

Exoplanet Tool Kit

New Mexico

Super Computing Challenge

Final Report

April 5, 2011

Team 35

Desert Academy

Team Members

Chris Brown

Isaac Fischer

Teacher

Jocelyn Comstock

Mentors

Henry Brown

Prakash Bhakta

Table of Contents

Executive Summary.....	2
Background	3
Astrophysics	3
Definition and Origin of Extraplanets	4
Methods of Detection.....	4
Project	6
Methodology.....	6
Data Results	7
Conclusion.....	10
Future Study.....	11
Acknowledgements.....	12
References	13
Appendix	14
Planet Generator Code	14
Planet Viewer Code.....	19

Executive Summary

Exoplanet studies are expanding with data from NASA's Kepler telescope. These exoplanet studies may provide answers to evolutionary planet formation. This project developed a processing/java application to perform orbital sequence models of exoplanet solar systems.

This project used several approaches to explore exoplanet data. A genetic algorithm generated possible solar system scenarios. These system scenarios were compared to actual Kepler data. The comparison allowed rules to be created to optimize the genetic algorithm. As more data is available, the genetic algorithm will be expanded to larger solar systems.

The advances in planetary data continue to build larger and larger data sets for analysis. In the course of a year, the number of planetary systems has moved from several to thousands. This original algorithm for modeling solar systems has successfully compared Kepler data and should provide valuable insights into solar system evolution.

Background

Since Galileo and the first telescopes, humans have gazed out at the stars dotting the universe. In the past 10 years these bright specs have begun to reveal their true natures and riches. In 1998 astronomers Bruce Campbell, G. A. H. Walker, and S. Yang took notice of the star Gamma Cephei wobbling around. Through their radial velocity measurements they predicted that there was a planet orbiting around the star. This observation was later confirmed. The first recognized exoplanet discovery was in early 1992 found by radio astronomers Aleksander Wolszczan and Dale Frail. They announced the discovery of planets around the pulsar PSR 1257+12. This discovery was confirmed and is considered to be the first example of an exoplanet. Since then the number has grown to what it is now, at the time of writing this report, 525 exoplanets. In the next few weeks NASA is supposed to release another 1000 confirmed exoplanets.

More of these new planets will be made visible by the recently launched Kepler orbital telescope. The Kepler project is a long term year round project with the goal of finding planets around other stars using the solar transit method as well as searching for stellar and cosmic anomalies. A brother telescope, called the James Webb space telescope (the successor to Hubble) is going to be launched in 2014. Once in orbit, The Kepler telescope will be used to find new planets and then Webb telescope will then be able to glimpse into the atmospheres of these newly discovered exoplanets.

Kepler's current mission is to continuously stare at more than 150,000 stars in a region of the constellation of Sigmas. Kepler is looking for one very specific sign of an exoplanet; this sign is both Kepler's greatest strength and its greatest weakness. The Kepler telescope is looking to detect solar transits. Unfortunately solar transits need to be detected multiple times in order to confirm an exoplanet's existence thus forcing Kepler to have to monitor the same stars for years. This means Kepler is biased towards finding large planets with very short orbital periods.

As important as discovering these worlds is, it's hard to find anything useful or anything at all. Currently it's like finding a sub atomic needle in a cosmologically scaled hay stack. To help us have better odds at finding these needles it would be helpful to know their properties and be able to predict where they might fall. This is the goal of our project – to come up with more ways to find exoplanets or, in other words, to better predict where to find exoplanets. The first step in being able to predict where a planet might be is to know where a planet comes from, as well as how solar systems interact such as how their planets grow and change. Thus, the following brief discussion on Astrophysics.

Astrophysics

To start off we are dealing only with solar systems for these examples. All solar systems have some basic elements; the most important and largest player in a solar system is the star. Stars usually form in stellar nurseries due to the abundance of gases, mainly hydrogen and helium. For some unknown reason these gases are slightly compressed from their normal states. From there, gravity takes over bringing the first amounts of gasses together. As this cloud of gas grows in mass, the rate at which the surrounding gases coalesce increases. Eventually a ball, on average 100 times the size of Jupiter, forms. At this point gravity is so strong that within the center of this ball of gas hydrogen ions begin to come

closer and closer together. In an instant the first hydrogen ions fuse and release a massive amount of energy. This energy causes a chain reaction forcing more ions to fuse and their energy causes even more ions to fuse. In a few seconds ignition is complete and a new star is formed.

Once the star is formed, it still has a cloud of gas and debris around it. Since the star is spinning, as a result of the fusion process, the particles around the star are pulled along for the ride. Over time centripetal force flattens these particles into a condensed disc. Again gravity works its magic -particles collide forming larger particles and those larger particles attract more matter growing larger. This cycle continues until the new object has cleared all the debris from and around its orbital path. Sometimes this object is large enough for gravity to squash it into a sphere and just like that a new planet is formed. In the process of cleaning out the protoplanetary disc around a young star the lighter elements escape gravity and are spun out to the outer edges of the disc. They then collect to form giant gas planets like Saturn and Jupiter.

After the planets have first formed, every solar system is completely unique in what happens next. In the case of our solar system, the large gas giants cleared out the remaining debris to form their many moons. Tidal forces between Jupiter and the inner planets tore apart a planet orbiting just beyond Mars forming our asteroid belt. A Mars sized planet was displaced from its orbit and sent on a collision course with our planet Earth to create our moon. In other solar systems these events could have taken completely different courses.

Even though each system is unique, some factors within the systems remain constant. The first and most prevalent of these factors is gravity. Gravity is one of the 4 main forces in the universe and is by far the one most responsible for things that happen at large scale, such as in the movement of stars and planets. For an object to orbit its star, two things must be occurring at once. First, the star must pull on the object and the object must be moving fast enough so that it is not sucked into the star. The same logic also dictates that if the star pulls on the planet the planet must also pull on the star. This relationship causes the planet to drag its star around with itself. Second, this causes the star to wobble which leaves a telltale sign on the star that allows for exoplanets to be detected using this relationship.

Definition and Origin of Exoplanet

Exoplanets are planetary bodies that orbit around other stars. Exoplanets can be any size from planetoids smaller than Pluto to planets that are many times the size of Jupiter. An exoplanet is any planetary body orbiting around another star. The word is relatively new since the first exoplanet was only confirmed 20 years ago. The roots come from Greek, “exo” meaning something outside or distant and “planet” meaning wanderer. Put together the direct translation is distant wanderer.

Methods of Detection

Exoplanets are found using 3 main methods. These methods are radial velocity, solar transiting, and pulsar timing. Each of these methods uses a different relationship between a planet and its stars. Radial velocity uses a relationship (described above) where a star pulls on a planet and a planet pulls on the star. As a star wobbles through space it casts light out across the universe. From an observer’s point of view the star would move horizontally and or vertically relative to the horizon. Unfortunately this does not give a 3 dimensional view of the star’s movement, for this reason a star’s light must be analyzed.

According to the Doppler effect, light coming at an observer faster than normal would be skewed toward the blue end of the light spectrum while light moving slower than normal would shift toward the red end of the light spectrum. Using this principal, as the star approaches the observer its light shifts slightly towards the blue and as it moves away from the observer its light shifts towards the red. The Doppler effect shows how quickly this shift occurs and thus the star's motion in the third axis can be calculated. This shows how fast the star is moving. Once how fast a star is moving is calculated, the mass, location, and orbital path of the planet can be calculated. The advantage of this method is that it can find planets normally undetectable by solar transits.

The next method requires a very special type of solar system. Pulsars are a solar remnant left over from a super nova. What makes them unique is that they spin and emit a beam of radio waves. This basically turns them into extremely precise lighthouses. Using the relationship that radio waves travel slower through matter than empty space, exoplanets can be found. Sometimes an exoplanet gets in the way of the spinning star's pulse. This causes a lag in the radio waves compared to the normal pulse sequence. The planets approximate position density and size can be determined from how long this lag is. This method is extremely rare and the results have high levels of inaccuracy.

The last and most common method is solar transit light curves. A planet orbiting a star sometimes crosses directly in front of the star and the earth. If one happens to be looking at this star as the planet crosses between us and its star we would notice the star's light dim. This is because the planet is blocking some of the light from getting to us. Sometimes a high resolution image can be taken which will show the planet's silhouette in front of its star. Most of the time, the instrumentation is not good enough to resolve an actual image; usually it is only sensitive enough to pick up the dip in light. When plotted on a straight line with zero being the normal light output of the star, this dip would show up as a curve. Measuring this curves size and magnitude the planets radius can be calculated. This method has a lot of downsides in it. Solar transits have the highest rate of false positives and can only provide the accurate radius of a planet. The other factors such as mass and orbit can be extrapolated but have high margins of error. This method has a high error rate because stars normally change in brightness levels and have solar flares sometimes large enough to dim the light like a planet would. This method also favors close solar orbiting large planets that can be confirmed quickly. The NASA Kepler telescope uses the transit method.

Our Project

Our project started off with the idea of a computer program that could calculate the positions of exoplanets from radial velocity measurements. My partner and I quickly realized neither of us had advanced enough math skills to accomplish such a task. Based upon this fact, we pattered around looking for mentors who could help us with the math. Unfortunately all of them said it was too far over our heads and tried to dumb it down enough. However, a math level less than calculus was inadequate for our needs. So we came up with the next idea, a star sorter. This was going to be used in the larger project as an Kepler data checker and data import element. We have been working with the star sorter, and for the most part, have this working. We await more Kepler data on stars to test the star sorter.

We have also been working on a solar system generator as another Kepler data checker. Other projects are experiments with ways to view the data and to graph it.

The Methodology

The Calculator

This original idea started the project and the eventual goal. Taking radial velocity information about a star, and using the gravitational formula $f = G (m_1 * m_2 / r)$, find the mass, orbital parameters and possibly anomalies in the orbit which could indicate the presence of smaller or other planets.

The Sorter

The star sorter is a processing language program. It serves two purposes a data importer that will eventually pass the data along to processing programs and an error checker. The data moving part is pretty self explanatory. The star sorter is a Kepler data checker comparing data from Kepler. This program would check for errors in classifying the star data. This tool correlates star properties and solar system properties. The Kepler data set of these stellar properties can be sorted into small groups and can be analyzed to look for patterns. The star sorter is used to compare a star's class to see if the star fits its class. These patterns can also be fed into the system generator to help improve the gene algorithm's rule sets. This star sorter allowed us to learn the basics of processing code and data structures involved.

The Generator

A solar system generator based on a genetic algorithm in processing language. Genetic algorithms are pieces of code that breed the best solution to a problem aka it evolves to solve a problem. Probably the best solution to modeling solar systems because solar systems don't just appear they themselves grows and changes over time. So a program that adapts in similar ways is an ideal solution. Genetic algorithms use rules to prune off their growth; we hope that the sorter and continuing growth in raw datasets will allow for patterns to be found that can be translated into rules to improve the generators accuracy.

Gravity Grid

A three dimensional processing application that we are looking into using for advanced simulation of solar system as well as a possible way to view outputted data.

Planet viewer

A processing program meant to output our data in a similar way to other data sets so that they can be compared visually.

The programming code (language and algorithm):

Processing Language

Processing is a Graphical Java interface developed by Ben Fry at MIT. The GA library comes within the processing 1.2. freeware distribution.

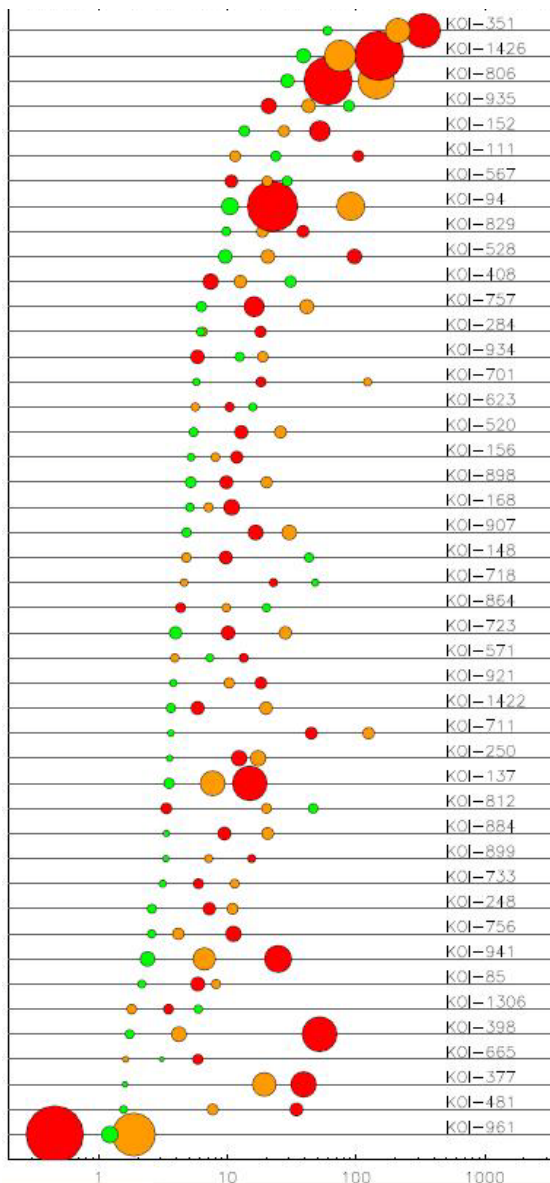
Genetic Algorithm Class (GA) by Aaron Steed

The genetic algorithm hosts an array list of chromosomes. The chromosomes contain an integer array called "dna". The dna values can represent numbers, an index to objects, alphabetical characters or traits.

Chromosome Class

This is an internal component of the genetic algorithm class used to come up with random ideas, solutions to problems, creatures, etc.

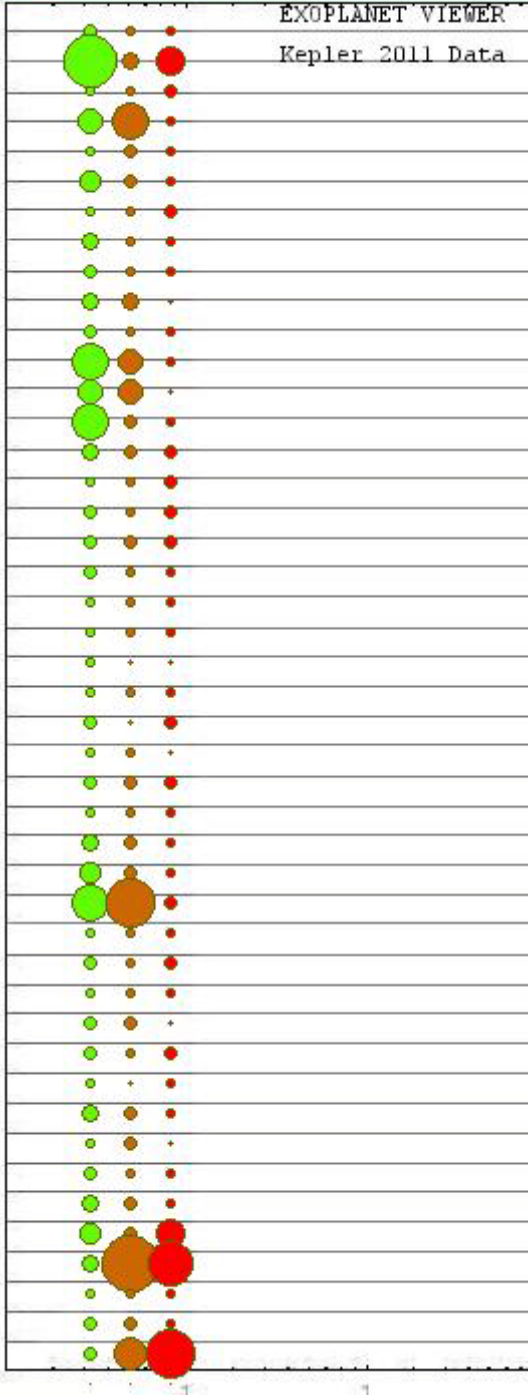
(Actual Code beneath closing statements)



Genetic algorithm:

From our analysis of the Kepler data we have found one pattern that holds constant most of the time the time. As shown by the graph below 34 out of 45 solar systems have the smallest planet in the system having the shortest orbital period and as such the closest orbit of any planet in their system. Using this data we created the system generator so that it builds triplet solar systems similar to what is shown in the graph. To compare generated solar systems, we use the same unit of Kepler radii as data from this graph, which was used to create multiple models of exosolar systems. By comparing these sets of generated data to real data we hope to find the differences between real and generated. Hopefully the differences can be turned in to rules common to all exosolar systems. This would further serve to improve the generator. We acknowledge that the current datasets contain too few solar systems to model an average system from. We await the release of more data from Kepler to improve the governing rule sets of the model. The only realistic hard set rule we currently have is the average total radii of these 45 triplet systems. The average system total radii is 9.6 which for our purposes

is rounded to 10 radii. Our genetic algorithm uses a target rule of the 10 radii to tune the outcome of the planets. The two graphs below compare Kepler 2011 data for 45 solar systems against our genetic algorithm generator data for 45 solar systems.

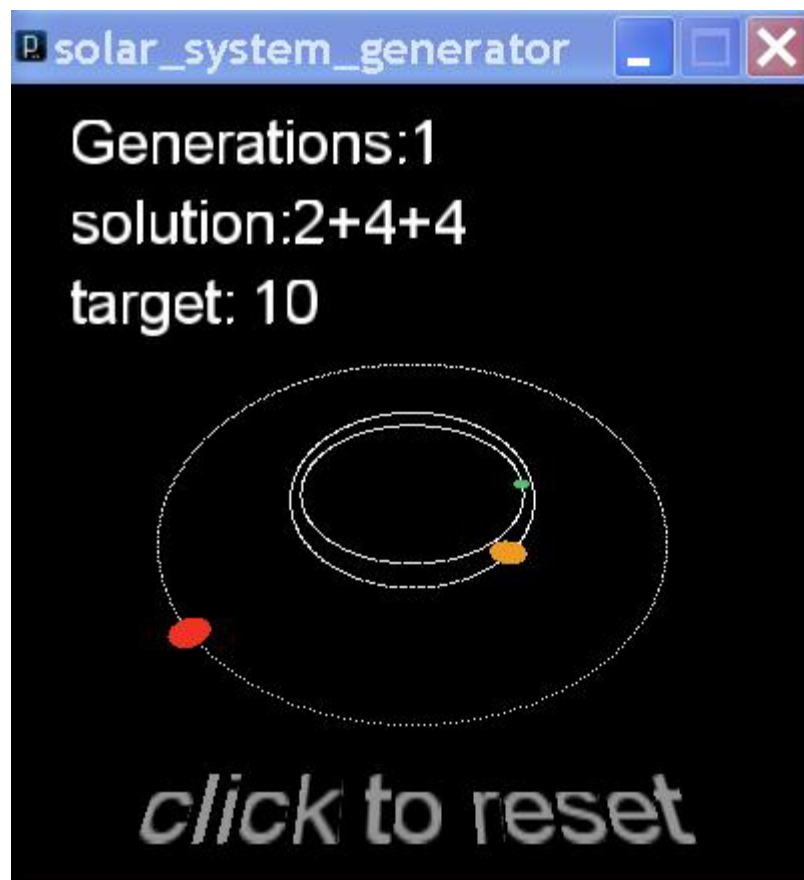


3

As you can see the system viewer still has room for improvement. It currently serves the purpose that it was intended to be used for in this stage of development. It currently shows the data in a way that is comparable to the original data. In the near future we hope to better organize the data as well as clean up the color coding of the planets. Currently as is shown the generator isn't much better than a random number generator this is because the rule to tell it to have a higher chance to place the smallest planet first is still not functioning correctly.

Solar System Generator Interface

The generator interface shows a solar system and the vital statistics used in creating the solar system. The interface shows the number of generations and the target 10 for the genetic algorithm. The genetic algorithm uses bit strings to hold a pattern for breeding. It uses these bit strings to hold potential values for recombination. The string will eventually combine numbers until the 10 target is reached. The solution holds the final bit string. A representation of the solar system is shown below the genetic algorithm's result.



Conclusion

Problems with the Kepler data lead to problems with our genetic algorithm solar system generator (GASSG). By adding together 3 planet radii we have found a rule for 10 radii for 3 planets. The first planet radii is smaller than the other two planets radii most of the time. However exceptions to both of these rules exist in the Kepler data and may be a significant factor even with the addition of more data.

As larger solar systems arrive the GASSG should be easily modified for 4-12 planets. With this information a much more precise model of exosolar systems should evolve with better rules for the genetic algorithm. The application of a genetic algorithm solar system model should become more instructive in producing rules and thus principles about the evolution of all solar systems.

Future Study

We hope to improve the genetic algorithm rule set by comparing larger numbers of solar systems with greater numbers of planets. We hope to add gravity grid and possibly other 3 dimensional simulators and viewers.

Hopefully this and other work into the field of exoplanets and the universe will lead to insights that will allow us to find another world like our own.

Acknowledgements

We would like to thank the following people for their assistance with this project.

My Advisor

Joycelin Comstock of Desert Academy

My mentors

Henry Brown of DOIT State of New Mexico

Prakash Bhakta of DOIT State of New Mexico

References

Lissauer, J. , et al, Architecture and dynamics of Kepler's candidate multiple transiting planet systems. arXiv:1102.0543v3 [astro-ph.EP] , Feb 25 2011

Charbonneau, P. , Genetic Algorithms in Astronomy and Astrophysics, The Astrophysical Journal Supplement Series,101:309-334,1995 December

GAMath - Aaron Steed 2008

Graphic Solar System - Ira Greenburg 2006

Software compiler <http://processing.org>

Software tutorial <http://processing.org/learning/index.html>

Internet articles

http://en.wikipedia.org/wiki/Extrasolar_planet

http://nsted.ipac.caltech.edu/cgi-bin/bgServices/nph-bgExec?bgApp=/Sieve/nph-sieve&mission=NStED&uniqueid¤tForm=goQuery&list_Multiplicity_uniqueid&listpath&frommu

<http://kepler.nasa.gov/>

<http://www.jwst.nasa.gov/>

http://en.wikipedia.org/wiki/Methods_of_detecting_extrasolar_planets

<http://www.space.com/>

http://en.wikipedia.org/wiki/Star_formation

http://en.wikipedia.org/wiki/Nebular_hypothesis

<http://www.enchantedlearning.com/subjects/astronomy/stars/startypes.shtml>

http://www.sciencedaily.com/news/space_time/extrasolar_planets/

http://en.wikipedia.org/wiki/Stellar_classification

http://en.wikipedia.org/wiki/Protoplanetary_disk

http://en.wikipedia.org/wiki/Solar_System

http://www.google.com/search?hl=en&defl=en&q=define:Extrasolar+planets&sa=X&ei=JtybTe_CO5DQsAPSi-2cBA&ved=0CBcQkAE

Note: Keep in mind these wiki articles are highly cited and are condensed information. Wiki articles cover what needs to be said without having hundreds of separate links.

Appendix

PLANET GENERATOR

```
import ai_ga.*;

//Software compiler http://processing.org
//Software tutorial http://processing.org/learning/index.html

//Code Sources:
// GAMath a simple program that finds collections of numbers that add up to a given value
// GAMath - Aaron Steed 2008
// Graphic Solar System - Ira Greenburg 2006

// Simple program to simulate planets in a solar system
// Chris Brown 2011

PrintWriter output;
PFont font;
Sum sum;
int initialTarget = 10;
String solution = "";
String target = "target: " + initialTarget;
Planet p1, p2, p3, p4, p5;
int gp0, gp1, gp2;

void setup(){
  size(400, 400, P3D);
  //File Handling Code
  // Create a new file in the sketch directory
  output = createWriter("ga_planets.txt");

  // initialise the GA with a number to pack other numbers into
  sum = new Sum(initialTarget);
  font = loadFont("ArialMT-30.vlw");
  textFont(font, 30);
}

void draw(){

  background(0);
```

```

fill(255);
text("Generations:" + sum.generation, 30, 40);
text(solution, 30, 80);
text(target, 30, 120);
if(sum.found){
    Chromosome temp = (Chromosome)sum.solutions.get(0);
    solution = "solution:" + sum.decode(temp);
    println(sum.decode(temp));
    String[] list = split(sum.decode(temp), "+");
    int[] nums = int(list);
    p1 = new Planet(0, nums[0]*2, #44cc66);
    p2 = new Planet(45,nums[1]*2, #EE9922);
    p3 = new Planet(135,nums[2]*2, #ef3323);
    gp0= nums[0];
    gp1= nums[1];
    gp2= nums[2];

    //background(0);
    translate(width/2, height/2);
    rotateX(radians(55));
    p1.orbit(55, 55+(gp0*2), 2.4, true);
    p2.orbit(60, 66+(gp1*2), 1.7, true);
    p3.orbit(120, 132+(gp2*2), 1.2, true);
}
else{
    sum.propagate();
}
fill(150);
text("click to reset", -80, 180);
}

```

// when the user clicks on the applet, restart the GA

```

void mousePressed(){
    int rand = (int)random(20,40);
    sum = new Sum(rand);

    target = "target: "+10;
    solution = "";

    if((gp0 > 0) && (gp1 > 0) && (gp2 > 0))
    {
        output.print(gp0);
    }
}

```



```

output.print(" ");
output.print(gp1);
output.print(" ");
output.println(gp2);
}

}

void keyPressed() {
    output.flush(); // Writes the remaining data to the file
    output.close(); // Finishes the file
    exit(); // Stops the program
}

// This extension of GeneticAlgorithm searches for a combination of 3 numbers that add up to a target
value

class Sum extends GeneticAlgorithm{

    // Variables specific to your task can go here

    int target;

    Sum(int target){

        // The first line must be a call to GeneticAlgorithm's constructor
        // The constructor defines (dnaLength, valueLength, poolSize) and initialises default variables
        // If you are using the method propagate() the poolSize must be divisible by 2
        // This algorithm is set to split dna at the binary level, experiment to see if you need this feature

        super(3, 10, 10, true);

        // Update your variables here
        // This is also the place to modify the GeneticAlgorithm's fields

        //this.target = target;
        this.target = 10;

    }

    // A method overriding scoreFitness must be defined for this task.
    // The better a Chromosome's dna is at doing it's job, the lower it's fitness should be
    // fitness should be a value ranging from 0 (perfect) to 1 (useless)

```

```
// if you don't override scoreFitness, it will always return 1
// In the following example the values in dna are being added together to see if they match a target
```

```
float scoreFitness(Chromosome o){
    int total = 0;
    for(int i = 0; i < dnaLength; i++){
        if(o.dna[i] < valueLength){
            total += o.dna[i];
        }
    }
    return (1.0 / target) * (target - total);
}
```

```
// If your dna values are returning an index in an ArrayList of objects
// or happen to be char values, you may want to put a decode() method here
// This method returns a string explaining how the dna was added up
```

```
String decode(Chromosome o){
    int total = 0;
    StringBuffer answer = new StringBuffer();
    for(int i = 0; i < dnaLength; i++){
        if(o.dna[i] < valueLength){
            total += o.dna[i];
            answer.append(o.dna[i]);
            if(i < o.dna.length-1){
                answer.append('+');
            }
        }
    }
    return answer.toString();
}
```

```
}
```

```
//Planet class for planet size, placement, position
```

```
class Planet{
    float planetOffset;
    float planetRadius;
    color surface;
    float angle;
    boolean isPathVisible;

    Planet(float planetOffset, float planetRadius, color    surface){
        angle = planetOffset;
    }
}
```

```

    this.planetRadius = planetRadius;
    this.surface = surface;
}

void orbit(float ellipticalRadiusX, float ellipticalRadiusY,
          float orbitSpeed, boolean isPathVisible){
// draw ellipse first, so it's under the planet
if (isPathVisible){
    drawOrbit(ellipticalRadiusX, ellipticalRadiusY);
}
float px = cos(radians(angle))*ellipticalRadiusX;
float py = sin(radians(angle))*ellipticalRadiusY;
fill(surface);
noStroke();
ellipse(px, py, planetRadius*2, planetRadius*2);
angle+=orbitSpeed;
}
void drawOrbit(float ellipticalRadiusX, float ellipticalRadiusY)
{
stroke(255, 50);
float angle=0;
for (int i=0; i<360; i++){
point(cos(radians(angle))*ellipticalRadiusX,
      sin(radians(angle++))*ellipticalRadiusY);
}
}
}

```

PLANET VIEWER

/*

@ Author Chris Brown - 3/10/11

Software compiler <http://processing.org>

Software tutorial <http://processing.org/learning/index.html>

Planet Viewer - Show relative size of planets in a Solar System

*/

/*Data structures and global variables*/

PImage img;

String[] files;

int nextimg;

int totimg=1;

int x=5;

int z=5;

int j=1;

int i=1;

int[][] kepler_data = new int[46][3];

int kepler_num=8;

int flag =1;

int kepler=9;

PFont font;

int num;

int MAX = 50;

/*Setup function loads processing images and data*/

void setup()

{

size(700,700);

files=new String[totimg]; //some example images.

files[0]="planet_grid.JPG";

nextimg=0;

img=loadImage(files[nextimg]);

nextimg=(nextimg+1)%files.length;

num=9;

int hsize = 10;

loadString(); /* loading the planet data */

}

```

/*Draw function creates Windows and controls*/
void draw()
{

/* Display Planet Map image */
background(0);
image(img,0,0,width,height);
int v=1;
int h=25;
for(int i=0; i<46; i++)
{
v= v + 15; // Stagger vertical on chart
h=25;
for(int j=0; j<3; j++)
{
//h= h + 10; // Stagger horizontal on chart
h = h+ 20;
if (j==0)
{
fill(100,250,0); //green -1st Planet
}
if (j==1)
{
fill(204,102,0); //orange -2nd Planet
}
if (j==2)
{
fill(250,0,0); //red -3rd Planet
}
//Print a planet to the screen
ellipse(h,v,kepler_data[i][j]*2,kepler_data[i][j]*2);
}
}
delay(1); //this decides how long an image should be displayed

/*Display titles*/
//Titles- Black
color cl = color(102,102,0);
stroke(cl);
font = loadFont("CourierNew-12.vlw");
textFont(font);
fill(0,0,0); //Black
text("EXOPLANET VIEWER",140,12);

```

```

//text("Kepler 2011 Data", 140,32);
text("Generator Data", 140,32);

}
/* File loading function */
void loadString()
{

String name = "all_ga_planets_10_45"; //Generator Data Integer
//String name = "3_col_multiplanet_round"; //Kepler 2011 Data Space

String[]t=loadStrings(name+".txt");
for(int i=0; i<t.length; i++)
{
String[]rowData=split(t[i], " ");
for(int j=0; j<rowData.length; j++)
{
kepler_data[i][j] = Integer.parseInt(rowData[j]);
}
}
}
}

```