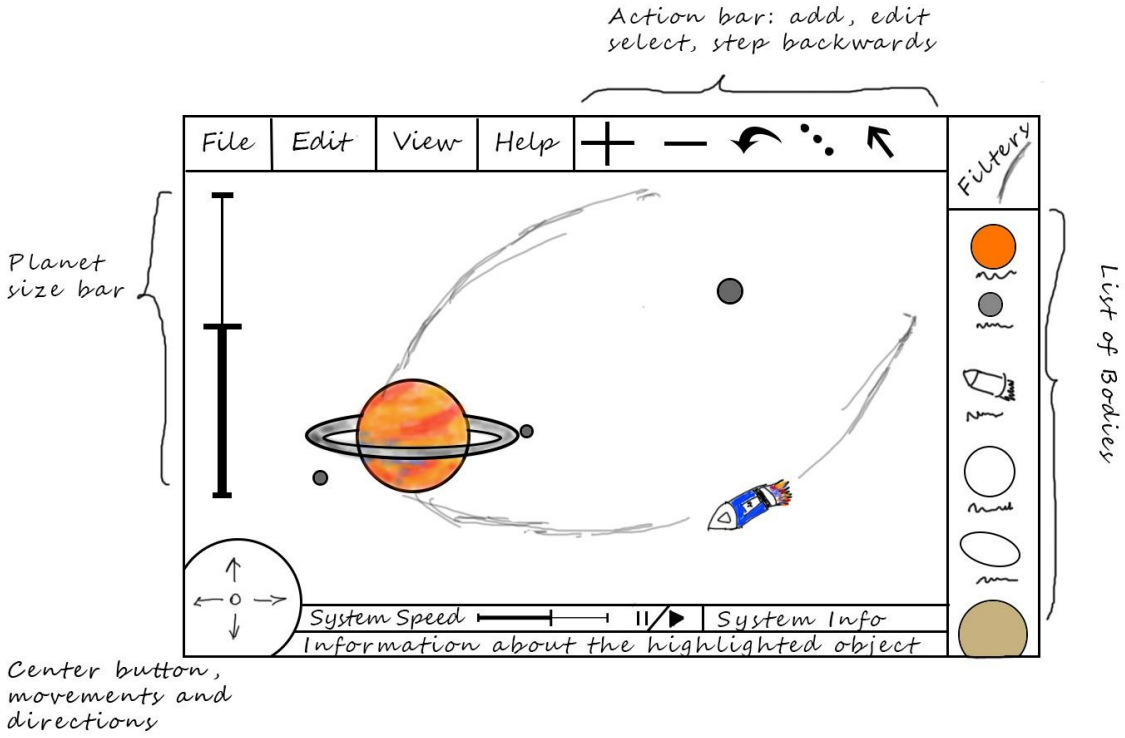
Gravity Simulator

EE810 Programming in C++

Constantine Davantzis, David Dietrick, Dillon Guarino, Anna Petruczynik

Github Repository: https://github.com/CDavantzis/GravitySimulator/

1. Computation
    a. Create a simulation of a Solar System
    b. Allow for creation of bodies in 2d plane
    c. Compute positions, velocities and forces of gravitation bodies
        i. Implement vector class to store positions and velocities
    d. Combine body masses after collision and resulting inertia
    e. Simulate many-body gravitational acceleration
    f. Allow the simulation to work in 3d space
    g. Simulate different types of celestial objects (stars, planets, etc)
    h. Simulate space missions by allowing objects to have thrust, trajectory correction etc.
    i. Simulate gravity effects on non-spherical objects
    j. Allow for a more precise simulation by changing the time between system updates
2. Graphics
    a. Implement 2D graphical representation of gravitational bodies.
    b. GUI interface for setting initial conditions.
        i. Allow user to create celestial objects of different mass, position, initial velocity, type (planet, stars, etc.)
        ii. Allow possible implementation for non-spherical objects
        iii. Ability to import and export initial condition files.
        iv. Ability to resize of the objects, scroll in and out of the system and speed up the gravitational effects for a better user experience.
    c. Implement 3D graphics with the ability to import planet pictures and lay them over bodies.
3. Web Interface
    a. Develop a server that provides a web interface for viewing simulation progress and statistics.



Potential Useful Links

- http://astronomy.stackexchange.com/questions/7806/exercise-2d-orbital-mechanics-simulation-python
- http://astrowww.phys.uvic.ca/~tatum/celmechs.html
- http://physics.princeton.edu/~fpretori/Nbody/
- http://scienceworld.wolfram.com/physics/topics/CelestialMechanics.html
- http://web.mit.edu/8.01t/www/materials/modules/chapter25.pdf
- http://web.mit.edu/pkrein/Public/Final%20Paper%20UW324.pdf
- http://wiki.tomabel.org/index.php?title=Gravitational_N-body_Problem
- http://www.astro.cornell.edu/~berthoud/alpsat/chapter4a.html
- http://www.cs.cmu.edu/~scandal/alg/nbody.html
- http://www.luc.edu/faculty/dslavsk/courses/phys301/classnotes/celestialmechanics.pdf
- https://sites.google.com/a/ucsc.edu/krumholz/teaching-and-courses/ast119_w15/class-11
- https://www.math.ksu.edu/~dbski/writings/twobody.pdf
- https://www.youtube.com/watch?v=TXY6NJm5se0

- Vectors:
    - http://mathworld.wolfram.com/Vector.html
    - http://hplgit.github.io/primer.html/doc/pub/class/._class-readable004.html#sec:class:Vec2D
    - https://gist.github.com/mcleonard/5351452

Potential Useful Links (Wikipedia)

- https://en.wikipedia.org/wiki/Cartesian_coordinate_system
- https://en.wikipedia.org/wiki/Celestial_mechanics
- https://en.wikipedia.org/wiki/Equation_of_the_center
- https://en.wikipedia.org/wiki/Euclidean_vector
- https://en.wikipedia.org/wiki/Gravitational_acceleration
- https://en.wikipedia.org/wiki/Gravitational_two-body_problem
- https://en.wikipedia.org/wiki/Jacobi_coordinates
- https://en.wikipedia.org/wiki/Kepler%27s_laws_of_planetary_motion
- https://en.wikipedia.org/wiki/Kepler_orbit
- https://en.wikipedia.org/wiki/Kepler_problem
- https://en.wikipedia.org/wiki/N-body_problem
- https://en.wikipedia.org/wiki/N-body_simulation
- https://en.wikipedia.org/wiki/Newton%27s_law_of_universal_gravitation
- https://en.wikipedia.org/wiki/Orbital_mechanics
- https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta%E2%80%93Fehlberg_method
- https://en.wikipedia.org/wiki/Two-body_problem