Graphical Display of a Physics Simulation

Steven Oetjen
TJHSST Computer Systems Lab 2005 - 2006

Abstract

A physics simulation, in order to adequately demonstrate physical laws and predict an unlimited number of scenarios, must implement a broad range of mathematical equations and provide the user with the ability to set up a scenario with whatever number of objects and arrangements of these objects that he desires. The goal of this project is to create such a simulation.

The purpose is to make a physics simulation that will display objects placed by the user graphically and display information about those objects on a graphical user interface. The goal is to allow the user to place any combination of objects, including particles, springs, and ramps, in a graphical display, input values for these objects, such as constants, coefficients, and variables, and run a simulation that will track these values and display the interactions and positions of the objects graphically in two-dimensions.

Background

The concepts involved are kinematics, dynamics and Newton's laws, conservation of momentum and collisions, gravitational force, and electric charge and force. Variables such as mass, displacement, and charge, only to name a few, are required, as well as the relationships between these variables in the form of equations.

The results may attract the interest of many, including students needing an aid in their understanding of the physical world and how it behaves. The results can also be applied to making or confirming predictions on how certain situations will be resolved.

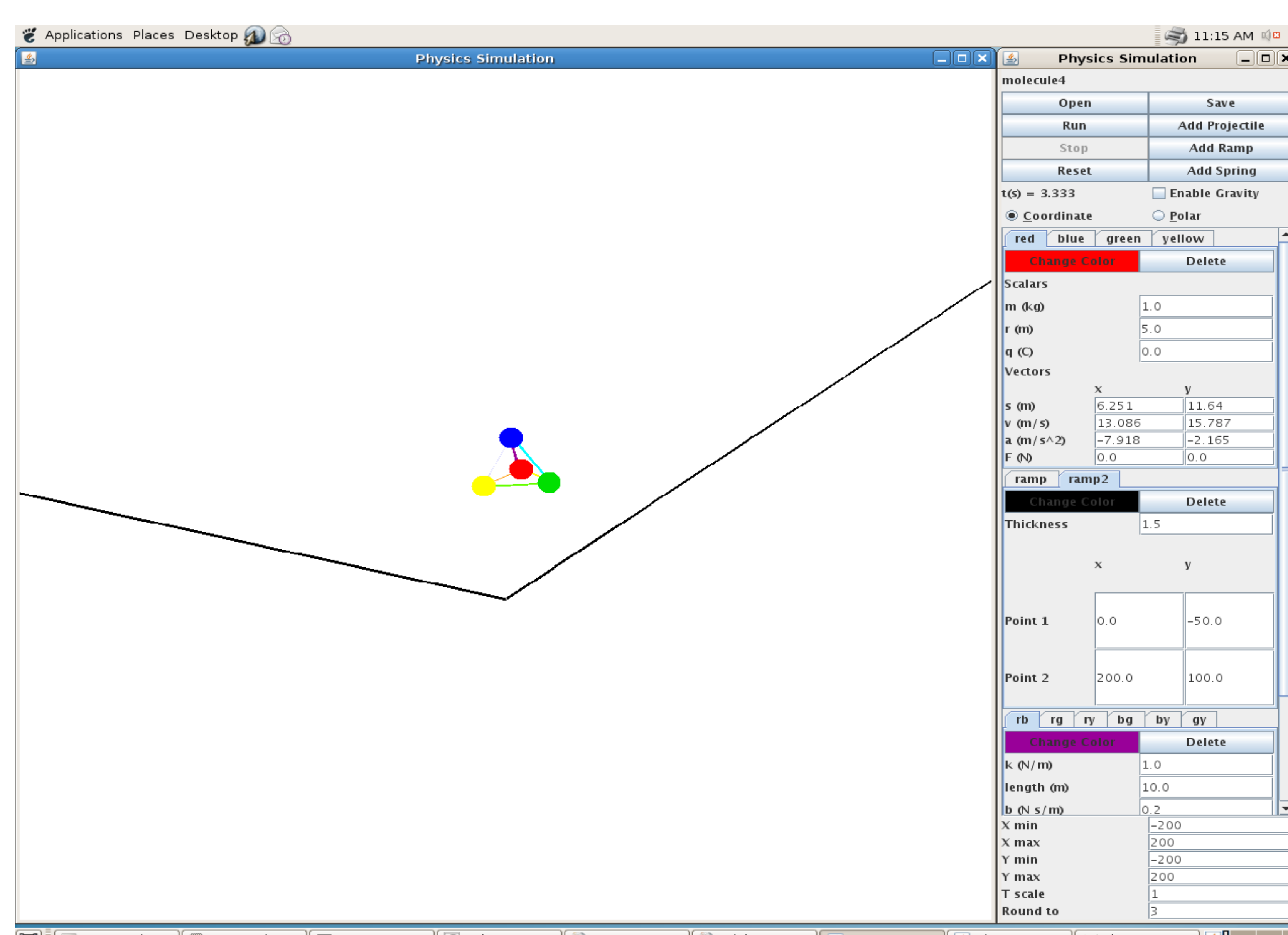
Procedure and Methodology

The project will be divided into several iterations of design, programming, and testing. The tasks of these iterations include implementing equations for kinematics, dynamics, momentum, gravitation, and electrostatics.

Aside from implementing equations and laws, several other components to the program must be implemented, including GUI structure and Layout, graphics panel distance and time scales, support of multiple objects and multiple types of objects, support of coordinate and polar vector conversions, collision detection, and improved integration methods.

Results and Conclusions

Collisions between projectiles now take into account angular effects that would not have been present in collisions that are not head-on. Springs exert correct force, but it is possible to create a scenario in which an object's maximum distance from a spring increases over time. Energy is created due to the propagation of error.



A screenshot of the simulation with a molecule represented by four projectiles connected by springs, and two ramps below. Springs exert forces on attached projectiles, and projectiles can collide with each other and the ramps.