

Niven Achenjang  
Luke Miles

CPS Project Proposal  
Space Simulator

2015-03-24

## short project description

Our project is a 2D space simulator.

This consists of a region of space with interacting celestial bodies (planets, stars, etc).

The user has the following powers:

- place objects in space with initial properties such as size, mass, and velocity
- move and delete said objects
- pause and speed up time

As objects are placed, the system evolves according to the laws of physics.

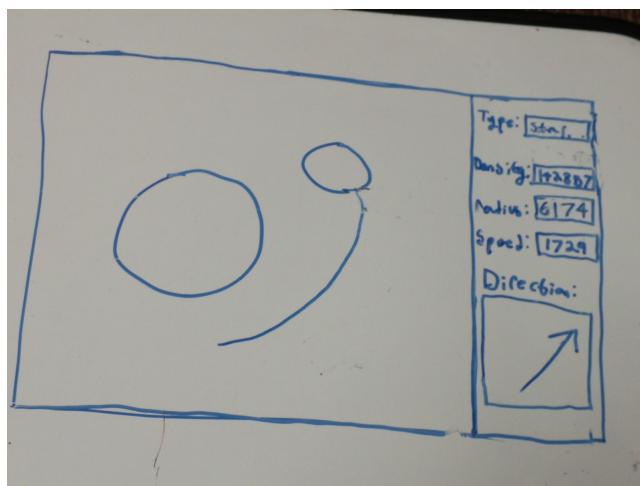
The objects are attracted to each other by gravity and collide in different ways.

## Description of Final Product

The user opens up the program and is presented with a blank canvas or a premade example.

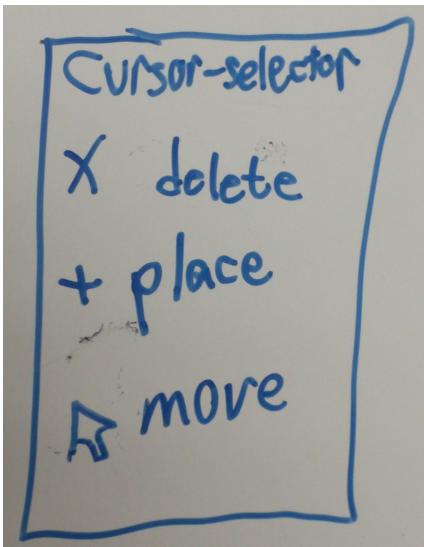
There is an object creation menu on the right, with the following elements:

- sliders for density, radius, and speed of object
- a choice for the type of object (particle, asteroid, planet, star, black hole)  
This affects how the object is displayed,  
how the object collides (star eats planet, but an asteroid and another asteroid bounce, etc...),  
and what the bounds are on the property sliders (Ex. can not have a particle with same radius as a star)
- an arrow (that the user can point) for the object's direction
- A color picker of some sort



There are multiple kinds of cursors:

- object-creation cursor: click to place object
- kill cursor: deletes object that are clicked on
- moving cursor: click and drag objects to throw them around



Say that the user makes 2 objects.

She first selects the object-creation cursor, then configures and places each object.

The first is a very massive star with 0 speed.

The second is a minuscule planet with a velocity tangent to its future orbit around the star.

Depending on how the parameters are defined, the orbit may or may not be stable.

Suppose in this instance it is not stable.

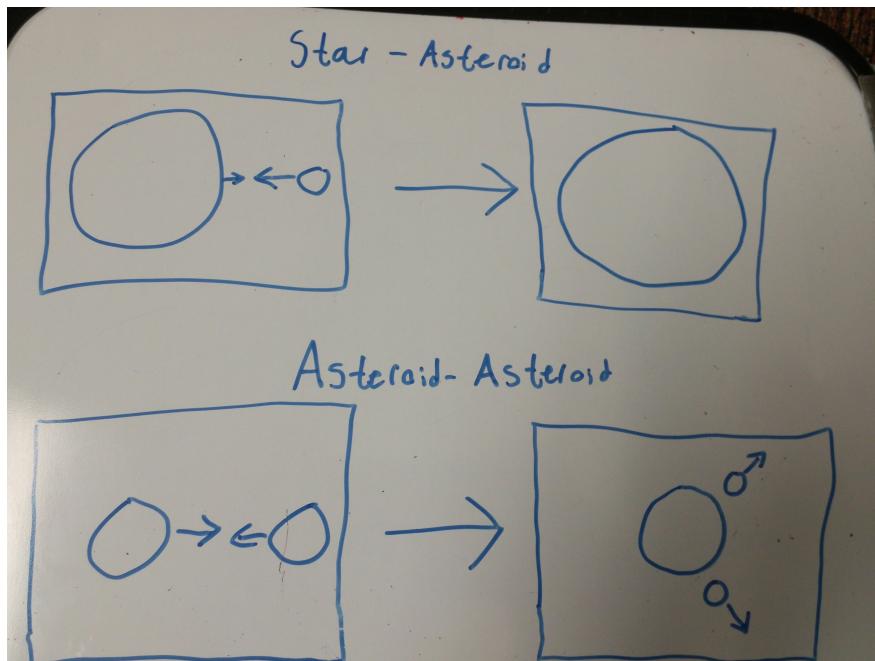
The planet goes around a few times, but is eventually absorbed by the star, making a small change to the star's mass and velocity.

This was a star-planet interaction.

If this were a planet-planet collision, the collision would probably break the objects apart.

If this were an asteroid-planet interaction, the asteroid would be absorbed into the planet.

If this were a particle-particle interaction, they might stick or bounce apart.



Another major function is the time multiplier.

This allows the user to change the rate of time flow to a natural multiple of the base time flow.

It could be 0x (paused), 1x (normal), 2x (twice speed), 3x, ...

For example, if the user has a stable-looking orbit that she wants to see the long-term development of, she can put things into 100x and see the years fly by.

#### THOUGHTS FOR LATER

- select an object and adjust its properties.
- save states
- relate project to Germany
- trace paths of objects
- allow user to change rules of universe

#### increments

- increment 1:  
Simple masses are floating around and being attracted and such. The user has no power, no collisions.
- increment 2:  
The user can place masses and change the rate of time. Collisions still don't work.
- increment 3:  
The 3 cursors (object-creation, kill, move) are all implemented and basically working.
- increment 4:  
Get object creation menu working smoothly with all options functional.
- increment 5:  
Get collisions working smoothly.
- increment 6:  
Implement thoughts for later as time allows