**Analysis**

This feature will allow planets to move according to the laws of physics as net out by sir isac newton in his theory of universal gravitation. It is important because it is the central mechanism around which all of this program is derived.

**Design**

Bodies will move in space as a result of their velocities (as given by ) which will be affected by the bodies acceleration (as given by ) which in turn will be affected by the gravitational force excreted on the body (as given by a=F/m and ) the universal gravitation constant (G) should be stored as an easily accessible variable as it will editable by the user in a future version.



**Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| Item for testing | Input value(s) | Expected result | Worked? |
| Function to change position according to velocity | non integral values (2 dp), vx+ vy+, vx+ vy-, vx- vy+, vx- vy- | Works fine and compounds position decimals, move to btm right, move to top right, move to btm left, move to top left |  |
| Function to change velocity based on acceleration | Negative, positive, non-integral | Works, works, works and compounds acceleration |  |
| Function to calculate gravitational attraction and overall force vector | Multiple random scenarios | Works fine and numbers check our |  |

**Documentation**

*Functions();*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Function | Purpose | how does it work | Inputs | return |
| applyVelocity | Alters a bodies x and y co-ordinates based on its velocity | Coordinate += velocity(coordinate) | Body object | NA |
| Apply acceleration | Alters a bodies velocity based on its acceleration as given by | Velocity(axis) += acceleration(axis) | Body object | NA |
| applyForceFromGravity | Calculates the force acting upon an object because of gravitational force as given by Sir Isaac Newtons universal law of gravitation | Calculates the overall force acting upon an object then splits it up into its components to be applied in the correct direction. It only calculates once for each force pair reducing the strain on resources | Body x,y,mass, | NA |

*Variables;*

|  |  |  |
| --- | --- | --- |
| variable | Purpose | type |
| Boddies.x | Stores the bodies x position | Float |
| Boddies.y | Stores the bodies y position | Float |
| Boddies.vX | Stores the velocity acting on a body along the X axis | Float |
| Boddies.vY | Stores the velocity acting on a body along the Y axis | Float |
| Boddies.aX | Stores the acceleration acting on a body along the X axis | Float |
| Boddies.aY | Stores the acceleration acting on a body along the Y axis | Float |
| Boddies.fX | Stores the force acting on a body along the X axis | Float |
| Boddies.fY | Stores the force acting on a body along the y axis | Float |
| Boddies.m | Gives each body its own mass | Float |
| Boddies.color | Gives each body its own colour which can be easily changed | String |
| F | Temporarily stores the resultant force vector acting on each body while its other attributes are being calculated | float |
| dMag | Temporarily stores the Pythagorean distance between two objects | float |
| thetaFromQuadLine | Temporarily stores the angle between a line between 2 objects and its quadrant line in radians | float |
| quadrant | Temporarily stores the trigonometric quadrant relating two bodies | integer |
| gravitationalConstant | Stores the universal gravitational constant of this universe increasing increases gravity and vice versa | float |

**Evaluation**

no major deviations occurred during the development of this feature however it did take an inordinate amount of time (likely due to a short design phase and stupid mistakes such as forgetting to change parts of code when copping it from other areas of the program). It should be noted that due to meaning that when r = 0 force is infinite (and hence acceleration and velocity) this however does not need much attention since planets when they occupy the same position in the future will employ collision mechanics.