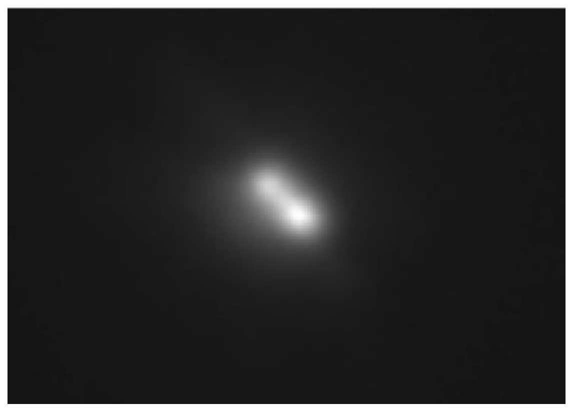
# CS486C – Senior Capstone Design in Computer Science

## Project Description

| **Project Title:** <u>**Three-Dimensional Simulation and Visualization of Binary Asteroids**</u> | |
|---|---|
| **Sponsor Information:** | **Dr. Audrey Thirouin**<br>Dr Will Grundy,<br><br>Lowell Observatory<br>1400 W Mars Hill Road, Flagstaff.<br>Emails: grundy@lowell.edu, thirouin@lowell.edu |

## Project Overview:

Throughout astronomy, binary systems are of particular interest. A binary system is composed of a satellite body orbiting a primary body, e.g., as in the Earth-Moon system. Our team at Lowell Observatory studies binary systems involving small bodies, with a special focus on such systems at the edge of our Solar System. These are of particular interest because several NASA missions have been planned to visit such binary systems, such as the *Lucy mission* which will characterize the Patroclus-Menoetius system (Figure); and the *New Horizons mission*, which flew-by a contact binary named 2014 $MU_{69}$ earlier this year. We are involved in these two missions, leading an international observational campaign to characterize the Patroclus-Menoetius system, and leading a survey to discover Kuiper belt objects similar to 2014 $MU_{69}$.



This image was obtained with the Keck telescope in Hawaii and shows 2 objects. This binary system, Patroclus-Menoetius, is a Jupiter Trojan. This system will be visited by the *NASA's Lucy mission* in 2033. *<u>Credits:</u>* A. Parker, F. Marchis.

Unfortunately, from far away and even with the largest telescope, all we can see is the brightness of a binary system (also known as its lightcurve), and how it changes over time as the two objects rotate around each other. It is possible, however, to infer quite detailed information about the objects and their orbital dynamics just from the careful observation of its lightcurve, thanks to lightcurve modeling.

Last year, our team supervised a CS project to create the first-generation of modeling software dedicated to binary systems (https://www.cefns.nau.edu/capstone/projects/CS/2019/PairedPlanet-S19). This first-generation prototype made use of several independent tools to create a coherent infrastructure for our modeling purposes. The entire infrastructure and documentation of this first-generation software are available.

For this project, we propose to leverage this existing infrastructure to develop one or more new novel modules that will provide a suite of new functions and capabilities. After designing, implementing, and testing these new modules independently, they will need to be connected in to be accessible as new tools within the existing infrastructure.

## Key features of this project include (but are not limited to):

- Implementation of 3D shape models of triaxial objects, and other equilibrium shapes. This represents the core module that we would like added, and include a suite of models related to these new shapes that are not captured in the existing prototype.
- Implementation of functions to deal with reflectance variations across the surface of the system components.

- Extension of the binary modeling infrastructure to be able to handle n-body systems, such as multi-satellite or hierarchical triple systems (one primary with two or more moons or a distant satellite orbiting a pair of objects).
- Implementation of GUI to visualize the behavior of the target system based on the modeling output. The current interface to the infrastructure is IDL-based. A GUI would be much more user-friendly, both to input modeling parameters, and to visualize output of the model. In this way, users could explore various parameters and immediately see the visual effect on the model.
- Ability to create a movie of the system's motion and an automatic plot of the lightcurve. The current prototype produces a series of screenshots of the system's motion and an ASCII file with the lightcurves values. An animation that one could control, e.g., with a time slider, that juxtaposes the lightcurve with the visualized behavior of the bodies would be much more useful.
- Statistical tests to check the accuracy of the model and the osbservational data.
- The minimum viable product must, at very least, be able to simulate and visualize the behavior of binary systems with elongated shapes and/or albedo features, produce the system's lightcurve, and use this to compute the best set of parameters to fit the observations. A truly useful system would implement the additional GUI features outlined above.

## Knowledge, skills, and expertise required for this project:

This project is typical of a software project in scientific computing but it is important to point out that the overall framework has been already developed through the existing prototype. The main aim for this second generation prototype is to extend the rough, generally non-GUI first prototype to implement new features/modules, as well as adding user-friendly GUI interfaces. The sponsors will provide technical support for all mathematical and/or astronomical concepts; the team is there to implement these into software modules. The following are skills the team must have or learn to bring to the project:

- Programming skill in C/C++.
- Very basic IDL knowledge for the main interface.
- Familiarity with the doxygen library is preferable or will have to be learned during the project.
- Familiarity with Linux/Windows computing environments.
- Skill in integrating new features to the existing infrastructure.
- Some knowledge of astronomical observation will need to be learned by the team during this project.

## Equipment Requirements:

- There is no specific equipment required for this project, except a standard software development machine.
- The already developed infrastructure will be provided by the sponsor, as well as the existing documentation. The developed infrastructure has been implemented in Linux and Windows, and documented.

## Software and other Deliverables:

- The new features as described above, incorporated into the infrastructure, deployed and tested successfully in the Lowell computing environment. Must include a complete and clear User Manual for configuring and operating the final software. User manual about the first-generation code is already available and can be used as a template.
- A strong as-built report detailing the design and implementation of the new features and their incorporation in the infrastructure in a complete, clear and professional manner.
- Complete professionally-documented codebase, delivered both as a repository in GitHub; and as a physical archive on a USB drive. The GitHub repository was already created during the first-generation of this project and can be used for the repository of the second-generation.