LOG

**December 15th:**

-created Newtonian Object class with inheritable properties such as mass, this class can be displayed on the compound interface by extending the circle class in flash

**December 16th:**

-edit the compound interface (stage) to accept mouse click events, worked out logic to make an array of Newtonian Objects (array of ‘things’)

**December 18th:**

-used vector logic to determine location of objects on stage, whether they need a plus or minus to their velocity to approach a given ‘thing’

**December 21st:**

-worked out vector analysis for objects, made uniform acceleration between objects using trigonometric logic

**December 28th:**

-started looping logic to encompass relationships between all objects of a given array and their given attractions

-created the add dimension function to edit arrays for later computation when loop solving for attraction between objects, basically repeats an array so that an object is always ‘i’ spaces away from the next different object

**January 1st:**

-used Newtonian gravitation (inverse square law) to calculate force magnitudes

-for some reason objects along the left side of the screen waver and move at invalid rates

**January 2nd:**

-fixed ongoing problem about wavering objects (just a distance calculation flaw)

-experimented with some repulsion math so that when objects get too close, they bounce… nothing seems to be working very well as they stop attracting when they interact

-need to implement friendlier UI, hard coding this thing is not as fun as it sounds!

**January 3rd, 4th:** Worked on feasibility study, no time for development

**January 5th:**

-now the radius of the circle reflects the radius of the planet it corresponds to

-found a background

**January 6th:**

-experimented with various repulsion math, none of which have succeeded thus far, one idea is to combine planets when they collide instead of continuing to calculate forces that are increasing more and more, could also use a Boolean similar to that of an “inAir” in one dimensional gravity programs

**January 7th:** I think a better name than 2D-UDUGS (2 dimensional user determined universal gravitation simulator) would simply be Fiziks Program… I like “Fiziks.”

-derived a more mathematically sound method of adding rebound physics without yet implementing, have to test tomorrow and if it is not effective, going to default to combining planets (I wanted to avoid making an agar.io clone.)

**January 8th:**

-rebound method works to an extent, it slows the other processes negligibly because the loop needs to step back and recalculate for the object again, just need to work on optimizing it

-added random colour generator for objects ☺

-Note to self: Need to fix problem where when 2 objects share the same x or y coordinates the program stops!