**Analysis**

\*Please note that this report encompasses the intended design of the model which may change to maximize user comfort.

**Description:**

The program will use a Newtonian object class to create circles on the screen as demanded by the user with various properties such as mass, velocity and direction. In turn, the program will calculate the varying degrees of attraction between the objects and act as a physics engine for gravity.

**Research:**

In order to create a visible editable object, the NewtonianObject class must extend the shape class already built into Flash, this way frame changes can still be listened for and the program will be able to draw shapes and store them into an array. Drawing and filling shapes is vital for user display (the class will have to be a shape with some properties such that an object of it can interact with others of the same type) so they must be well researched before the program is written. The inverse square law, conservation laws, and rigid body physics in two dimensions must be well understood also.

**Ideal Operation:**

-program requests input from user to draw a Newtonian Object on click

-program draws where user determined asking for velocity and direction as an option

-when user presses play simulation button all of the objects interact with one another as if gravity influenced them, their velocity and mass properties hold true

-when objects collide, they are forced in the opposite direction as a function of their respective masses (assumes an ideal conservation of force)

-(ASSUMING TIME PERMITS)user can press pause button to be able to edit traits or remove existing objects as well as add new ones

**Ideal Timeline (Objectives follow intended date of completion):**

-December 18th: Moving direction logic worked out for two objects on stage use trigonometry and directional vector logic

-December 21st: Simple UI implemented, begin working on array boundaries for three objects

-December 27th: Configure magnitudes to correspond to direction of objects

-January 4th: Fix ongoing errors

-January 10th: Implement more advanced user interface, allow user to manipulate certain properties on the stage in a friendly and practical way

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 direction

**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)

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