

# Real-Time Physics Simulator

Project Proposal

Jon Roelofs

## Abstract

A comprehensive simulation of complex physical models requires the simultaneous calculation of an enormous amount of data. A 2D computational procedure and software tool, RTPS, will be developed to provide a user-friendly way of quickly viewing changes in variables in a static or dynamic environment. When completed, such a system will provide the end user with a lab away from the lab, allowing quick research and development of new ideas. Due to their complex nature, physics simulations are extremely computationally intensive, and thus a strategy to alleviate the user of the background math needs development. Development of RTPS will employ this strategy in order to allow simple evaluation of challenging physics problems drawn from class work and other problems of interest.

## Objective

The aim of this project is to design, code, and test a physics simulator, which can model various systems in static and dynamic environments. The construction of a graphical user interface (GUI) will ease of use by anyone. Java will be the programming language of choice for this project because it forces the programmer to write cleaner code, and it will allow portability across operating systems. Essentially a user will be able to run this program on a Mac, a PC, or even a Linux computer, providing the computer is powerful enough to run the simulation in real-time.

The first phase of development will focus on writing code that will handle the various members (masses, fixed points, springs, strings) which are present in the system. Later on in the development, implementation of a GUI into the system will commence, and finally a physics base will be added to facilitate interaction of members. In this way, interactive testing of code will be simpler, reducing the amount of time spent writing down paper analogues to the problems that are simulated.

## Project Scope

This project will be limited to first semester calculus based physics and therefore will involve only Newtonian mechanics. Thus, there will be a limited set of members made available to the user. This set will include point-masses, fixed-points, springs, as well as rigid & flexible fixed-length mass-less strings and gravitational attractions between particles. All masses are point-masses in this system and therefore collisions between particles are not calculated. This is an attempt to simplify the scope of the project as writing collision detection system would be a large undertaking in itself. Environmental conditions such as gravity and drag are calculated by this system. The design of the project will allow for future expansion into the realm of second semester physics in its design. Therefore, as new modules are created, they can be seamlessly added into the system in the form of updates to the software.

## Project Time-Line

Phase I - Oct 12 – Complete Specifications and class design of Physics Elements.

Phase II - Oct 26 – Design and code graphical user interface.

Phase III - Nov 9 – Complete code for GUI and Physics Core.

Phase IV - Nov 30 – Complete software testing and write-up.

Phase V - Dec 11 – Complete proofreading of code and write-up, Project Due.