# Solar System Formation:
# How Common is a Planet Like Earth?

Sandia Preparatory School

Team Number: 128

Area of Science: Astrophysics

Programing Language: Python

Team Members:
Cal Boye-Lynn; 9th; 21cboye-lynn@sandiaprep.org
Noah Peterson; 9th; 21npeterson@sandiaprep.org
Parker Willis; 9th; 21pwillis@sandiaprep.org

Sponsoring Teacher and Project Mentor:
Neal Holtschulte; nholtschulte@sandiaprep.org

Table of Contents:

## **Project Summary:**

Earlier this year, our group decided from our class-created list of possible topics to investigate our solar system. Our solar system, like our universe as a whole, is highly complex, but governed by the same, common rules that we all are bound by which means that, in short, simulating a small section of the galactic empire-to-be (we're nerds, okay) is the perfect job for a computer. Give it a set of starting conditions, feed it rules and vóila your solar system begins to form. So we set out to find out how best to simulate the formation of a solar system.

Two weeks later, we realized our mistake. Solar systems, by their nature, are VERY complex. You have objects colliding, joining, and falling into the star. Radiation blowing out from the primitive sun like a scourging tidal wave, sweeping gas and dust out into the distance. Planets and comets are flying past the sun at massive speeds. Orbits, ellipses, clouds, gravity wells and, for some reason, the cursed collision code, are all problems that we'll have to figure out and overcome. And so far, we've gone a fair ways.

We've made planetesimals fly past and around a primitive sun, written code to display "Goldilocks" Zones, a location where a planet is capable of having liquid water and, thus, life, with the input of a variable and we've got [put something here]. But we've got a lot more to do before we're satisfied.

We're trying to make these pre-planet particles collide in space, either merging or colliding, a HUD (Heads Up Display) to see the data behind an object in real time with a simple click and turn our small, two dimensional screen into a full, three-dimensional display of the simulation going on behind the scenes.

**Problem Statement:**
Describe why the topic you investigate is relevant. Describe what the topic is about. Provide background information.

Someday Earth will die. It will be scorched by powerful solar winds as our star, the Earth's nurturing mother, expands into the inner planets, killing anything left on our home. But we cannot be alone. If we exist then there must be other planets like our green jewel and, probably, they do. Small, rocky planets much like our own with gaseous atmospheres have been observed but, with no way to see them ourselves, our only way of understanding

**Methods:**
We began with what might appear to be a rather simple game much like the board game Hungry Hungry Hippos", where you, a blob, tried to gather up smaller blobs to grow and became the largest blob of them all. From these humble beginnings, we used the blob mechanics we knew and, with aid from our professor, built a simple program where a bob would orbit another blob. From there, we complexified the model, creating a gravity-like force that acted on every object.

We used Python and its add-on, Pygame, for the visuals.

**Results and Conclusions:**
We will freely admit that we ran out of time. Our goal was far too complex for us to complete in the time provided. However, when testing there was a pattern that emerged. Not one planet ever landed entirely in the habitable zone. This may have been because the scale of the objects in our code was off or that the starting position of the model was inaccurate but, regardless, we had few to none applicable objects in a stable orbit. Earth is a very unique planet, but there are infinite possibilities and we have not even come close to that number.

Survival is probably the greatest achievement of our project. But, in all seriousness, we are most proud of the effort we put in, despite the fact that most of it appears to have been wasted.

If we continued to work on this project the first features we would add would be physics based collisions, scale the model solar system, and make the visual representation of the program 3D. The physics based collisions would create a more realistic solar system and give us more accurate data to the possibility of another Earth like planet. Scaling the solar system would also give us more realistic and accurate data. Finally, making the program 3D would allow the particles to move in another direction giving us more realistic data.

## **Miscellaneous Articles:**

Collisions, and Why They Aren't Currently Present

As we reached the middle stage of our project, we decided to try and add collisions to make the simulation more accurate. In order to do so, while not neglecting other projects, one of our members, Cal, began to work on the collision code on his own. As it turned out, the calculations required for accurate collisions were far outside of any of the group's mathematical repertoire, making it impossible (as of the current) for the code to function.

Despite this, we soldiered on, the code passing from Cal to other members before, eventually, dying on the cutting room floor.

Of the iterations we had devised, none worked. The first and second iterations, making no sense and being uncommented, were quickly scrapped. The third version, which was the last non-elastic collision code created, repeatedly failed to actually integrate into the rest of the code. In explanation, from what we could gather the code simply failed to implement changes for an unknown reason.

From here we decided to try and create a testing area in order to test the code in a one-on-one collision setting, but the errors persisted. So, unable to actually produce working code, our priorities have shifted to other aspects of the code.

Had the code functioned properly, what would have happened is, given the angle and velocity of the impact, the objects involved would break apart, merge together or spray dust into the aether.

## **Acknowledgements:**

An acknowledgment of the people and organizations that helped you.

Our Programming 1 teacher, Neal Holtschulte, helped us by teaching us the methods we used in our code. Websites such as https://sites.google.com/site/3dprogramminginpython/ and ____(the other website we used at the beginning of the project for something idk). Finally, Parker Willis' step father helped fix bugs and is helping in the effort to make the model three dimensional.