

# SC2 Project Proposal

---

Project Name: solids-vis-1

## Background

---

Understanding the results of a simulation is equally as important as running a simulation, and one of the best ways to understand a result is to visualize it. Many visualization softwares already exist, but it might be helpful to develop your own so you can customize the software for your unique simulations. Also, it could be a fun and rewarding exercise.

This project would develop a program in python as an interface with blender, a general-purpose 3D modeling tool. The program would produce visualizations related to solid bodies and their motions or deformations. The software could take advantage of colors or materials in the 3D modeling tool to represent temperature, stress, or other states on the solid along with animating deformations and motions. The developed program would take output from well known, standard simulation tools or custom simulations, convert it to a standard input for the software, and create custom visualizations for it.

I am sure this has been done many times before, but I think it would be a fun project to work on and could even be useful. Additionally, this would provide all who work on it with programming experience.

The final goal of this initial project is to create a software that can visualize the motion of rigid bodies. However, in future projects, we can implement special elements to visualize, visualize deformation, or visualize other state variables as mentioned earlier.

## Duration

---

I plan for this project to take two months

## Example Tasks

---

1. Create a standard input file: In order to consistently define visualizations, the input data from the simulation must be represented in an expected way. For this task we will need to define what the format of the file type will be and how the file will be structured. JSON works pretty well for a file format when the data is not too large. The good thing about JSON is it is in plain text so it is easy to edit if needed. The bad thing about JSON is it is in plain text so it is not very efficient.
2. Create routines to connect to a blender instance: Ideally this should work on many platforms. Blender has a python api, so connecting to blender should be relatively straight forward, but there

will need to be configuration done on the user's behalf to point our program to where they have blender installed.

3. Create a routine to instantiate objects in the simulation tool: At this point the objects will be defined in the python program's memory, but we will need to define them in blender. Blender separates its element into objects and meshes. Objects can contain many meshes or no meshes. Objects without a mesh a defined shape will not display but will exist in the modeling environment. The goal of this task is to make an object in blender and have it display in the visualization.

## Extra Resources Required

---

This project requires no extra resources.