Many introductory astronomy service courses incorporate labs or other interactive components which use web-based applications. Each of the currently available software, either from textbook publishers or astronomy 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. We wrote some of the interactives in Python running in Jupyter notebooks and have been made available as open source software. We deployed the interactives to students using The Littlest JupiterHub 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 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 web browsers.

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.

### 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, 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.

### Try It Yourself

A GIT Repository of all our code is at
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.

### Technical Specifications

Interactives were written using Python 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),
- @bpyplot (https://github.com/bloomberg/bpyplot),
- @traitlets (https://github.com/jupyter/traitlets),
- @numpy (http://www.numpy.org),
- @pandas (https://pandas.pydata.org).

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

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

We made two modifications to the base JupiterHub instance, including the process to kill old Jupiter sessions every morning to keep them from eating up memory. We also modified some of the CSS used in the JupiterHub server to “hide” certain interface buttons so that students would not be able to exit the appmode view accidently.

### 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 JupiterHub server.

If you have any questions, feel free to contact:
Juan Cabanela
Juan_Cabanela@msum.edu
Twitter: @Juan_Kinda_Guy

Minnesota State University Moorhead
Department of Physics and Astronomy
112th Ave South, Hector 307B
Moorhead MN 56563 USA