Brooke J. Leinberger

7/16/21

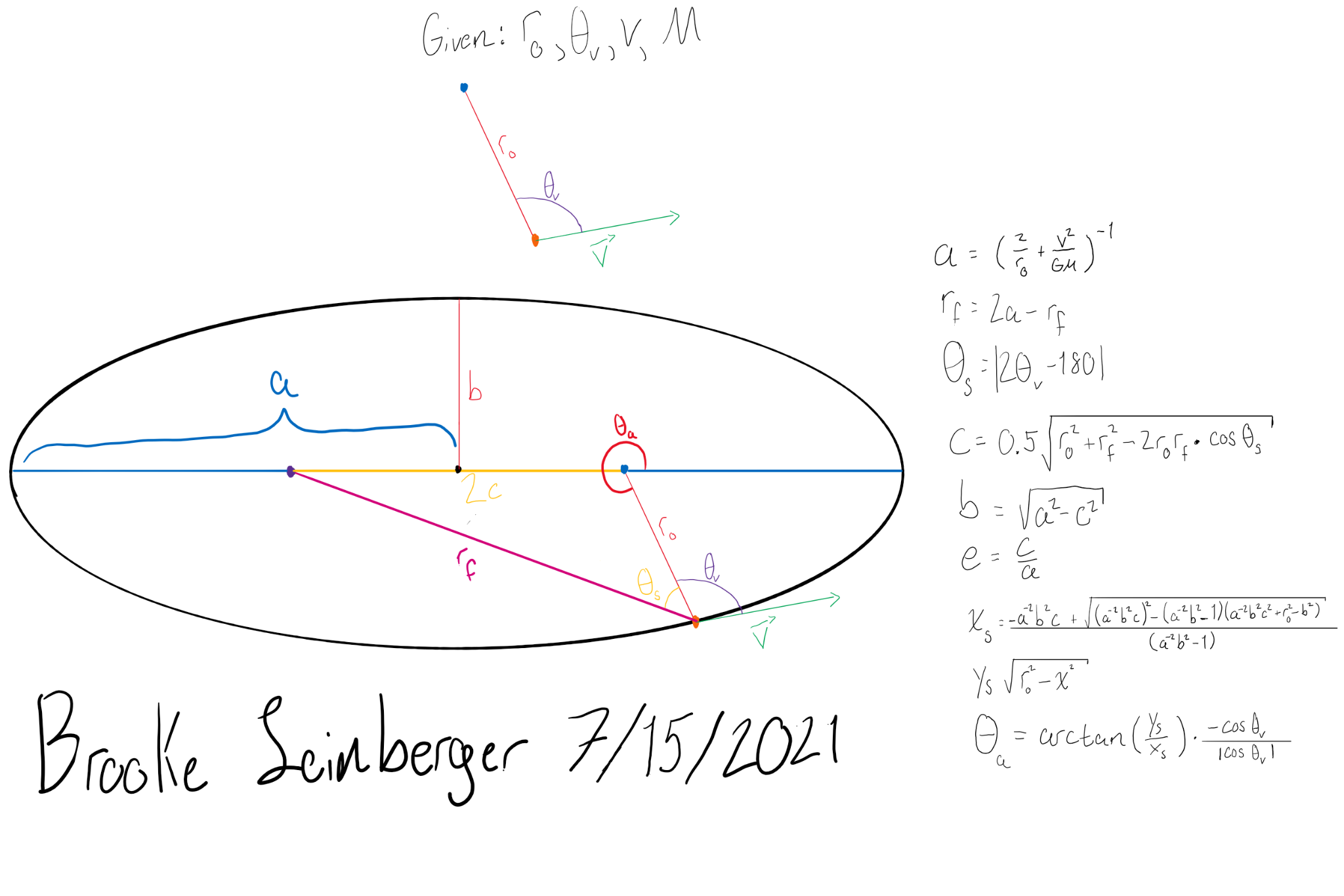
A few weeks ago, just after starting my job, I repurchased an audible subscription to entertain myself while I did repetitive warehouse work. Soon after I found a book called “Apollo” which was a 20 hour audiobook about the events of Nasa’s Apollo program. It’s a very interesting book and I’d encourage anybody interested in manned space flight to give it a listen! Just last week, about 2-3 weeks after finishing the book, it inspired me to dive back into an idea I had last summer after watching a video of deep-space Apollo navigation - A space sim not unlike Kerbal Space Program. So during the past week, my hyper-fixated mind found its target on this project. It’s been all I can think about since. I am very proud to say today, July 16th, 2021 at 20:00 EDT that I have ***FINALLY*** worked out the math, simulated it in desmos, and have it coded in C# on my laptop (I will have a link to the Desmos file at the bottom of this log).

I am very proud (and frustrated) at the fact that most of the resources I found for this project either didn’t have enough information/application or went a bit over my head with matrices and such. I went down many avenues that didn’t exactly pan out but that I might become useful in future, like finding polar equations for some of the shapes graphed. The only equations that I found usable in my research was the vis-viva equation (to relate velocity, planetary mass, and orbital distance to a critical orbital parameter: the semi major axis) and the equation of an ellipse, which is the shape of every orbit according to Kepler’s first law of planetary motion.

One revelation I discovered that I’m personally especially proud of is: *the angle (A) formed between a line drawn by one of the foci of the ellipse to a point on the ellipse, and the tangent of the ellipse at that point,* ***is equal to the angle (B) formed by a line drawn from the other focus to the same point, and the same tangent of the ellipse at that point.*** (It’s easier to see this in the below diagram (not to scale) as the angle formed between rf and v is equal to the angle formed between ro and v). I was elated to notice this, as this unlocked the last necessary key to charting the rest of the ellipse with the limited variables the games would use: velocity, angle between planet and velocity vector, distance to planet’s center of mass, and mass of the planet. This ellipse would then represent a 2D model of the orbit of the satellite, which is the minimum requirement when determining if you are in a safe orbit, or if you’re going to crash into the surface. You can even find your current position on the ellipse by graphing a circle with a radius of the distance between the satellite and the center of the planet over the focus representing the planet. This leaves two intersection points, which you can reduce to one depending on whether the angle between planet and velocity vector is obtuse or acute. With this working model, I can finally start work in Unity to bring this model to life.

I have a couple ideas which this idea can evolve to: A KSP / Outer Wilds like simulation game about flying around the solar system, visiting space stations, with realistic procedures, and/or a *Spacewar!* Inspired dog fighting game, with a heavier focus on orbital mechanics.

Desmos simulation: <https://www.desmos.com/calculator/sub5z9dw83>

Geometric diagram: ******