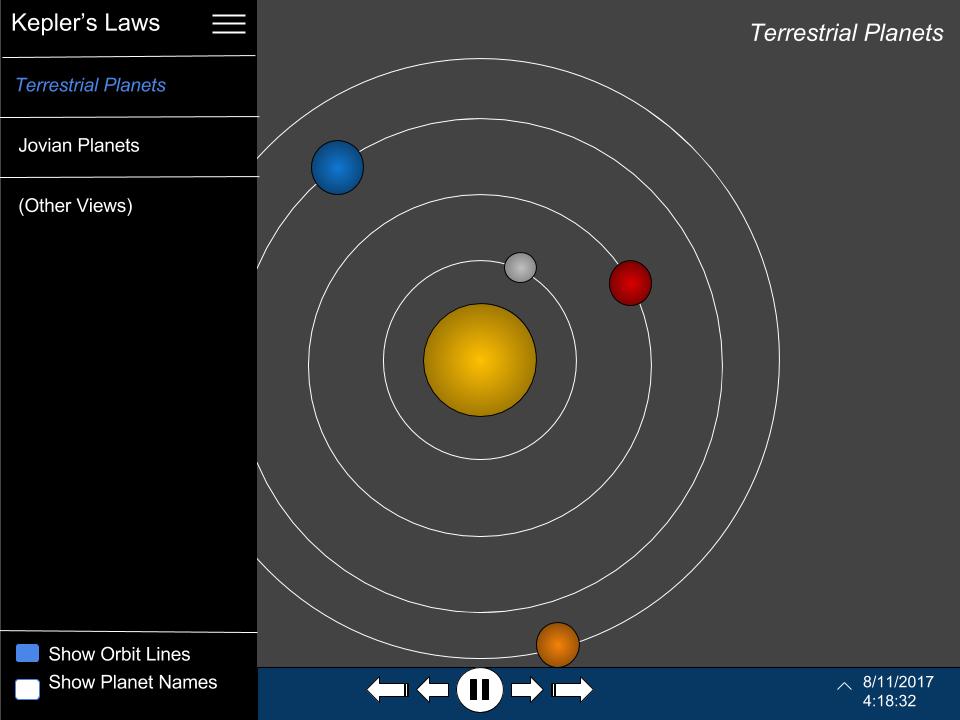


Figure 2

Concept image of the simulators menu.

**Features**

Features are being listed from most to least critical.

* A scale display of the solar system accurately depicting the locations of celestial bodies relative to each other.
* Being able to accurately move the display backwards and forwards in time, displaying both the Gregorian and Julian dates.
* Having an option that can toggle on and off an overlay connecting bodies to each other through lines.
* The ability to focus the view on a particular object.
* A field that allows the user to input their own custom time increment to move forward or backward through the simulation.
* An easy installation wizard.
* An option to free zoom the view with a slider.
* Custom icons
* The ability to enter new objects into the simulation at run time.

**Similar Existing Programs**

Kepler’s Laws

This is the program we are replacing.

Solar System 3D Simulator

A more graphically advanced program that what we are proposing, but lacking the specific tools needed to complete the lab project.

<http://download.cnet.com/Solar-System-3D-Simulator/3000-2054_4-10477538.html>

Celestia

This program is much more ambitious than ours. A full 3D simulation that aims to simulate not only our solar system, but also deep space extra solar objects. Again, this program, while very graphically impressive, still does not display the data the lab requires.

<https://celestia.space/index.html>

**Experience**

Our team has some experience with graphical applications through various school projects, which should transfer to this program directly. In addition, we have both used the program we are replacing, and are familiar with its features.

**Technology**

Our team is most confidant proceeding with the project in Java. We are familiar with its graphical capabilities, and syntax, in addition to simply having the most experience with the language. We plan on testing through the use of automated inputs which we can compare to expected results. Code Style will be performed by the IDE, personally Atom’s ‘Beautify’.

**Risks**

This is the largest project that either of us has ever worked on from start to finish. Neither of us are experienced with auto-style checking. We are both rusty with our graphical elements, and this project requires very accurate graphics.

**End-Users**

The professors and students in the Physics and Astronomy department, for the purpose of learning about Kepler’s laws through observing the simulation.