Niven Achenjang
Luke Miles
2015-05-12
CPS Final Report

1. how to run the code

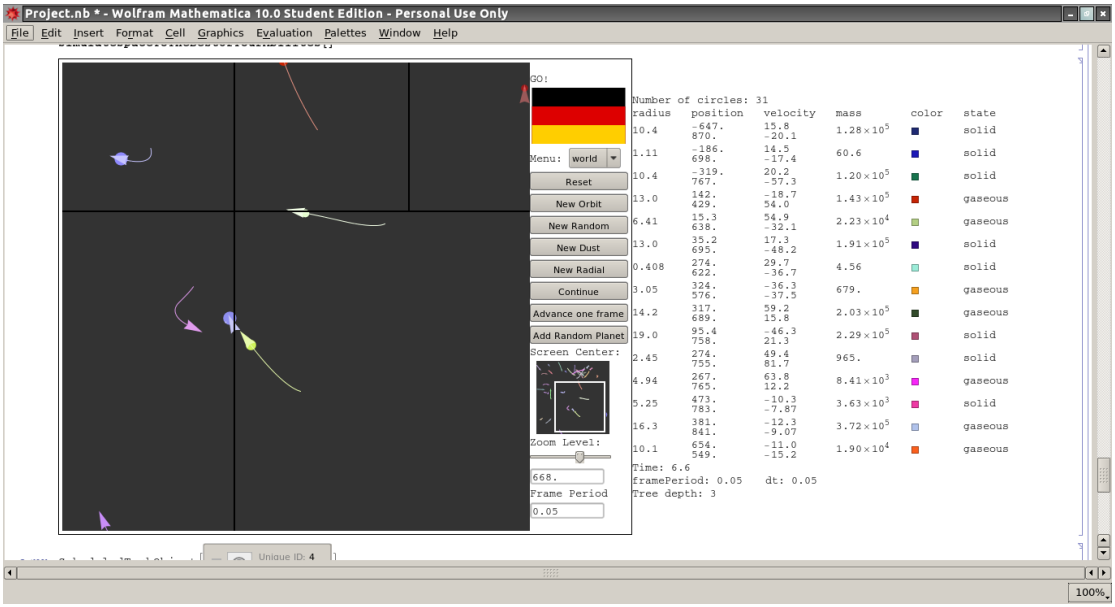
First, evaluate QuadTree.nb in a fresh kernel. Then, in the same session, evaluate Project.nb. At the very bottom of Project.nb, there should be some graphics showing. This is our program

2. detailed description

Space-sim is a physics sandbox for simulating object interactions in space. It features precise (ish) and realistic gravitational interactions, along with a full suite of collisions. Space-sim also has a sleek and useable interface for governing and preparing said interactions.

To be an expert at space-sim, you need to understand the menus. You can select a menu from the popup menu labeld menu. There are 5 menus: world, object, cursor, rules, and file.

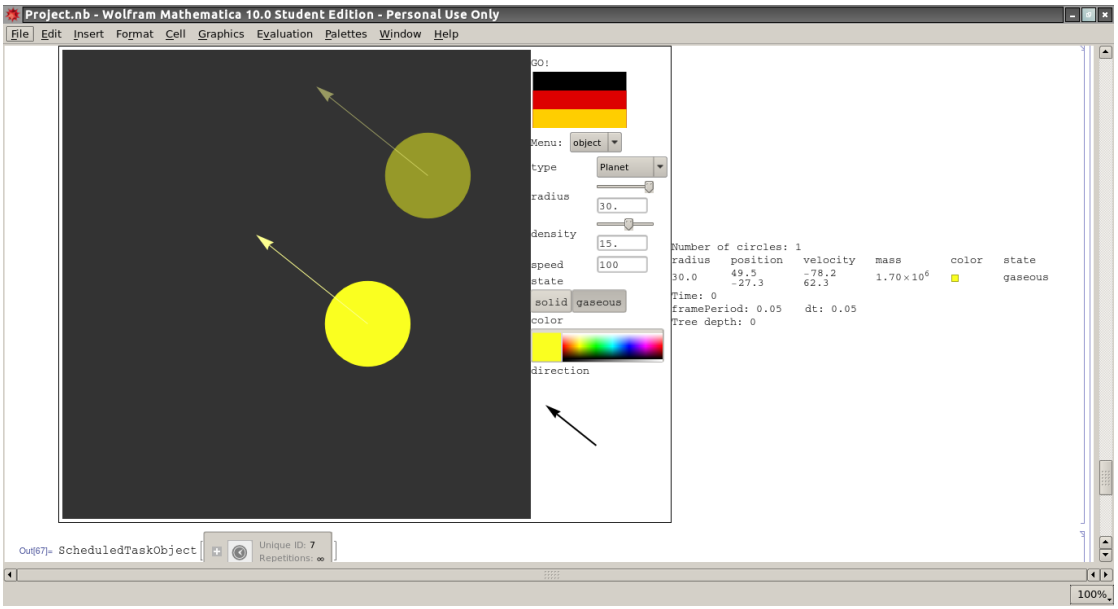
World Menu:



- the big graphic on the left is called the **viewing plane**. This is where you can see the all the physics take place.
- **Reset**: Deletes all circs. The screen is cleared.
- **New Orbit**: Reset and load a simple, stable orbit on screen.
- **New Random**: Reset and load random circles throughout space. Perhaps the best feature in Space-sim. This is what is in the screenshot.
- **New Dust**: Like new random, but smaller and more objects
- **New Radial**: Like new random, but all the circs have velocities tangent to a cirle centered at the center of space.
- **Pause/Continue**: Pause or Resume the simulation. Essentially, changes whether or not the positions of the circs are updated each frame.

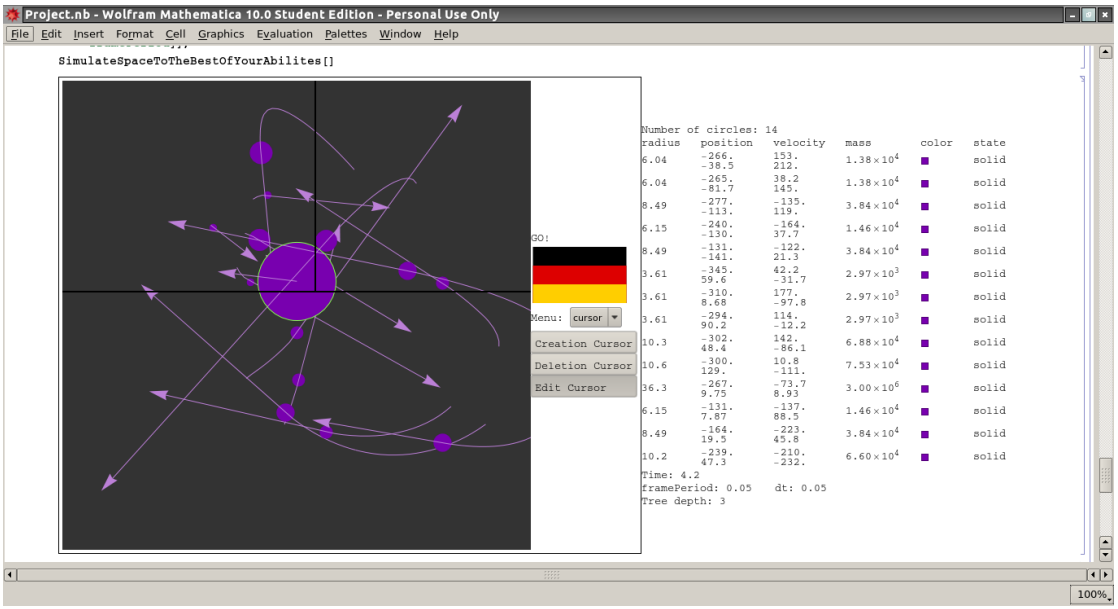
- **Advance one frame**: Updates all the circs' positions once.
- **Add random planet**: Adds a planet somewhere in space, without resetting things.
- **Screen Center**: This is a map of all of space. The white square (which you can drag around) covers what is in the viewing plane.
- **Zoom Level**: Changes the side length of the square in Screen Center. Make it smaller to zoom in more.
- **Frame Period**: How long to wait between updating the frames. A smaller number will make a smoother, faster simulation (if your computer can handle it).

Object Menu:



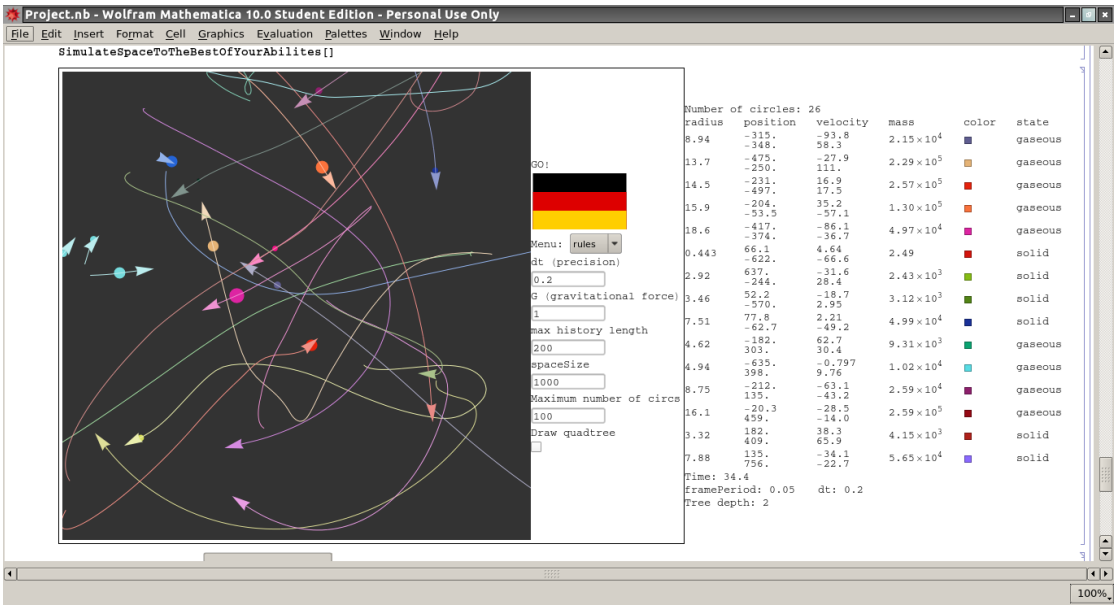
- **type**: Planet, Star, Particle, Asteroid, or Black Hole. These types are just guidelines for a reasonable radius and density. This is not stored in the circ.
- **radius**: The radius of the circ you place. Stars can be bigger than particles, etc.
- **density**: How dense your circ is. Mass scales linearly with density and cubicly with radius. Black Holes can be very, very dense, which may cause computation problems. If you do place a black hole, make precision (in the Rules menu) very very small. on the order of 10^-23.
- **speed**: How far your circ moves in a single frame.
- **state**: Whether your circ is a solid or a gas. Affects collisions, which are detailed in section 3.
- **color**: Only affects appearences.
- **direction**: multiplies with speed to give your circ a velocity

Cursor Menu:



- **Creation Cursor**: The default, most useful cursor. If you click on screen with this cursor, the circ described in the object menu will be placed there.
- **Deletion Cursor**: Click on a circ to remove it.
- **Edit Cursor**: Click on a circ to select it. Then go in the object menu to edit its properties.

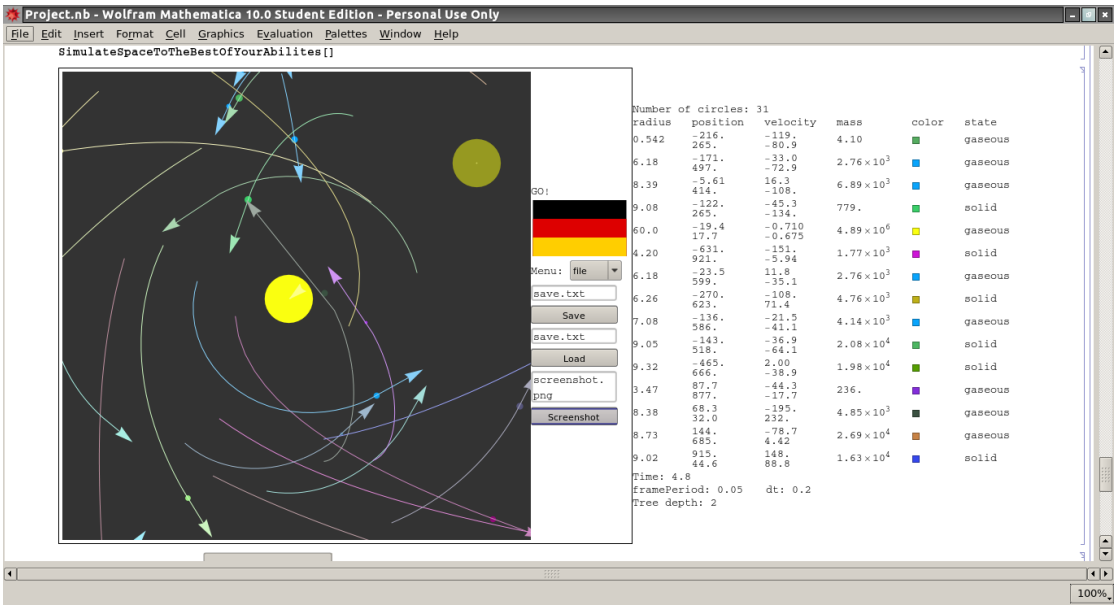
Rules Menu:



- **dt (precision)**: A couple of the physics equations used under the hood use a number called dt. A smaller dt means a smoother, more precise simulation. It also means the simulation will run slower.
- **G (gravitational force)**: The constant in $\frac{G m_1 m_2}{r^2}$. A larger G makes circs more strongly attract. A negative G makes circs repel.

- **max history length**: Each circ has a tail drawn behind it that is the path it just followed. A tail is stored as a list of points, a *history* if you will. It is expensive to store and modify large histories, so a maximum length is kept. A larger number here means the tails go farther back in time.
- **spaceSize**: How large the universe is. This controls how large the overall box is in the Screen Center slider in the world menu. Important because circles that have an x or y position larger than spaceSize are deleted.
- **Maximum number of circs**: Occasionally, if you have many large solid objects to start with, collisions will result in a fast growth of the number of circs. Although it is inherintly limited by the physics of collisions, the number of circs might get up to like 1000 before it stabilizes. Since *Mathematica* might crash with that many circs, instead the program automatically pauses when the number of circs exceeds a threshold. The default is 100. Once the program pauses, you can delete circs with the deletion cursor, or hit Reset, or make some objects gaseus.
- **Draw quadtree**: Gravity calculations are done using a structure called a quadtree. You can choose to display it or not.

File Menu:



- **Save**: Saves the state of the circs (radii, positions, masses, colors, etc...) in the filepath above. Note that settings such as rules and object menu are not saved.
- **Load**: Compliment to save.
- **Screenshot**: If you have something really gorgeus on the viewing plane, you can save it to some arbitrary filepath. Extension is not automatically added, so make sure to include it.

3. explanation of algorithms

The physics ≈uses three major algorithms: calculating gravity, calculating gravity faster w/ a quadtree, and collisions.

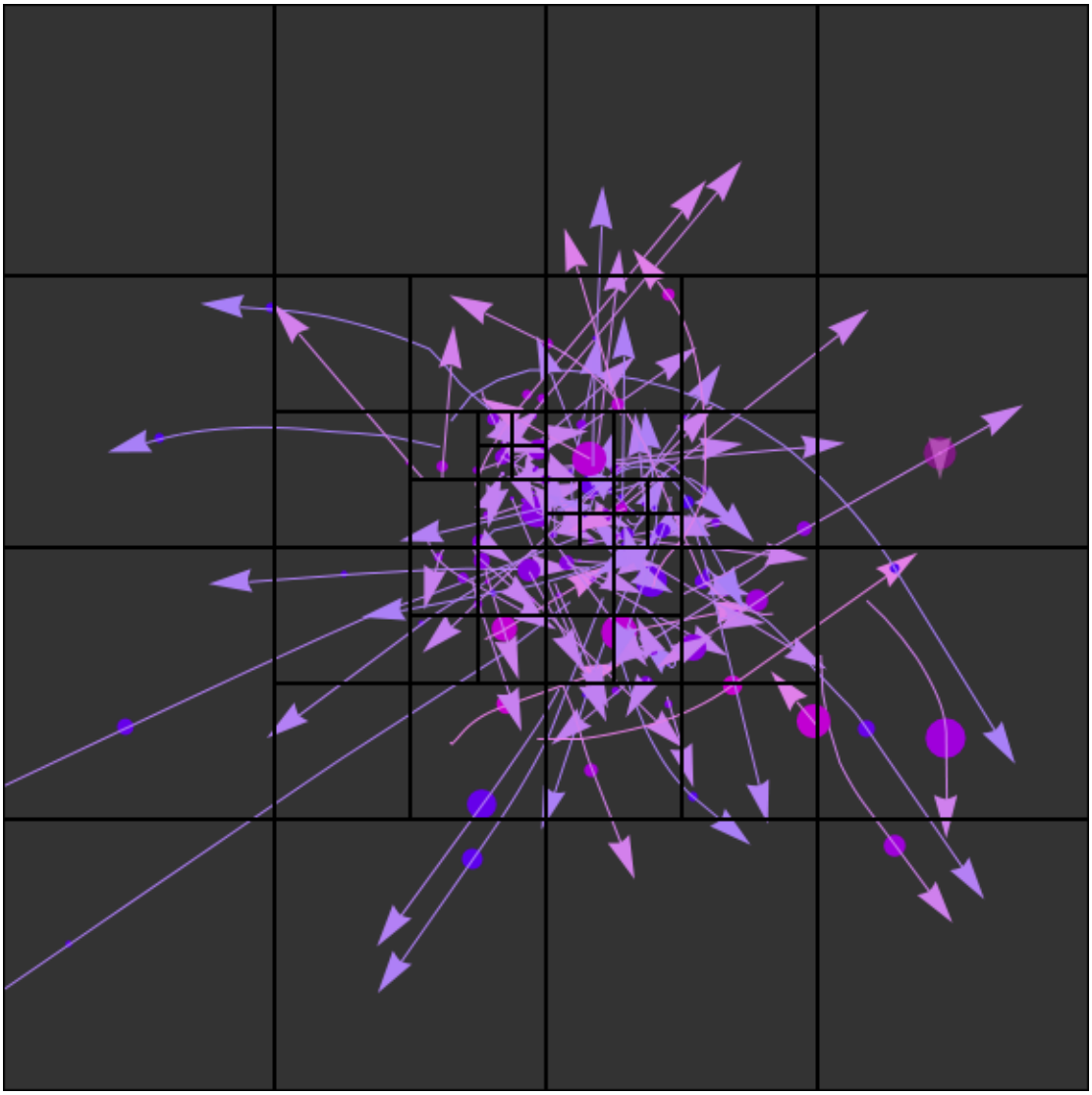
calculating gravity:

Although not really research-based, gravity causes all non-collision acceleration, and is worth explaining.

The equation for gravitational force from a mass m_2 on a mass m_1 is $F = \frac{G m_1 m_2}{r^2}$ where r is distance

between m_1 and m_2.
Keeping track of vectors, you can sum these to get the net gravitational force on an object
$F = \sum \frac{G m_1 m_2}{r^2}$ for m_2 in the universe $= G m_1 \left(\sum \frac{m_2}{r^2} \text{ for } m_2 \text{ in the universe} \right)$
Since $F = m\ a$, $a = G\left(\sum \frac{m_2}{r^2} \text{ for } m_2 \text{ in the universe} \right)$.
And since $a = \frac{dv}{dt} \approx \frac{\Delta v}{\Delta t}$, $\Delta v = \Delta t\, G\left(\sum \frac{m_2}{r^2} \text{ for } m_2 \text{ in the universe} \right)$. You can make Δt smaller for a better approximation of acceleration. Each frame, every planet's velocity is changed according to this formula. And, every planet's velocity is added to it's position. This is what causes orbits and weaving paths and so on.

quadtrees:



If you have 100 circs, each circ has to sum 99 other circs, resulting in a total of $100 * 99 = 9900$ total m / r^2 calculations. A quadtree dynamically splits space into squares, allowing a dramatic reduction in this number. If there are more than a few objects in a square, it is split again until each square only holds a certain number of objects. Then, a circ that is far away from the square can approximate all the circs inside it as one large circ (whose mass is the sum of the masses in the square and position is the center

of mass) and cut things down from $100*99 = 9900$ to say $100*20 = 2000$ calculations.

collisions:

There are three major collision types: gas-gas, solid-solid, and solid-gas.

First, gas-gas. If two gaseus objects collide, they completely merge. The new object has the average color, velocty, etc... of the 2 original circs, weighted by mass. E.g. a large yellow star merged with a small red gas-planet will make a very light orange, slightly larger star.

Second, solid-solid. If two, about-equally-sized, solid objects collide, they have a merged (as if gas-gas) kernel in the center, with the remaining mass blasting outwards in a number of circs (between 2 and 5). If one is significantly smaller (> GoldenRatio) than the other, they merge as if gaseus.

Last, solid-gas. If the solid is moving quickly and a little smaller than the gas, the solid will be slowed and split the gas. If the solid is way smaller than the gas, the gas will eat the solid. If the solid is way larger than the gas, the solid will eat the gas (consider it an atmosphere).

4. description of files/structure

Just the two files, **Quadtree.nb** and **Project.nb**.

5. major data structures

A circle is an association list with keys like “vel” for velocity, “col” for color, and “rad” for radius.
All the circles are stored in a big list called **circs**.

A quadtree is a recursively definied association list with keys “chlds” for children quadtrees, “crnrs” for the corners of the square it encapsulates, “CofM” for the center of mass of the circs in the tree, and “mas” for the total mass of all the circs in the tree. The base quad trees have a list of circs associated with the “chlds” key instead of four children quadtrees. It is called **circTree**.

6. overview of how code works

It boils down to having the user run the simulation and running a round of physics each frame. In addition to sections 2, 4, and 5, we are assuming you have read section 3.

Functions in QuadTree.nb:

- divideQuadTree
  - in: a quad tree
  - out: A new quad tree that is the result of splitting the inputted tree into four parts
  - algo: Creates four quad trees where their confining squares combine to to the original trees square. It then partitions the circs by storing them in the new quad tree thats square contains them.
- addCircToQuadTree
  - in: a quad tree and a circ
  - out: A new quad tree that is the result of adding the circ to the inputted tree
  - algo: Recursively traverse the tree until you end up at the undivided tree that has the circ inside of its square. Store circ in that tree
- removeCircFromQuadTree
  - in: a quad tree and a circ

- out: A new quad tree that is the result of removing the circ from the inputted tree
- algo: Recursively traverse tree and remove circ when found

- collectCircs
  - in: a quad tree
  - out: a list of circs contained in the tree
  - algo: joins all the circs from all of the trees children
- undivideQuadTree
  - in: a quad tree
  - out: a quad tree that is the result of undiving the inputted tree
  - algo: collect all circs from the tree's children and have the tree store them directly instead of through children
- gravitationalForce
  - in: a quad tree, a position, and a precision parameter
  - out: the force of gravity acted upon an object at that position
  - algo: Barnes-Hut Simulation. Traverse tree and calculate gravity directly with all circs in squares near the position. Approximate gravity for circs in squares far from the position by using the center of mass and total mass of the square. The actual definition of near and far is determined by the precision parameter
- mapQuadTree
  - in: a function and a quad tree
  - out: a quad tree that is the result of applying f to all circs in the inputted quad tree
  - algo: Traverse the tree and call map on all leaf nodes
- positionPred
  - in: a list and a predicate
  - out: The indices of all elements of the list that satisify the predicate
  - algo: Have MapIndexed return the indices of the elements that satisfy the predicate (and missing for those that don't) then select the ones that are not missing
- updateCircInTree
  - in: a quad tree, a position, and an association list of changes
  - out: a quad tree that is the result of applying changes to the circ located at the position
  - algo: Recursively traverse the tree and apply the changes to the first circ found such that contains position (the distance between its center and the given position is less than its radius)
- graphicsQuadTree
  - in: a quad tree
  - out: a graphics list used to display the tree
  - algo: Create a list to diplay the square confining the tree and then do the same for all of its children

Funtions in Project.nb:

- makeCirc

- purpose: create circs from properties specified in object creation menu
- input: the radius, position, velocity, density, state, color, and history of a circ
- output: a circ (association) with those properties.
- algorithm: just makes an assoc

- updateAssoc
  - purpose: updating circles' properties
  - input: an original assoc and an assoc of changes
  - output: an assoc with the the changes applied
  - algo: makes a copy of assoc, changes said copy, and returns it
- moveCirc:
  - purpose: updating a circle with a new position and history
  - in: a circ, precision, and minimum step between point in history
  - out: the new circle
  - algo: obvious
- splitDirs:
  - purpose: generate random directions pointing out in a circle, like equally dividing a pizza
  - in: number of directions
  - out: list of pairs
  - algo: generate nums between 0 and 2π, take sin and cos of each
- mergeCircs:
  - purpose: used in collisions to make a new circ w/ combined properties
  - in: 2 circs
  - out: 1 bigger circ
  - algo: use makeCircMass to make a new circle, feed it the summed properties
- breakCircs:
  - purpose: used in solid-solid collisions to make the kernel in the middle and the flying away bits
  - in: 2 circs
  - out: several circs
  - algo: use mergeCircs for the kernel and splitDirs for the bits. Uses a random number of bits and a random fraction of mass remains in kernel.
- resolveCollisions:
  - purpose: do all collisions
  - in: list of circs
  - out: new list of circs
  - algo: apply mergeCircs or breakCircs to touching circs, depending on states
- makeRandCirc:

- purpose: make a random circle
- in: maximum values for various properties (radius, position, velocity, density)
- out: a random circle
- algo: obvious

- makeRandCircRadial:
  - like makeRandCirc, except has the tangent velocity stuff
- SimulateSpaceToTheBestOfYourAbilites
  - in: notta
  - out: the entire interface
  - algo:
    - First, make and print all the menus.
    - Then, start a scheduled task that refreshes the frame and everything every framePeriod seconds.
    - refreshing the frame involves calling resolveCollisions, the function in QuadTree.nb which deos gravity, and removing circs outside space

7. external code

All code by Luke Miles or Niven Achenjang.

8. limitations

No known bugs. Everything is a bit slow at times. The all-mouse interface can be limiting at times.

9. references

no

10. signed statement

Our accomplishments are best seen through git commits:

```
Author: Luke Miles <luke.miles090@topper.wku.edu>
Date:   Mon May 11 02:36:57 2015 -0500
    added instructions for running

Author: NivenT <nachenjang@gmail.com>
Date:   Fri May 8 13:49:59 2015 -0500
    Added field lines

Author: NivenT <nachenjang@gmail.com>
Date:   Thu May 7 16:01:29 2015 -0500
    Screenshots added to File menu

Author: NivenT <nachenjang@gmail.com>
Date:   Wed May 6 11:17:54 2015 -0500
    Final changes v2

Author: NivenT <nachenjang@gmail.com>
Date:   Wed May 6 09:45:32 2015 -0500
    Final edits

Author: NivenT <nachenjang@gmail.com>
Date:   Tue May 5 23:49:32 2015 -0500
    Automatically pauses when too many circs

Author: NivenT <nachenjang@gmail.com>
Date:   Tue May 5 23:25:25 2015 -0500
    I am not sure

Author: NivenT <nachenjang@gmail.com>
Date:   Tue May 5 20:55:42 2015 -0500
    Added new type: black hole. Changed bounds on number of random planets created

Author: NivenT <nachenjang@gmail.com>
Date:   Tue May 5 20:45:17 2015 -0500
    Fixed bug where solid planets were displayed as gaseous and vice versa

Author: NivenT <nachenjang@gmail.com>
Date:   Tue May 5 20:44:00 2015 -0500
    Update rules menu and other small changes
```

Author: NivenT <nachenjang@gmail.com>
Date: Tue May 5 20:40:39 2015 -0500
Gaseous-Solid collisions update

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Tue May 5 17:10:05 2015 -0500
all collisions now preserve momentum

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Mon May 4 13:49:02 2015 -0500
user can change rules of universe

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Sun May 3 19:56:15 2015 -0500
user can change other rules of universe

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Sun May 3 19:46:22 2015 -0500
user can change the rate of time

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Sun May 3 19:09:54 2015 -0500
some code cleaning

Author: NivenT <nachenjang@gmail.com>
Date: Sat May 2 13:19:02 2015 -0500
Did some collision stuff. Still needed improving

Author: NivenT <nachenjang@gmail.com>
Date: Thu Apr 30 16:56:33 2015 -0500
Usefull commit(comment?)

Author: NivenT <nachenjang@gmail.com>
Date: Thu Apr 30 16:54:29 2015 -0500
Screen center graphic updated

Author: NivenT <nachenjang@gmail.com>
Date: Thu Apr 30 16:43:58 2015 -0500
it goes wrong

Author: NivenT <nachenjang@gmail.com>
Date: Tue Apr 28 14:35:18 2015 -0500
Quad tree now drawable

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Mon Apr 27 16:23:46 2015 -0500
added (kinda crappy) solid-solid collisions

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Mon Apr 27 16:19:08 2015 -0500
made "reset" into "reset[]" as it is more of a function

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Mon Apr 27 16:07:11 2015 -0500
progress report 2

Author: NivenT <nachenjang@gmail.com>
Date: Sun Apr 26 14:39:58 2015 -0500
Changes

Author: NivenT <nachenjang@gmail.com>
Date: Sun Apr 26 14:31:19 2015 -0500
bug fixes

Author: NivenT <nachenjang@gmail.com>
Date: Sun Apr 26 14:23:50 2015 -0500
Cursor menu

Author: NivenT <nachenjang@gmail.com>
Date: Fri Apr 24 10:44:44 2015 -0500
a

Author: NivenT <nachenjang@gmail.com>
Date: Fri Apr 24 10:41:39 2015 -0500
a

Author: NivenT <nachenjang@gmail.com>
Date: Sat Apr 18 11:40:04 2015 -0500
Evaluate quadtree first, then project

Author: NivenT <nachenjang@gmail.com>
Date: Fri Apr 17 16:16:55 2015 -0500
Updated orbit

Author: NivenT <nachenjang@gmail.com>
Date: Fri Apr 17 15:47:54 2015 -0500
Minor changes

Author: NivenT <nachenjang@gmail.com>
Date: Wed Apr 15 21:57:49 2015 -0500
Update quad tree stuff. Fixed bugs

Author: NivenT <nachenjang@gmail.com>
Date: Wed Apr 15 19:53:46 2015 -0500
Code for quad tree. Will soon combine with main project

Author: NivenT <nachenjang@gmail.com>
Date: Wed Apr 15 18:13:04 2015 -0500
New button

Author: NivenT <nachenjang@gmail.com>
Date: Sun Apr 12 23:45:49 2015 -0500
Minor changes

Author: NivenT <nachenjang@gmail.com>
Date: Sun Apr 12 21:45:28 2015 -0500
Small change

Author: NivenT <nachenjang@gmail.com>
Date: Sun Apr 12 21:43:47 2015 -0500
No more None error

Author: NivenT <nachenjang@gmail.com>
Date: Fri Apr 10 01:26:17 2015 -0500
Only renders circles on the screen

Author: NivenT <nachenjang@gmail.com>
Date: Fri Apr 10 01:01:35 2015 -0500
Fixed some bugs. Reintroduced random crashing

Author: NivenT <nachenjang@gmail.com>
Date: Thu Apr 9 16:52:43 2015 -0500
Started updating object creation menu

Author: NivenT <nachenjang@gmail.com>

Date: Thu Apr 9 16:35:27 2015 -0500
Started updating object creation menu

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Tue Apr 7 14:26:39 2015 -0500
got tabs working

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Tue Apr 7 14:04:53 2015 -0500
EVERYTHING WORRRKKSSgit commit -m

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Wed Apr 1 12:57:15 2015 -0500
now checks if 2 planets are gaseous before combining them

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Wed Apr 1 12:51:23 2015 -0500
changed makeCircMass to use makeCirc. so much shorter. computation is insignificant

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Wed Apr 1 12:40:19 2015 -0500
fixed and abstraced resolveCollisions

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Wed Apr 1 11:33:49 2015 -0500
some code cleaning, broke hakf of object-creation

Author: NivenT <nachenjang@gmail.com>
Date: Tue Mar 31 19:42:49 2015 -0500
No longer using framePeriod. Not sure if good change or not.

Author: NivenT <nachenjang@gmail.com>
Date: Tue Mar 31 19:19:27 2015 -0500
Minor changes and bug fixes

Author: NivenT <nachenjang@gmail.com>
Date: Tue Mar 31 16:04:33 2015 -0500
Little changes made.

Author: NivenT <nachenjang@gmail.com>
Date: Tue Mar 31 14:10:09 2015 -0500
Continued with updates from last time. Minor changes/bug fixes. Appears to run more smoothly and quit kernal less. If you haven't, look at commit before this.

Author: NivenT <nachenjang@gmail.com>
Date: Tue Mar 31 13:34:47 2015 -0500
Undid changes to resolveCollision because they were stupid IMO. Updated frame in SimulateSpace... Added planet to beginning of circs for diplaying new planet to add. Got rid of useless color and stuff between frame and tabview.

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Tue Mar 31 09:35:17 2015 -0500
small improvement to resolveCollisions

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Mon Mar 30 20:39:07 2015 -0500
Efficientized drawCircs

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Mon Mar 30 20:14:20 2015 -0500
added some info about project

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Mon Mar 30 20:09:03 2015 -0500
added proposal from beginning of project

Author: NivenT <nachenjang@gmail.com>
Date: Mon Mar 30 19:25:53 2015 -0500
Draw circle where you are going to place circle. Also, do not add features for now. Just make efficient.

Author: NivenT <nachenjang@gmail.com>
Date: Mon Mar 30 19:23:18 2015 -0500
Draw circle where you are going to place circle. Also, do not add features for now. Just make efficient.

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Mon Mar 30 17:37:59 2015 -0500
made 2 selectors into a selector bar

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Mon Mar 30 17:31:42 2015 -0500
made direction arrow dragable

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Mon Mar 30 16:27:21 2015 -0500
added object creation menu

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Sat Mar 28 10:36:27 2015 -0500
combined line and arrow. added back makeframes for testing.removed some calls to Timing.

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Fri Mar 27 18:49:17 2015 -0500
removed useless makeFrames function. fixed framerate issue in main func

Author: NivenT <nachenjang@gmail.com>
Date: Fri Mar 27 18:40:11 2015 -0500
Added state to circles. Made history update based on distance travelled

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Fri Mar 27 18:11:29 2015 -0500
made circle into a association rather than a list

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Fri Mar 27 11:58:15 2015 -0500
cleaned code a little. tableform for planet info

Author: NivenT <nachenjang@gmail.com>
Date: Thu Mar 26 22:00:18 2015 -0500
Updated circle representation (colors, density, etc.). Did UI stuff

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Thu Mar 26 14:57:07 2015 -0500
initial ball physics

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Tue Mar 24 19:29:28 2015 -0500
first commit

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Tue Mar 24 19:25:01 2015 -0500
first commit

Author: Luke Miles <luke.miles090@topper.wku.edu>
Date: Tue Mar 24 19:23:20 2015 -0500
test commit

Luke Miles:_____

Niven Achenjang:_____