#257.06

Introductory Astronomy Interactives developed with Undergraduates

Juan Cabanela,
Andrew Louwagie Gordon,
& Sam Holen
(Minnesota State University Moorhead)

Abstract

Many introductory astronomy service courses incorporate labs or other interactive components which use web-based activities. Much of the currently available software, either from textbook pubishers or astronoy educators such as the Nebraska Astronomy Applet Project, was written using *Adobe Flash*. Adobe Systems is dropping support for Flash at the end of 2020. This problem hit our service courses particularly hard, with approximately half of our lab activities requiring updates. Faced with this challenge, we exploited the fact that our department has incorporated *Python* programming into our curriculum for physics majors to come up with a solution. For 10 weeks in Summer 2018, two undergraduates collaborated with a professor to develop 16 replacement web-based activities for these labs. The interactives were written in *Python* running in *Jupyter* notebooks and have been made available as open source software. We deployed the interactives to students using *The Littlest JupyterHub* server. Students simply log into the server and the interactives are executed automatically. We are presenting our interactives as well as a discussion of what we learned to help make this collaboration so productive.

Motivation

Large introductory astronomy courses are more and more often being taught online. As science courses move online, instructors face the challenge of how to offer these online students lab and lab-like experiences (de Jong, Linn, and Zacharia 2013, Waldrop, 2013). Online interactive applications simulating physical and astronomical systems are one means of providing some of these experiences.

Interactive software in the form of visualizations or simulations of physical and astronomical systems has been used in introductory physics and astronomy classrooms since the advent of the personal computer (e.g. *Interactive Physics* by Design Simulation Technologies or *Starry Night* by Simulation Curriculum). By early 2000s, the development of the Java and Flash programming languages allowed for the development of interactive software run inside a web browser. Shortly thereafter, interactive web applications simulating astronomical and physical systems started appearing for use in the classroom, including the popular **PhET** (https://phet.colorado.edu) and **NAAP** (http://astro.unl.edu/naap/) software.

However, both *Java* and *Flash* became associated with computer security issues and by 2017 both *Java* and *Flash* in the web-browser had been deprecated with all support for *Flash* ending in 2020. Today, interactivity for web browsers is typically provided by a combination of *HTML5*, *JavaScript*, and *CSS*.

As a result of the upcoming shutdown of Adobe *Flash*, many of the simulations our introductory astronomy labs at MSUM have been will no longer function in the near future. As such, in the Summer of 2018 a collaboration of two Physics majors working with a faculty mentor proposed to develop new interactive software for use in the introductory astronomy classroom with an eye toward future use in online lab experiences for our online astronomy courses.

de Jong T, Linn MC, Zacharia ZC 2013, Science, **340**, 305
Waldrop MM 2013, Nature, **499**, 268

If you have any questions, feel free to contact:
Juan Cabanela
cabanela@mnstate.edu
Twitter: *@Juan_Kinda_Guy*

**Minnesota State University Moorhead
Department of Physics and Astronomy
1104 7th Ave South, Hagen 307B
Moorhead MN 56563
USA**

Physics Majors as Programmers

The American Association of Physics Teachers urges that every physics and astronomy department provide its majors and potential majors with appropriate instruction in computational physics.
- AAPT UCTF Computational Physics Report (2016)

Our department has had *Python* programming as part of our curriculum since 2013, using *Matter and Interactions* (Sherwood and Chabay, 1999) as our introductory two-semester Physics sequence and developing labs using modelling with *Python* to go with it. Shortly thereafter, we switched our second-year Computational Physics course which had been a mish-mash of *Maple* and *IDL* programming to a *Python*-based curriculum in 2015. Today, our upper-division physics courses regularly include computational problems as part of the course work.

With our students as a resource, in Spring 2018, Cabanela decided to re-implement the interactive software used in our introductory astronomy courses using *Python* running in *Jupyter* notebooks in a web browser. This would allow the development of web-based apps by students who understood the basics of the physics and astronomy we needed to convey.

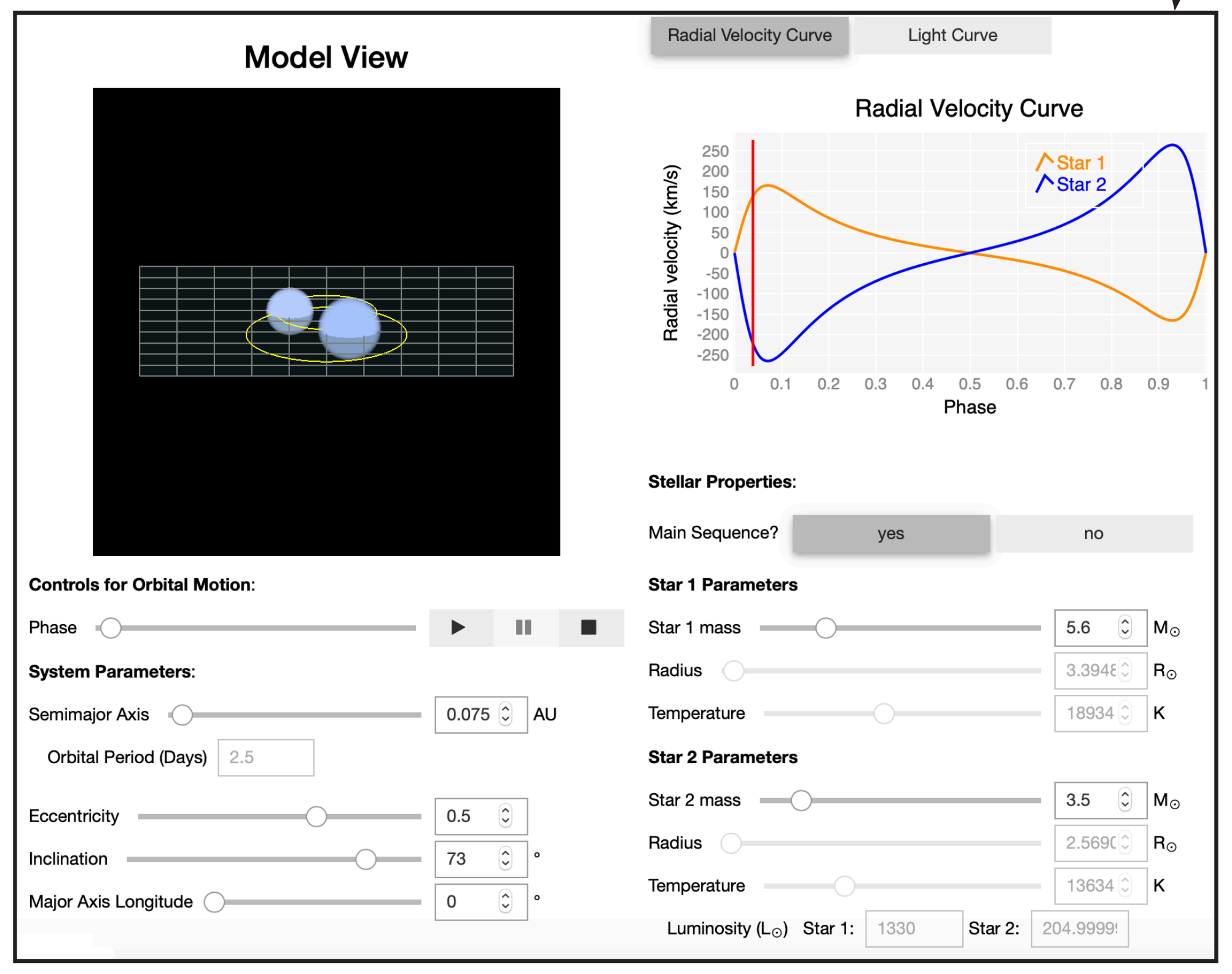
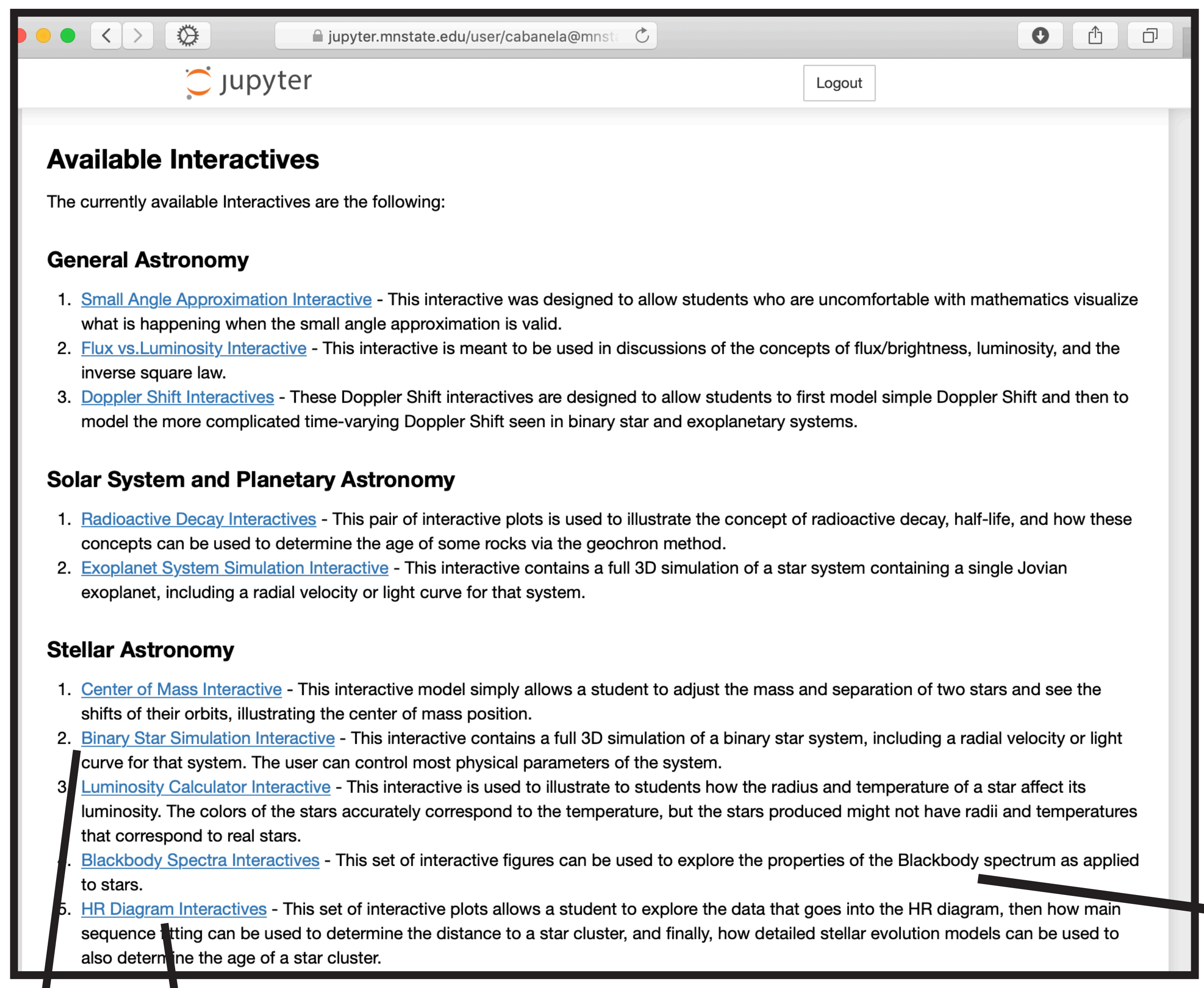


Figure 2: The Binary Star simulation allows students to model radial velocity and light curves for binaries.

A Successful Approach

While both of the student programmers had *Python* programming experience, they had not used *Jupyter* or developed graphical user interfaces (GUIs) prior to this experience. We started the summer with a 1-day *ipywidgets* workshop. This introduced the students to the packages allowing development of GUI interfaces within *Jupyter* notebooks. Immediately after that, the collaboration worked on developing initial versions of the "interactives" (*i.e.* Interactive software running in the web-browser). We used a shared online lab notebook and weekly meetings to iron out issues with the software and come up with the short term goals.

This approach proved very productive. Our intention had been to complete 10 interactives by the end of the summer. Instead, we completed beta versions of 16 interactives by the end of the summer! This suggests the *Python* programming we teach our majors makes them capable developers for these sorts of interactives. Other opportunities exist for these sorts of interactives within our department and elsewhere and we are already looking at replacing some software written in *MATLAB* for our Physics of Music intro course with *JupyterLab*-hosted activities instead.



Available Interactives

The currently available Interactives are the following:

General Astronomy

1. Small Angle Approximation Interactive - This interactive was designed to allow students who are uncomfortable with mathematics visualize what is happening when the small angle approximation is valid.
2. Flux vs.Luminosity Interactive - This interactive is meant to be used in discussions of the concepts of flux/brightness, luminosity, and the inverse square law.
3. Doppler Shift Interactives - These Doppler Shift interactives are designed to allow students to first model simple Doppler Shift and then to model the more complicated time-varying Doppler Shift seen in binary star and exoplanetary systems.

Solar System and Planetary Astronomy

1. Radioactive Decay Interactives - This pair of interactive plots is used to illustrate the concept of radioactive decay, half-life, and how these concepts can be used to determine the age of some rocks via the geochron method.
2. Exoplanet System Simulation Interactive - This interactive contains a full 3D simulation of a star system containing a single Jovian exoplanet, including a radial velocity or light curve for that system.

Stellar Astronomy

1. Center of Mass Interactive - This interactive model simply allows a student to adjust the mass and separation of two stars and see the shifts of their orbits, illustrating the center of mass position.
2. Binary Star Simulation Interactive - This interactive contains a full 3D simulation of a binary star system, including a radial velocity or light curve for that system. The user can control most physical parameters of the system.
3. Luminosity Calculator Interactive - This interactive is used to illustrate to students how the radius and temperature of a star affect its luminosity. The colors of the stars accurately correspond to the temperature, but the stars produced might not have radii and temperatures that correspond to real stars.
4. Blackbody Spectra Interactives - This set of interactive figures can be used to explore the properties of the Blackbody spectrum as applied to stars.
5. HR Diagram Interactives - This set of interactive plots allows a student to explore the data that goes into the HR diagram, then how main sequence fitting can be used to determine the distance to a star cluster, and finally, how detailed stellar evolution models can be used to also determine the age of a star cluster.

Figure 1: The index page that comes up when students log in to the *JupyterHub*. They select their interactives from the list.

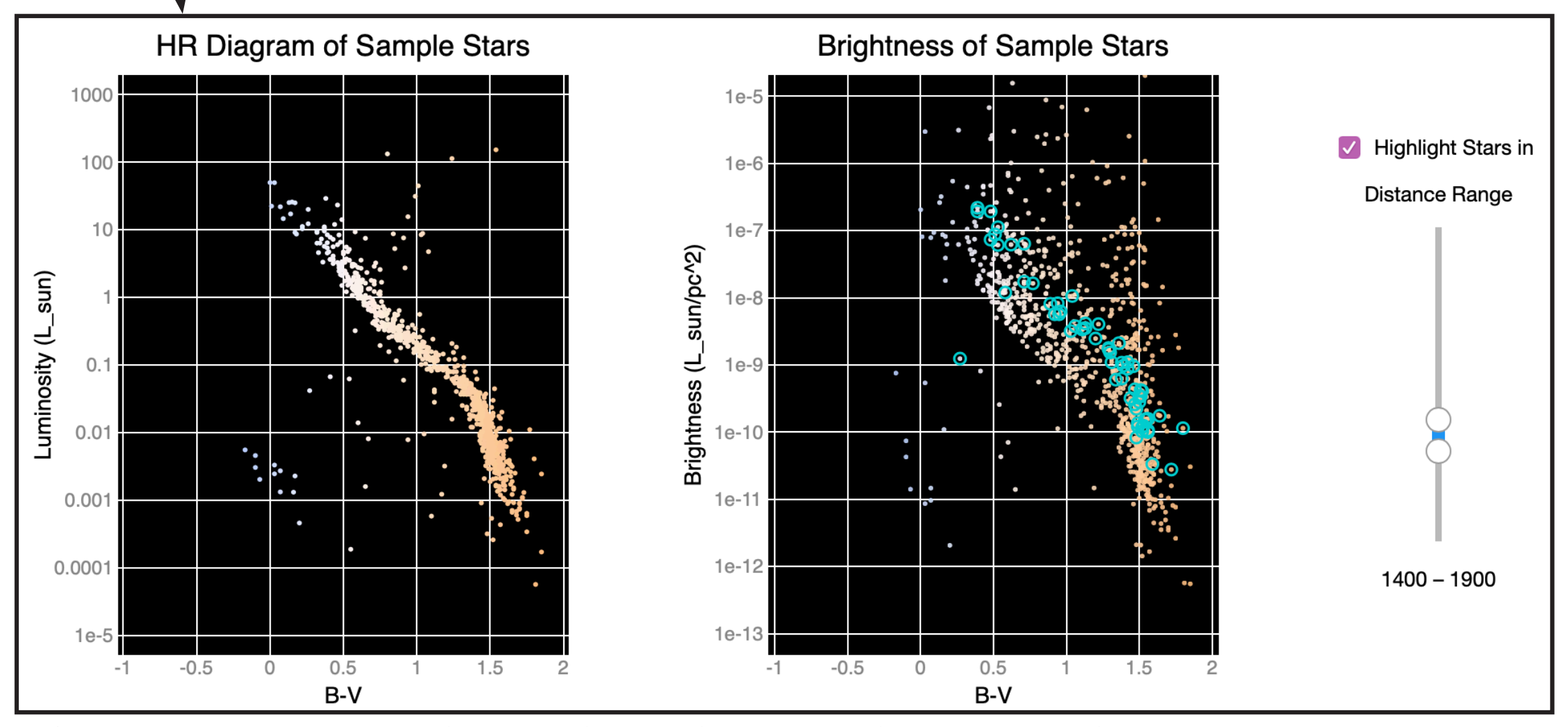


Figure 3(a): When they begin the HR diagram interactive, students explore the relationship between an HR diagram and the corresponding brightness-color diagram. Here, the student has selected stars between 1400 and 1900 pc away, and they are highlight in the brightness diagram, illustrating how the main sequence of a cluster is recognizable in a brightness-color plot.

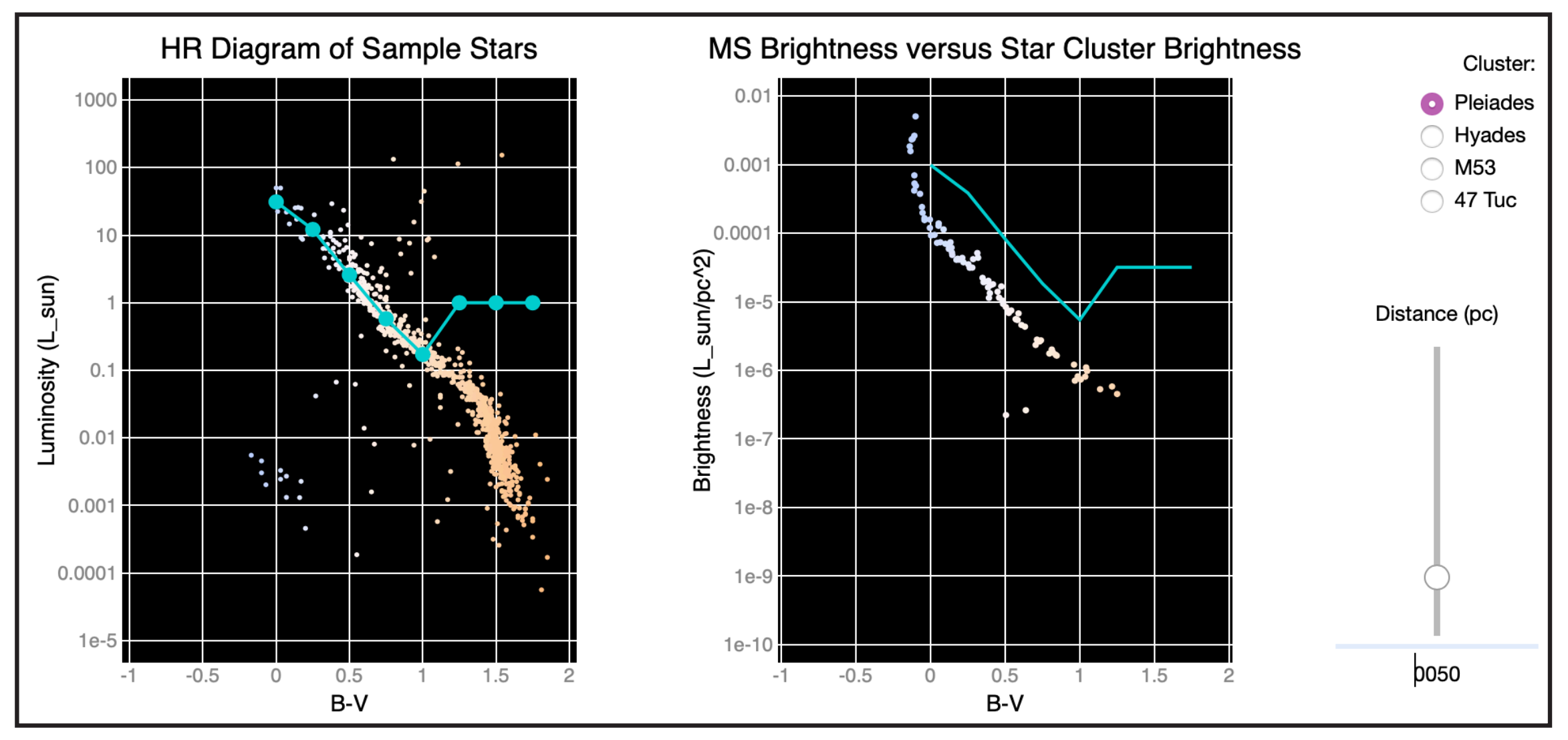


Figure 3(b): A student identifies the main sequence in the HR diagram and the corresponding curve is drawn on the brightness diagram, allowing them to "fit" the distance to a star cluster.

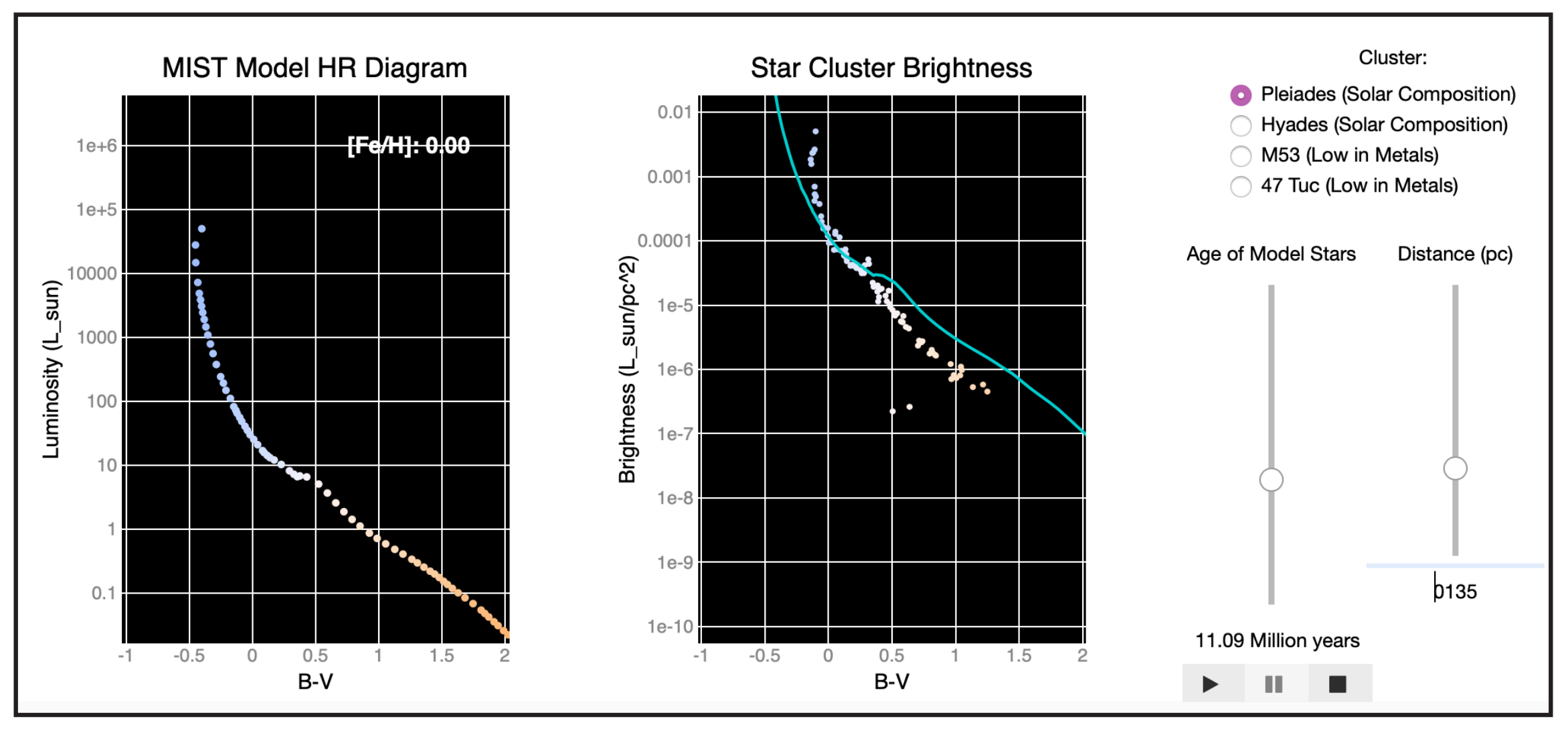


Figure 3(c): Stellar evolution models from MESA are used to allow students to simultaneously fit the distance and age of a star cluster.

Try It Yourself

A GIT Repository of all our code is at
https://bit.ly/AstroInteractives

Or test it out by running the interactives in using
http://bit.ly/AstroInteractivesDemo

Binder will take a few minutes to set up the initial session (Sorry!). We suggest you run the interactives with as wide a browser window as you can.

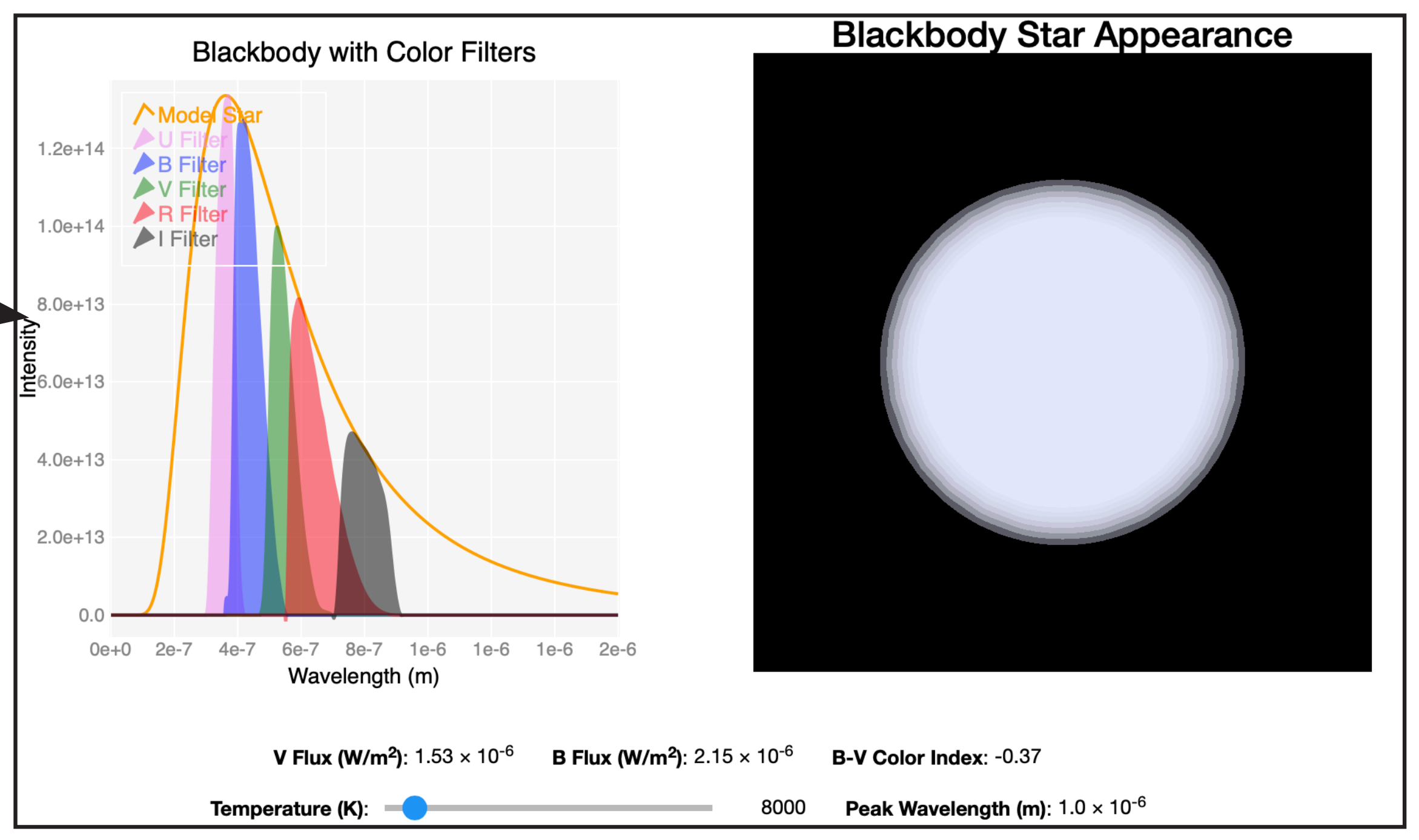


Figure 4: One of the panels on the Blackbody Spectra Interactives page. This allows the student to slide the temperature slider and see just how the B-V color changes (and why) as well as displaying the apparant color of the star.

Technical Specifications

Interactives were written using *IPython* running in a *Jupyter* notebook interface. The specific *Python* packages we used were

- ***ipywidgets***(*https://github.com/jupyter-widgets/ipywidgets*),
- ***pythreejs*** (*https://github.com/jupyter-widgets/pythreejs*),
- ***bqplot*** (*https://github.com/bloomberg/bqplot*),
- ***traitlets*** (*https://github.com/ipython/traitlets*),
- ***numpy*** (*http://www.numpy.org*),
- ***pandas*** (*https://pandas.pydata.org*).

The interactives are run off an implementation of ***The Littlest JupyterHub*** (*https://github.com/jupyterhub/the-littlest-jupyterhub/*) running on an installation of Ubuntu 18.04 on a virtual server allocated four Intel Xeon 2.3GHz CPUs and 6GB of RAM.

Students in the introductory astronomy course log into the *JupyterHub* server using a hyperlink that automatically triggers a GitHub synchronization to update the code and then automatically executes the Jupyter notebook using the ***appmode*** Jupyter extension (*https://github.com/oschuett/appmode*).

We made two modifications to the base *JupyterHub* installation. We added a process to kill old *Jupyter* sessions every morning to keep them from eating up memory. We also modified some of the CSS used in the *JupyterHub* server to "hide" certain interface buttons so that students would not be able to exit the *appmode* view accidentally.

Acknowledgements

ALG and SH would like to thank the MSUM College of Science, Health, and the Environment for their financial support. JEC would like to thank Matthew Craig for presenting the *ipywidgets* workshop and the MSUM IT department for their help setting up our *The Littlest JupyterHub* server.

MINNESOTA STATE UNIVERSITY
MOORHEAD