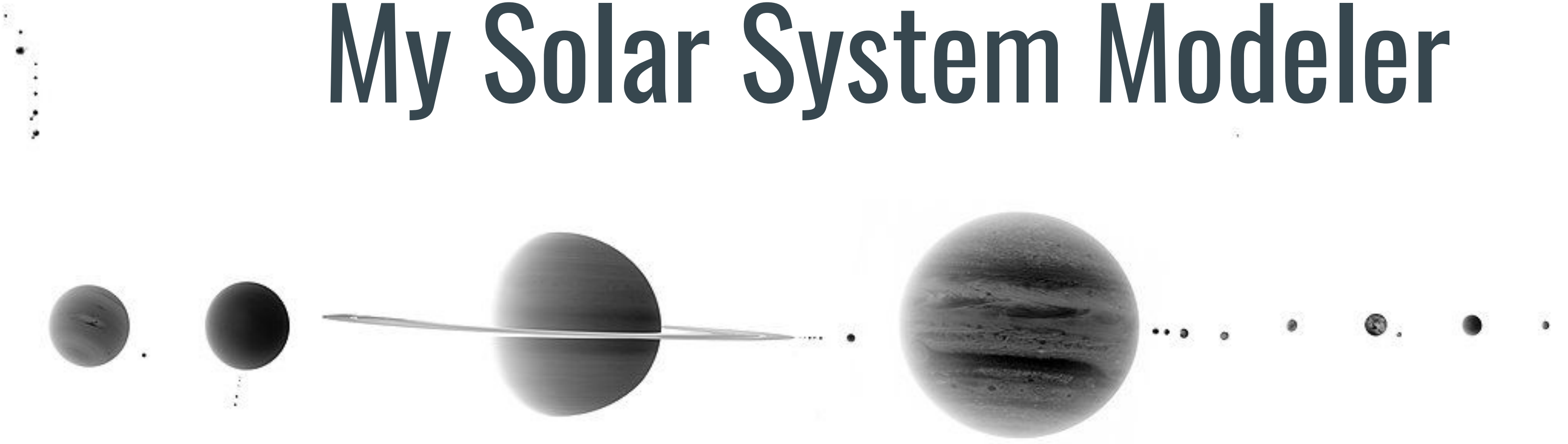
My Solar System Modeler



Senior Capstone Presentation
Jadon Koegel

System Goals

What I was trying to do.

Learning tool first and foremost.

① Accurate modeling of Newtonian orbital dynamics.

② Modularity of as many astrophysical variables as possible.

③ Realistic visual representations.

Expectations of a Physical Model

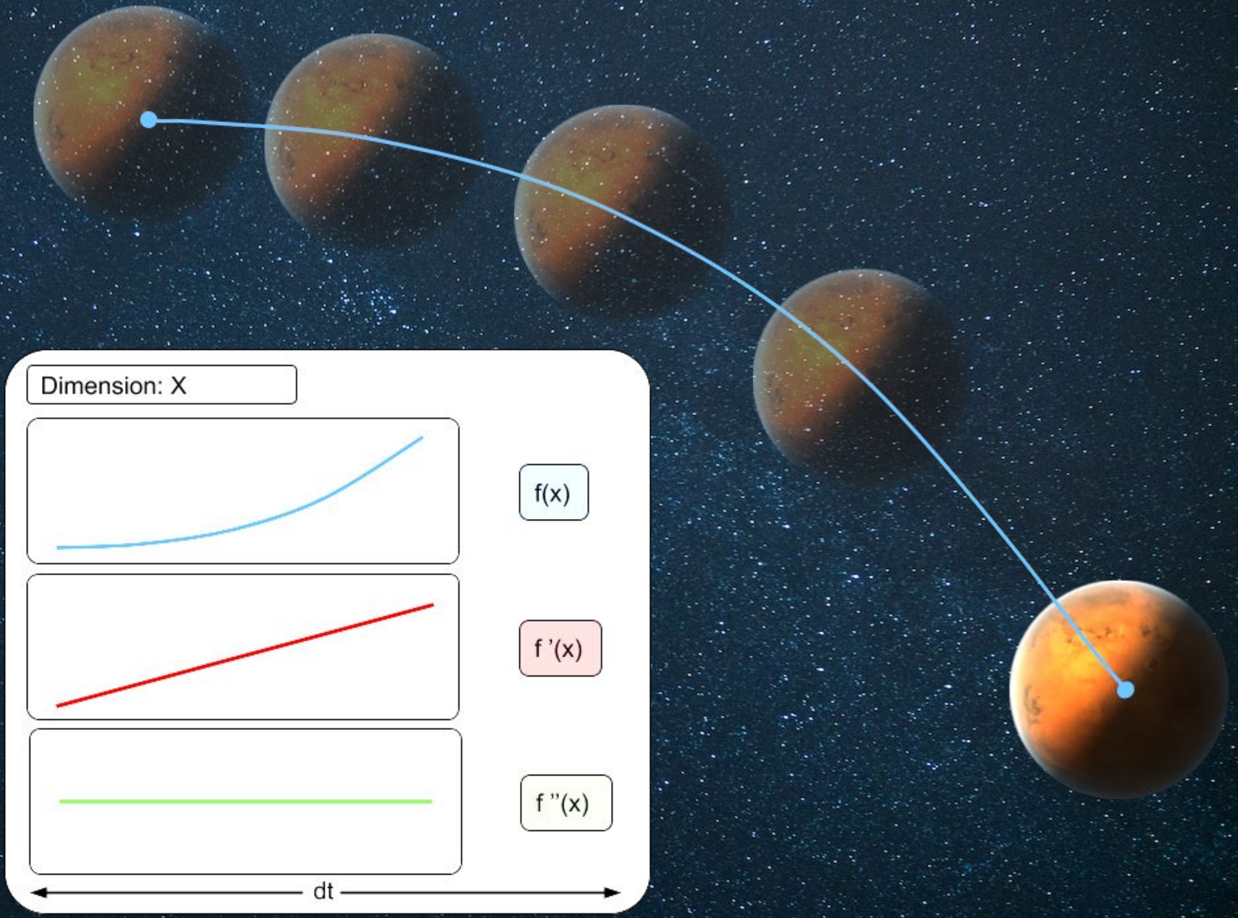
This is a simulation not an animation.

- Physical fidelity requires mathematical accuracy.
- Iterative models require high precision.
    - Small errors lead to significant deviations over time..
- Physical Theory → Computation Modeling → Coding Execution.

①

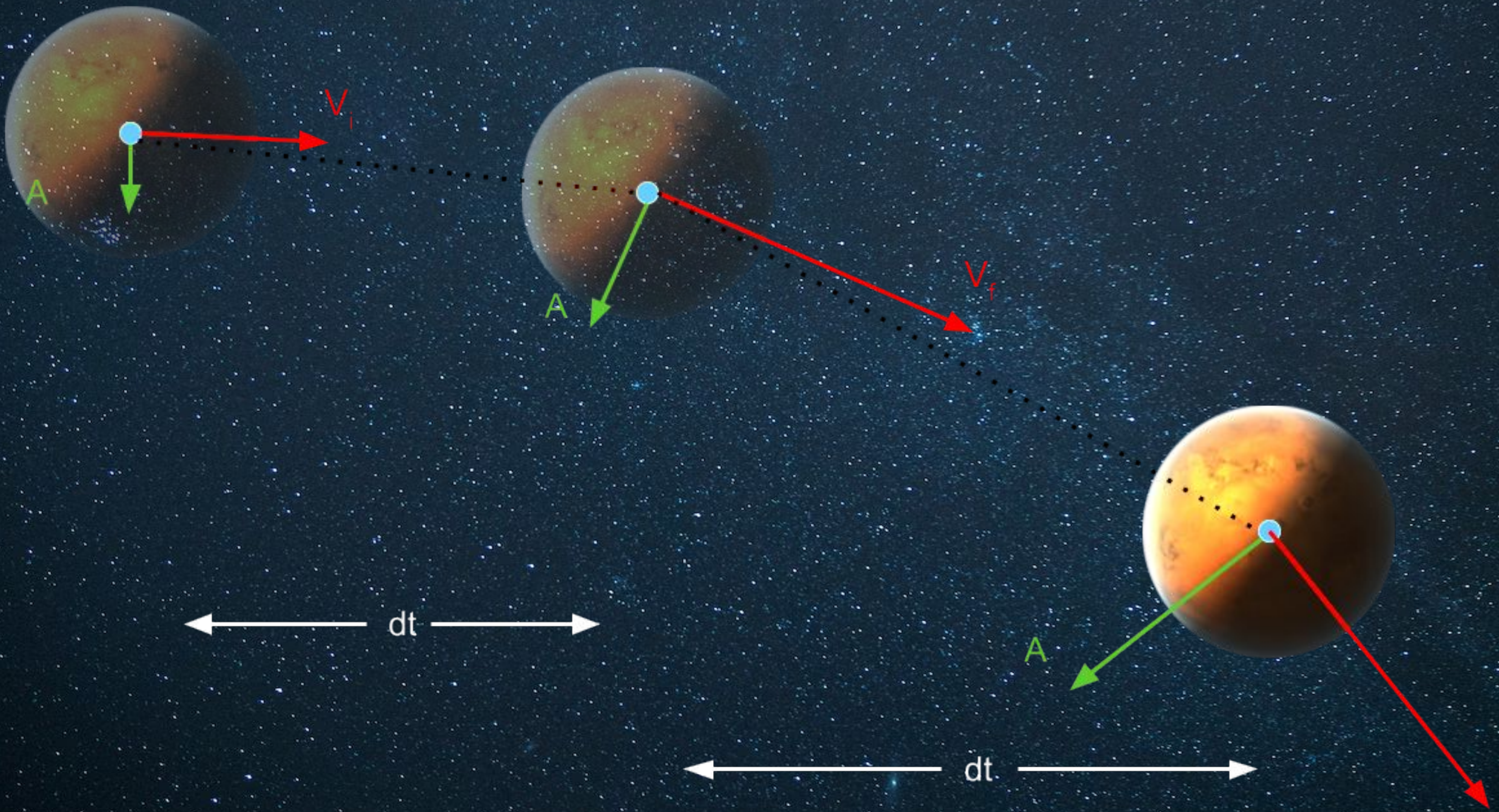
Orbital Mechanics

Calculus who?



Orbital Mechanics

Extrapolating from the bare minimum



Orbital Mechanics

①

*All movement is in reference to (0, 0, 0)

$$g = \frac{GM_{\odot}}{r^2}$$

$$r = \sqrt{x^2 + y^2}$$

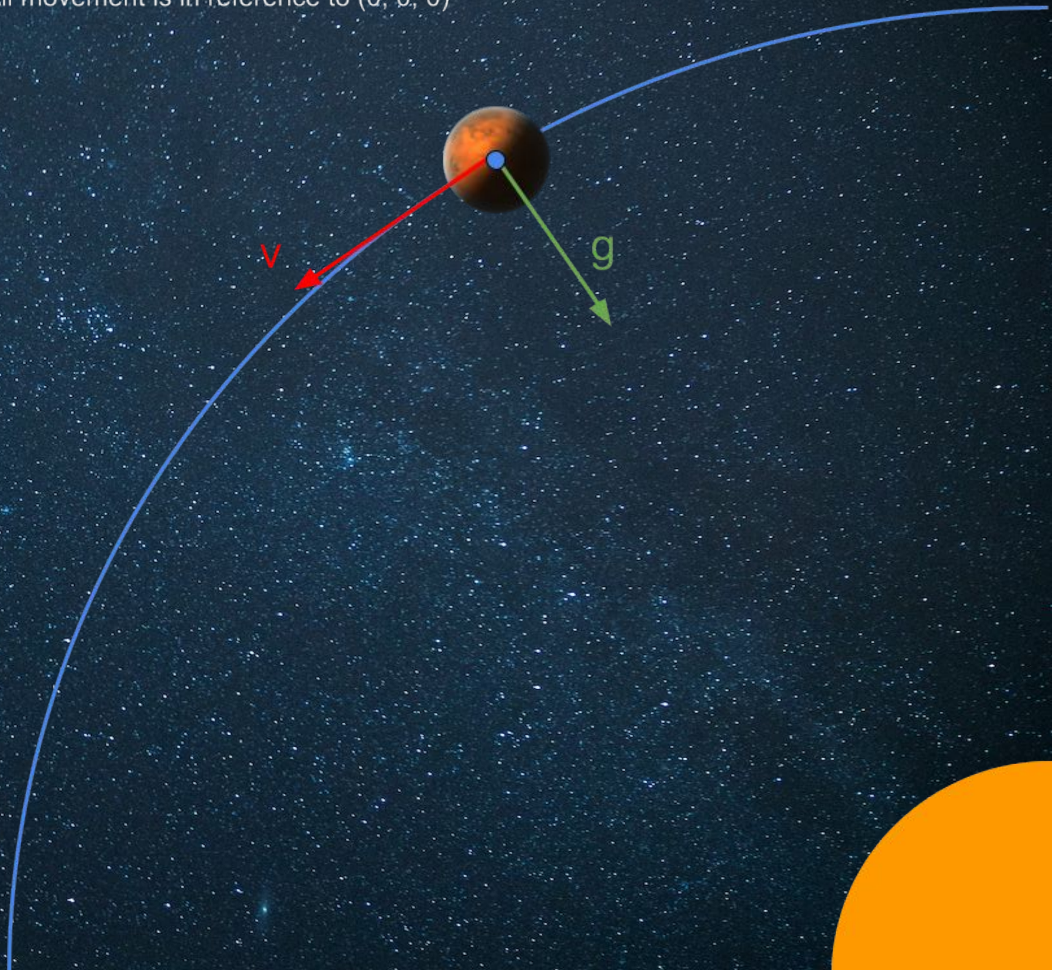
$$g = \frac{GM_{\odot}}{\left(\sqrt{x^2 + y^2}\right)^2}$$

$$\cos\theta = \frac{g_x}{g}$$

$$g_x = \frac{GM_{\odot}}{(x^2+y^2)}\cos\theta = \frac{GM_{\odot}}{(x^2+y^2)}\frac{x}{\sqrt{(x^2+y^2)}}$$

$$g_x = \frac{GM_{\odot}x}{(x^2+y^2)^{\frac{3}{2}}}$$

$$g_y = \frac{GM_{\odot}}{(x^2+y^2)}\sin\theta = \frac{GM_{\odot}y}{(x^2+y^2)^{\frac{3}{2}}}$$



Orbital Mechanics

$$g = \frac{GM_{\odot}}{r^2}$$

$$r = \sqrt{x^2 + y^2}$$

$$g = \frac{GM_{\odot}}{\left(\sqrt{x^2 + y^2}\right)^2}$$

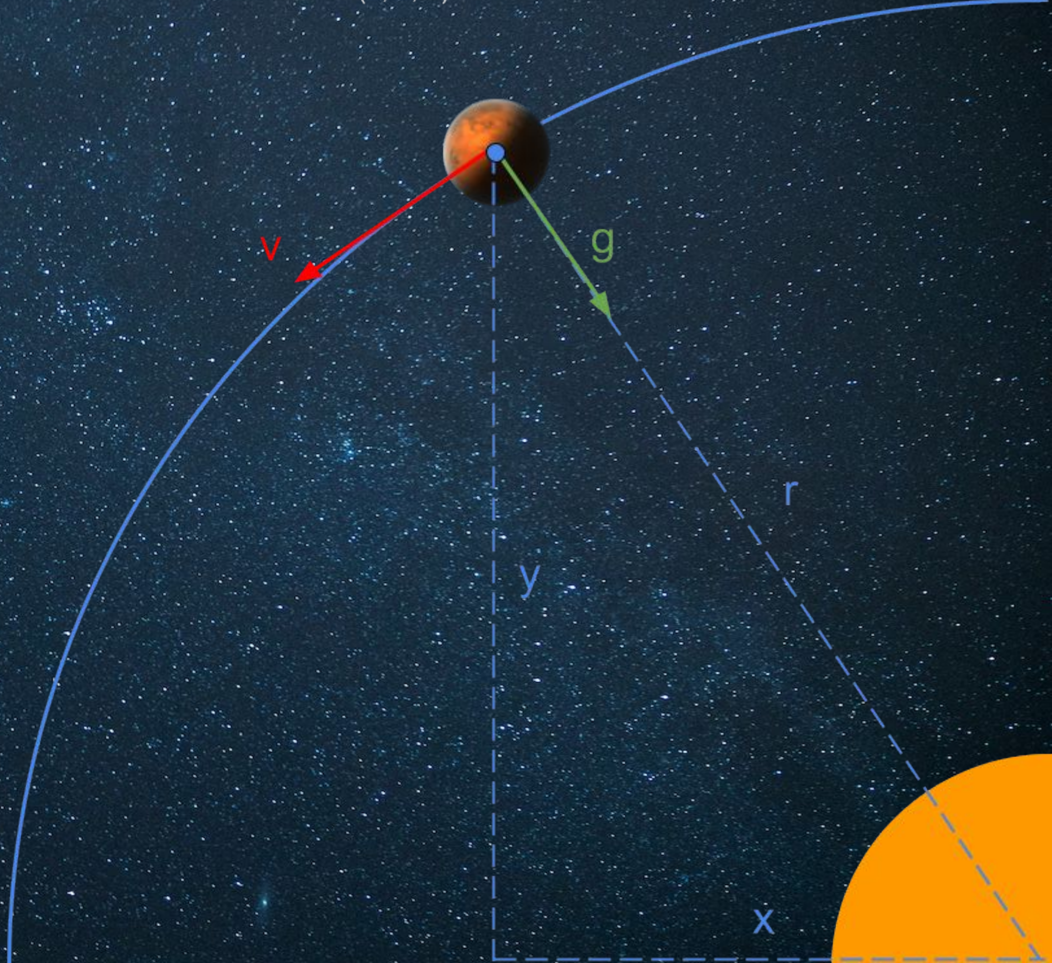
$$\cos\theta = \frac{g_x}{g}$$

$$g_x = \frac{GM_{\odot}}{(x^2+y^2)}\cos\theta = \frac{GM_{\odot}}{(x^2+y^2)}\frac{x}{\sqrt{(x^2+y^2)}}$$

$$g_x = \frac{GM_{\odot}x}{(x^2 + y^2)^{\frac{3}{2}}}$$

$$g_y = \frac{GM_{\odot}}{(x^2+y^2)}\sin\theta = \frac{GM_{\odot}y}{(x^2+y^2)^{\frac{3}{2}}}$$

*All movement is in reference to (0, 0, 0)



Orbital Mechanics

*All movement is in reference to (0, 0, 0)

$$g = \frac{GM_\odot}{r^2}$$

$$r = \sqrt{x^2 + y^2}$$

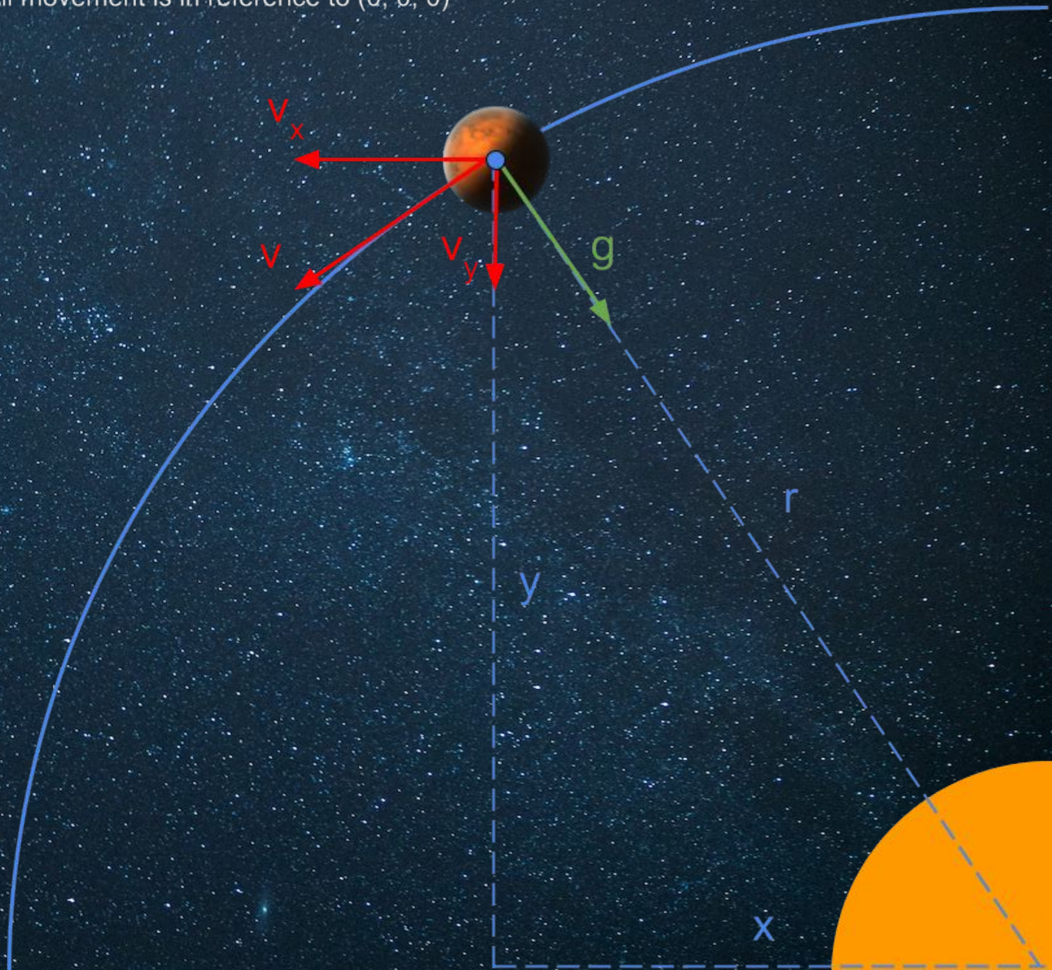
$$g = \frac{GM_\odot}{\left(\sqrt{x^2 + y^2}\right)^2}$$

$$\cos\theta = \frac{g_x}{g}$$

$$g_x = \frac{GM_\odot}{(x^2+y^2)}\cos\theta = \frac{GM_\odot}{(x^2+y^2)}\frac{x}{\sqrt{(x^2+y^2)}}$$

$$g_x = \frac{GM_\odot x}{(x^2 + y^2)^{\frac{3}{2}}}$$

$$g_y = \frac{GM_\odot}{(x^2+y^2)}\sin\theta = \frac{GM_\odot y}{(x^2+y^2)^{\frac{3}{2}}}$$



Orbital Mechanics

$$g = \frac{GM_\odot}{r^2}$$

r = $\sqrt{x^2 + y^2}$

$$g = \frac{GM_\odot}{\left(\sqrt{x^2 + y^2}\right)^2}$$

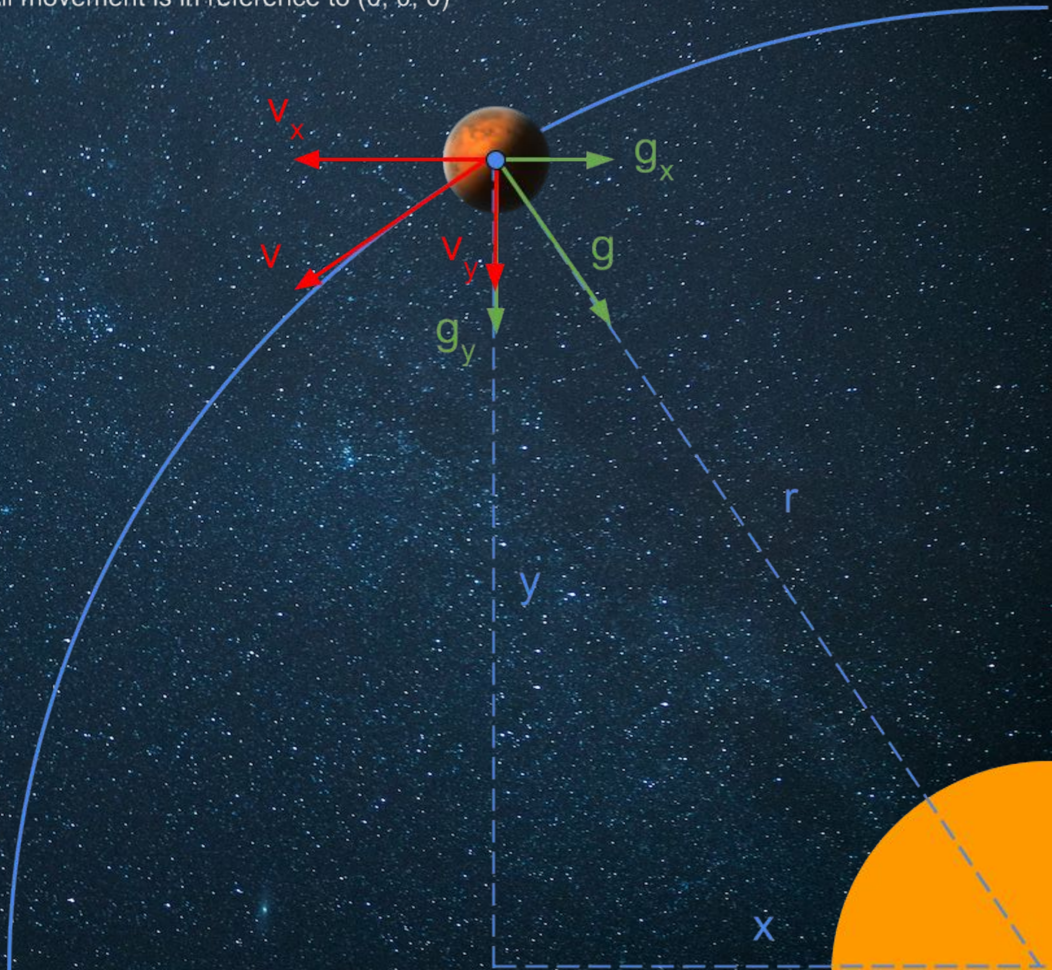
$$\cos\theta = \frac{g_x}{g}$$

$$g_x = \frac{GM_\odot}{(x^2+y^2)}\cos\theta = \frac{GM_\odot}{(x^2+y^2)}\frac{x}{\sqrt{(x^2+y^2)}}$$

$$g_x = \frac{GM_\odot x}{(x^2 + y^2)^{\frac{3}{2}}}$$

$$g_y = \frac{GM_\odot}{(x^2+y^2)}\sin\theta = \frac{GM_\odot y}{(x^2+y^2)^{\frac{3}{2}}}$$

*All movement is in reference to (0, 0, 0)



Orbital Mechanics

Acceleration:

Velocity:

Position:

$$g_x = \frac{GM_\odot x}{(x^2+y^2)^{\frac{3}{2}}} \qquad v_{xf} = v_{xi} + g_x dt \qquad x_f = x_i + v_{xi}dt + \frac{1}{2}g_x(dt)^2$$

$$g_y = \frac{GM_\odot y}{(x^2+y^2)^{\frac{3}{2}}} \qquad v_{yf} = v_{yi} + g_y dt \qquad y_f = y_i + v_{yi}dt + \frac{1}{2}g_y(dt)^2$$

Orbital Mechanics

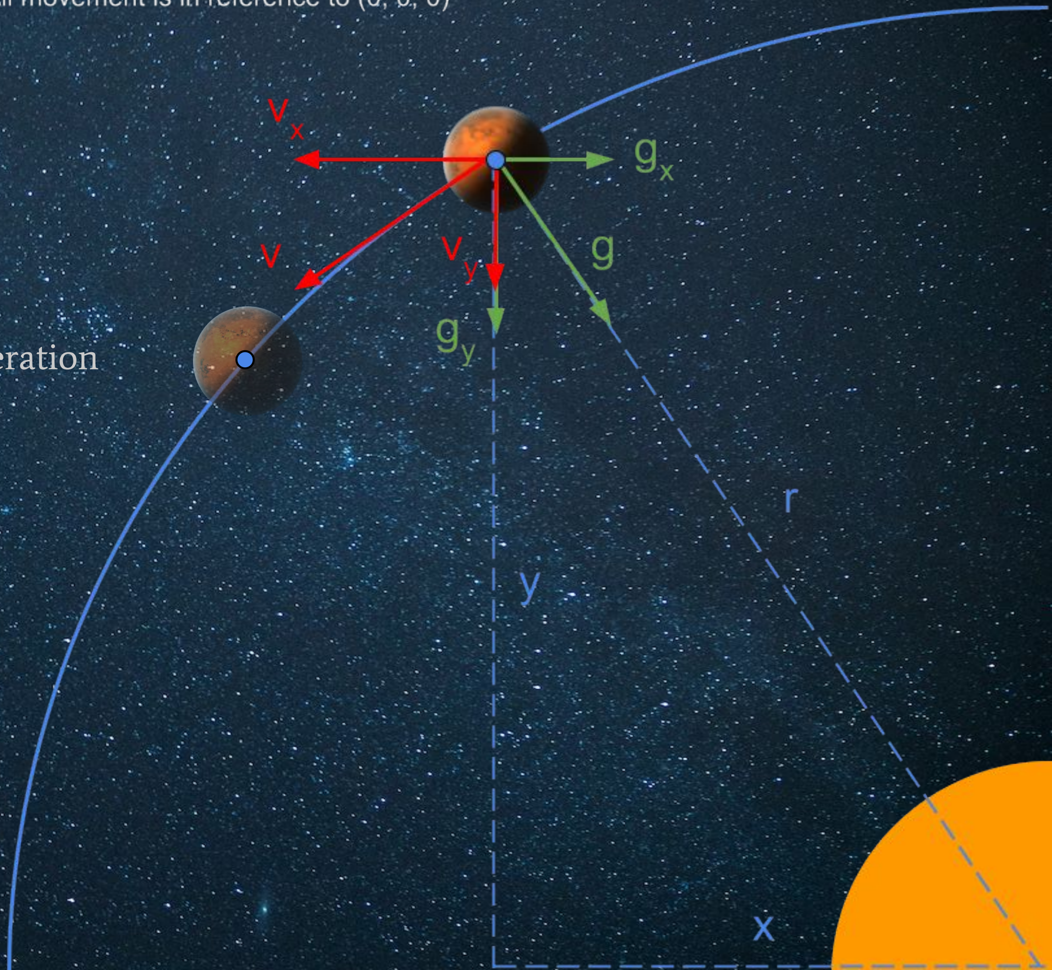
*All movement is in reference to (0, 0, 0)

Current X Position

Current X Acceleration (g_x)

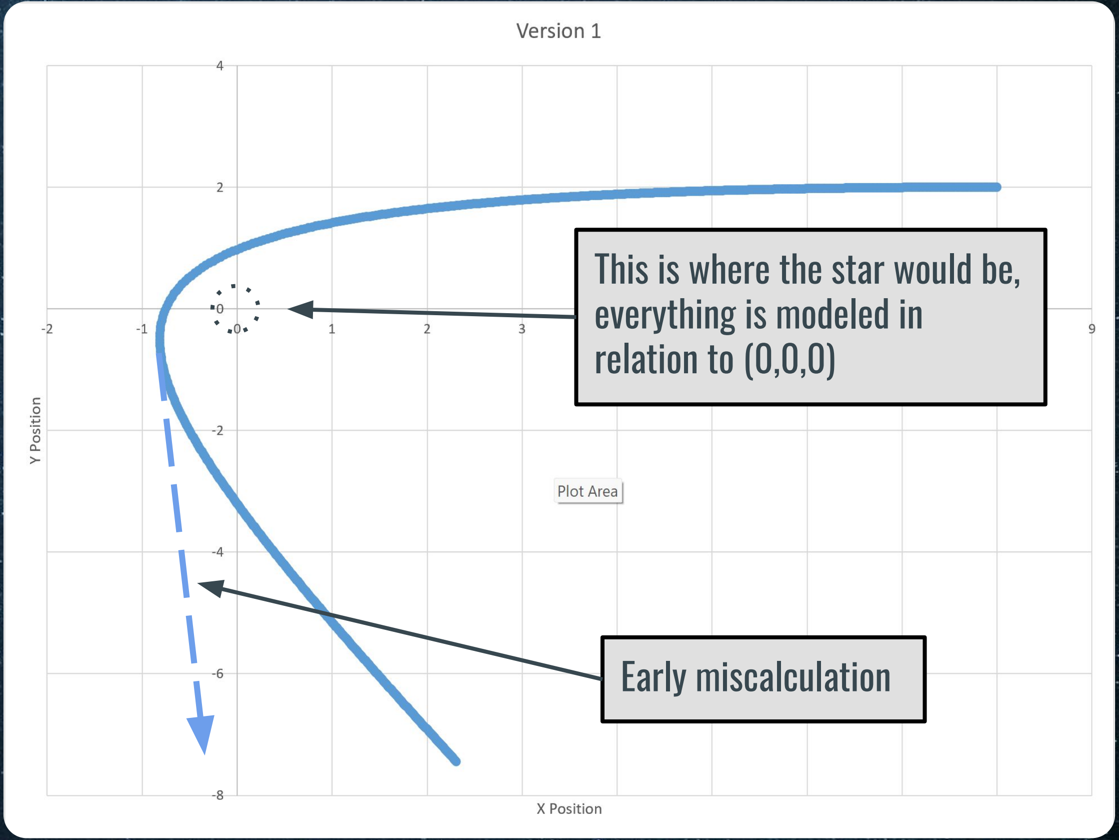
Current Velocity (v_x)

Next X Position



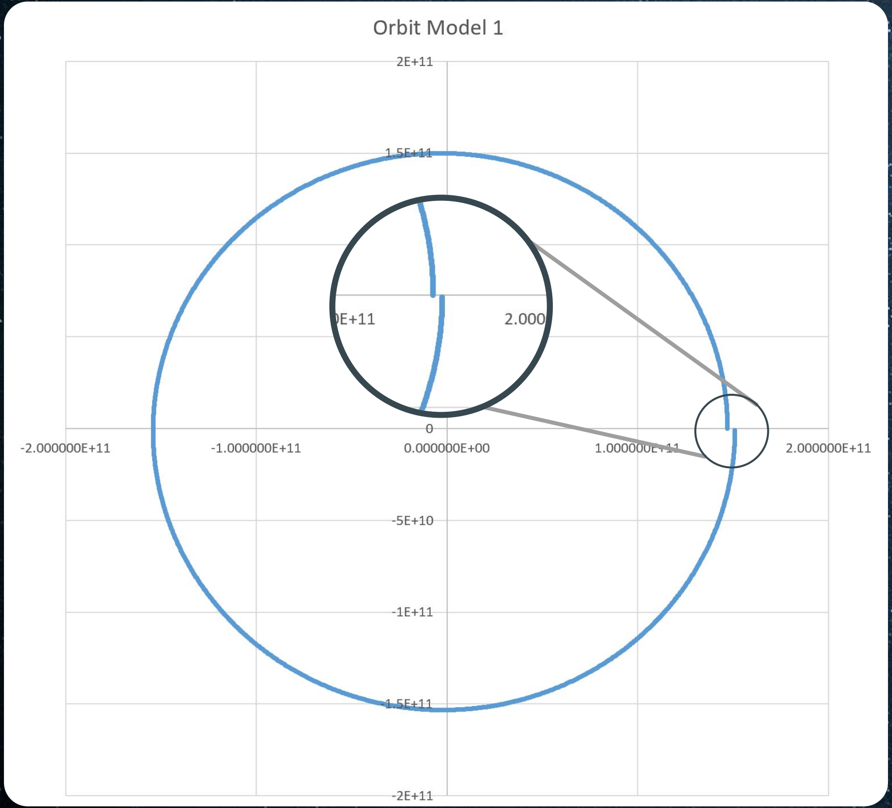
Mathematical Modeling

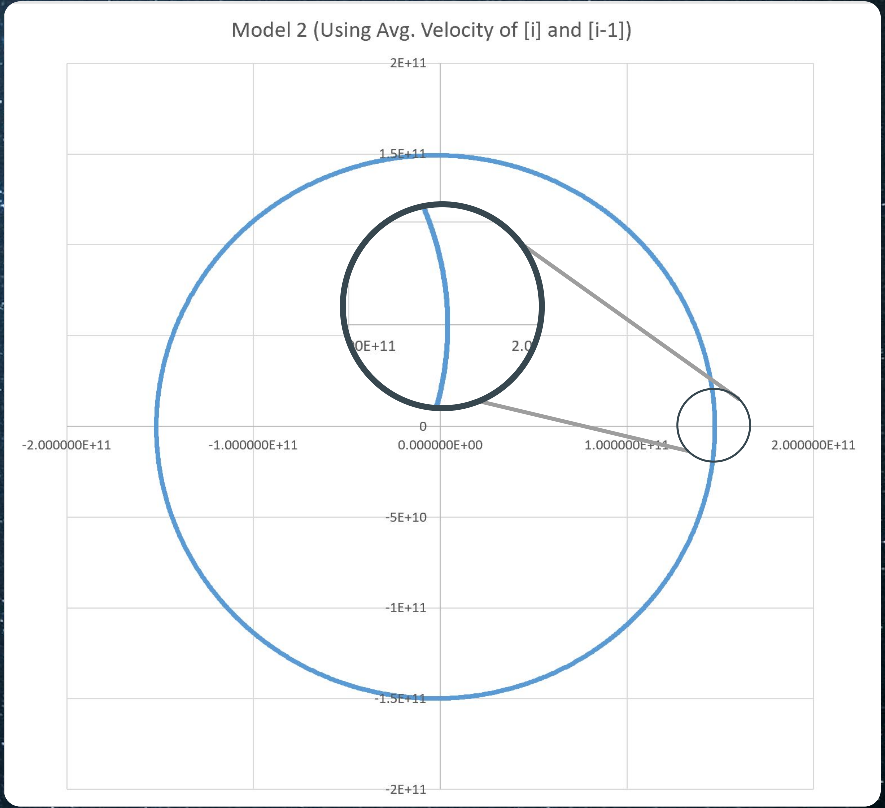
How hard could it be to make a circle?



Mathematical Modeling

How hard could it be to make a circle?





Numerical Complications

Space is only as big as you make it

- Large magnitudes in either direction.
- Size limits differ from language to language.
  - Javascript vs Python
- Data transfer hurdles.
- Scientific notation to the rescue.
  - Information and Magnitude

Large negative exponents

$$\boxed{G = 6.674 \times 10^{-11}}$$

Large positive exponents

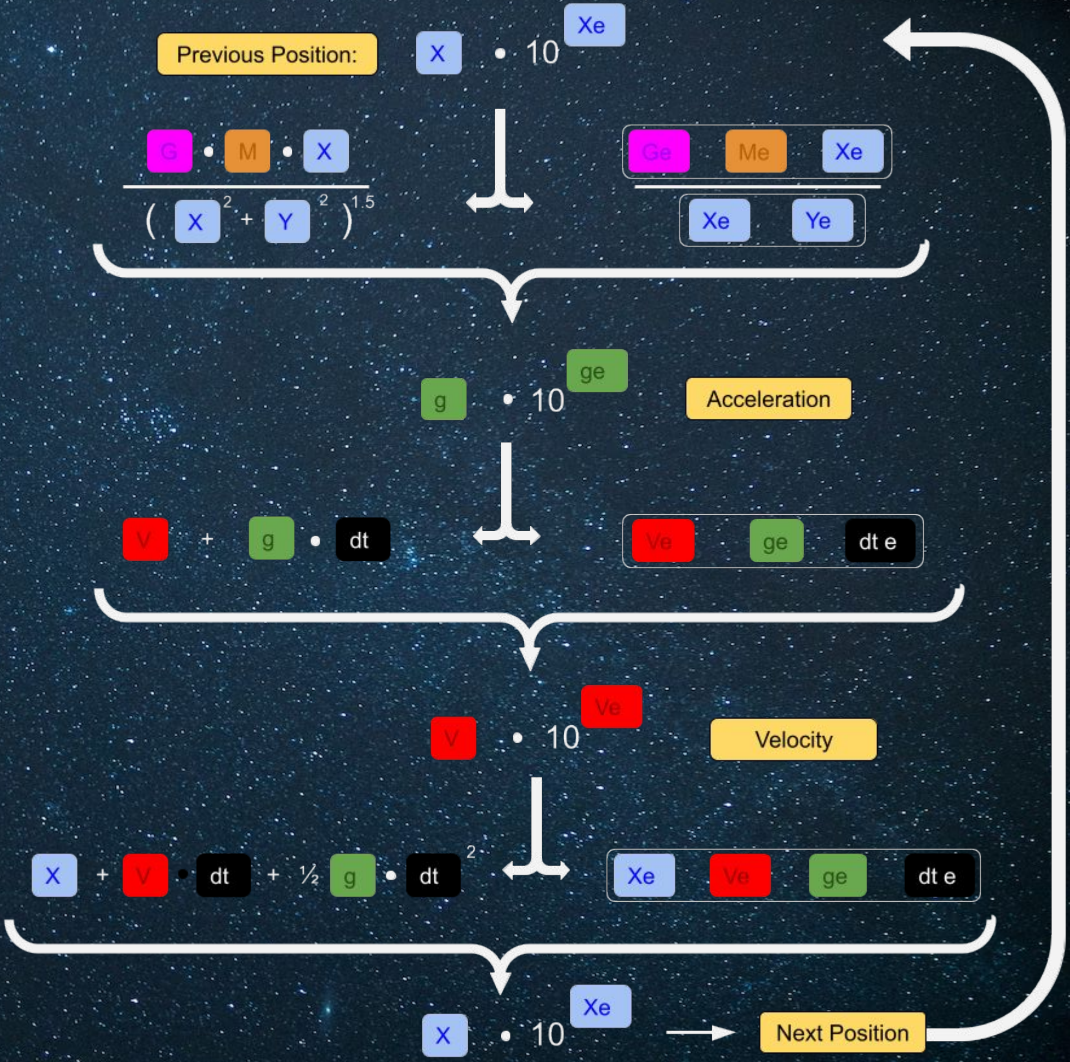
$$\boxed{M_{\odot} = 1.989 \times 10^{30}}$$

Algorithms

Finally some coding

G	Gravitational Constant	Ge
M	Dominant Mass	Me
X	X Position	Xe
V	X Velocity	Ve
g	X Gravitational Acceleration	ge
dt	Time Interval	dt e

Scientific Notation Modifiers ($\times 10^{n}$)



①

My Algorithm

I’m almost done talking about it

- Is lightweight → keeps as little data on-hand as possible.
- Allows for instant adjustment of current parameters.
- Is modular:
  - Offloading to object-by-object model

System Goals
What I was trying to do.

Learning tool first and foremost.

① Accurate modeling of Newtonian orbital dynamics.

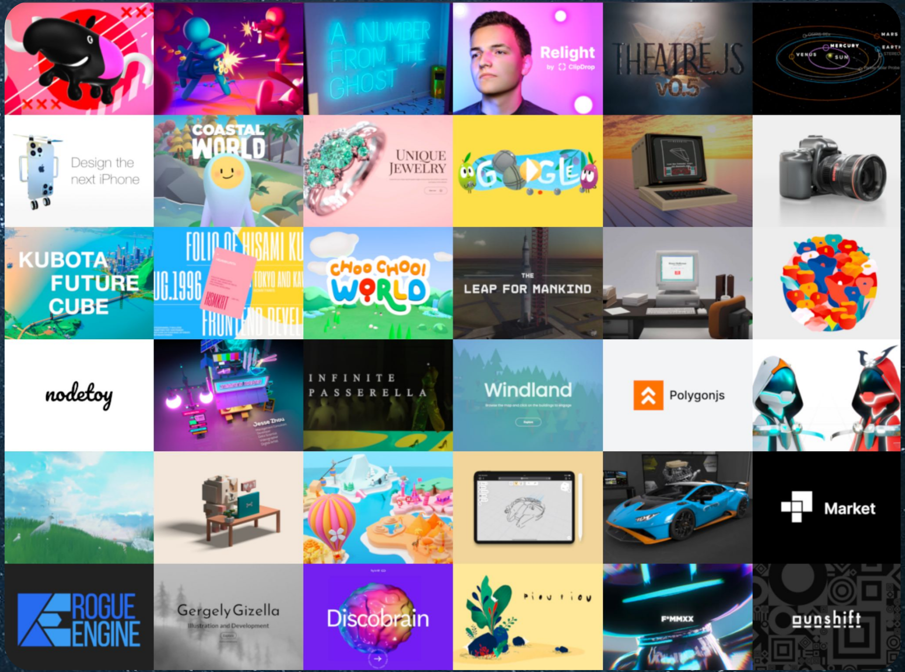
② Modularity of as many astrophysical variables as possible.

③ Realistic visual representations.

Animation

Algorithms are only cool if you can see them

- THREEjs
- Why it works so well:
    - Built-in coordinate system.
    - Object-based.
    - Built for browsers.

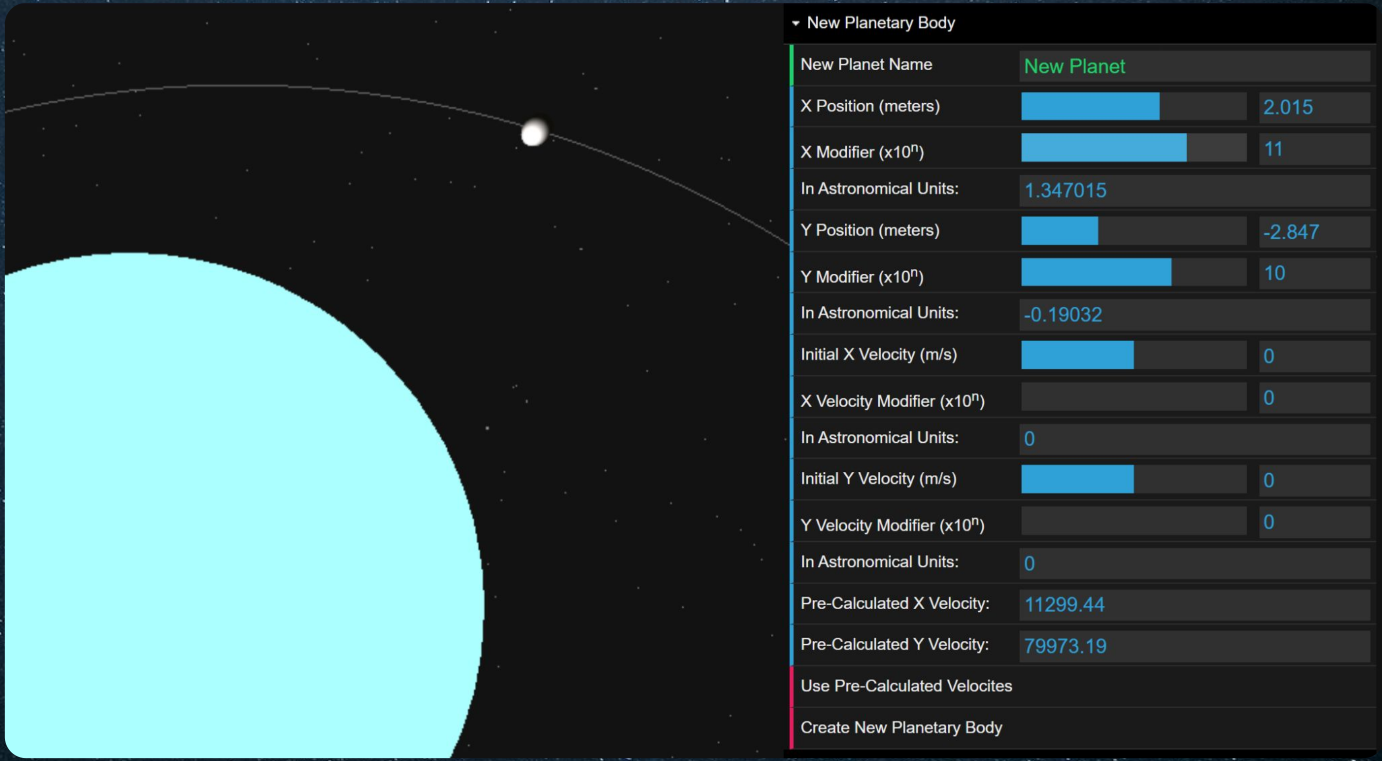


User Control

Buttons make everything better

Dat GUI

- Single parameter alterations.
- Allows for real time iterative adjustments.



System Goals

What I was trying to do.

Learning tool first and foremost.

① Accurate modeling of Newtonian orbital dynamics.

② Modularity of as many astrophysical variables as possible.

③ Realistic visual representations.

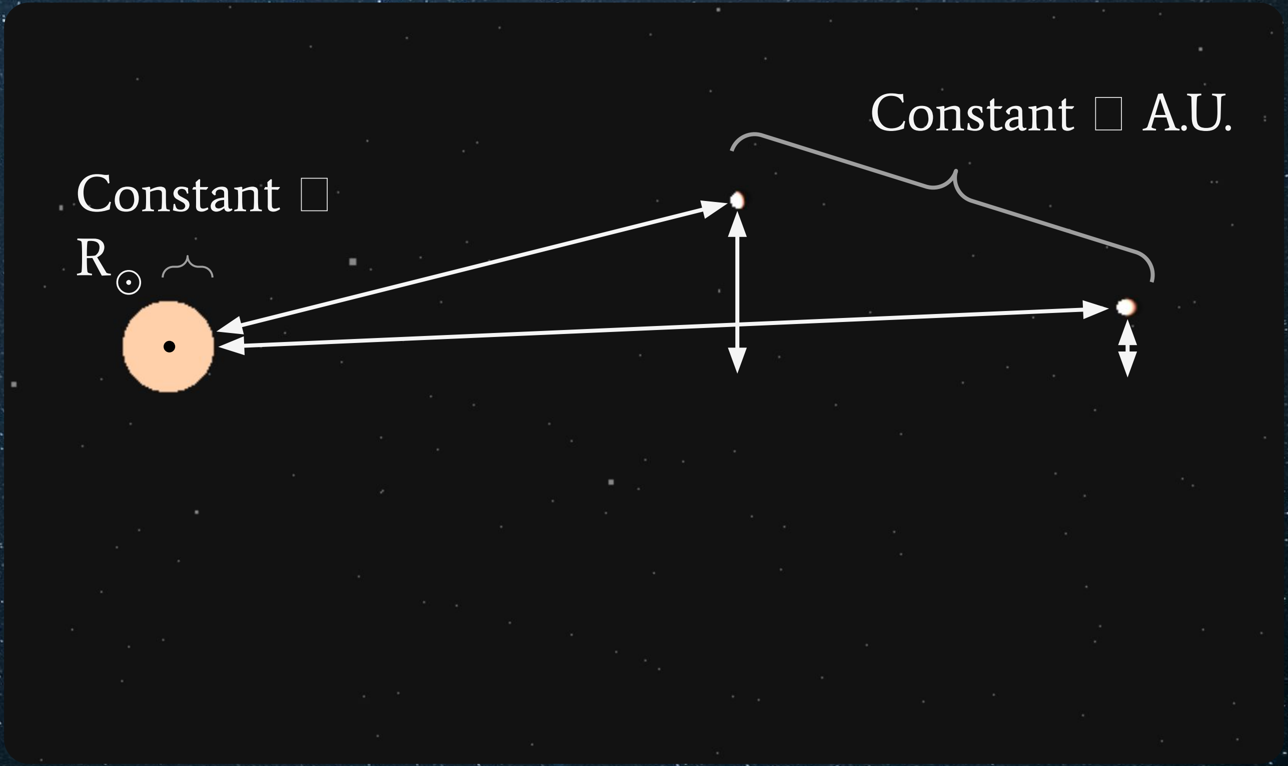
Scaling

Eyeballing isn’t going to work here.

$R_\odot$ Solar Radii

$M_\odot$ Solar Mass

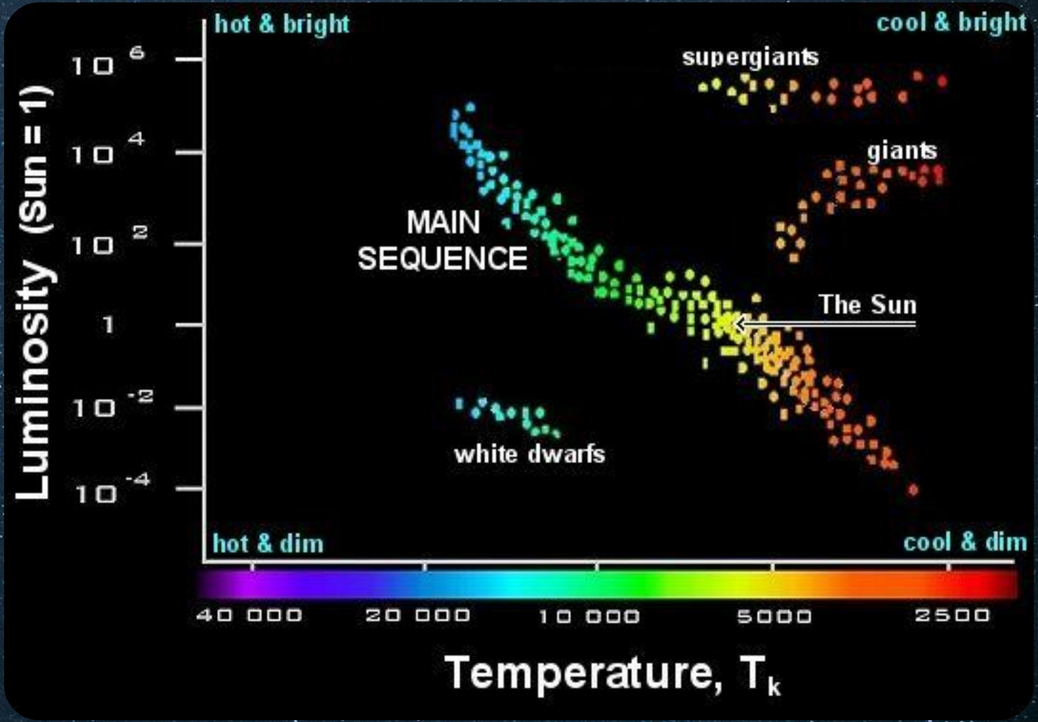
A.U. Astronomical Units



Stellar Mechanics

The only thing holding my project together

- Mass determines everything.
- Main Sequence stars
- Solar Units
- $L_{\odot} = R_{\odot}{}^{2} T_{\odot}{}^{4}$
- Temperature determines color.

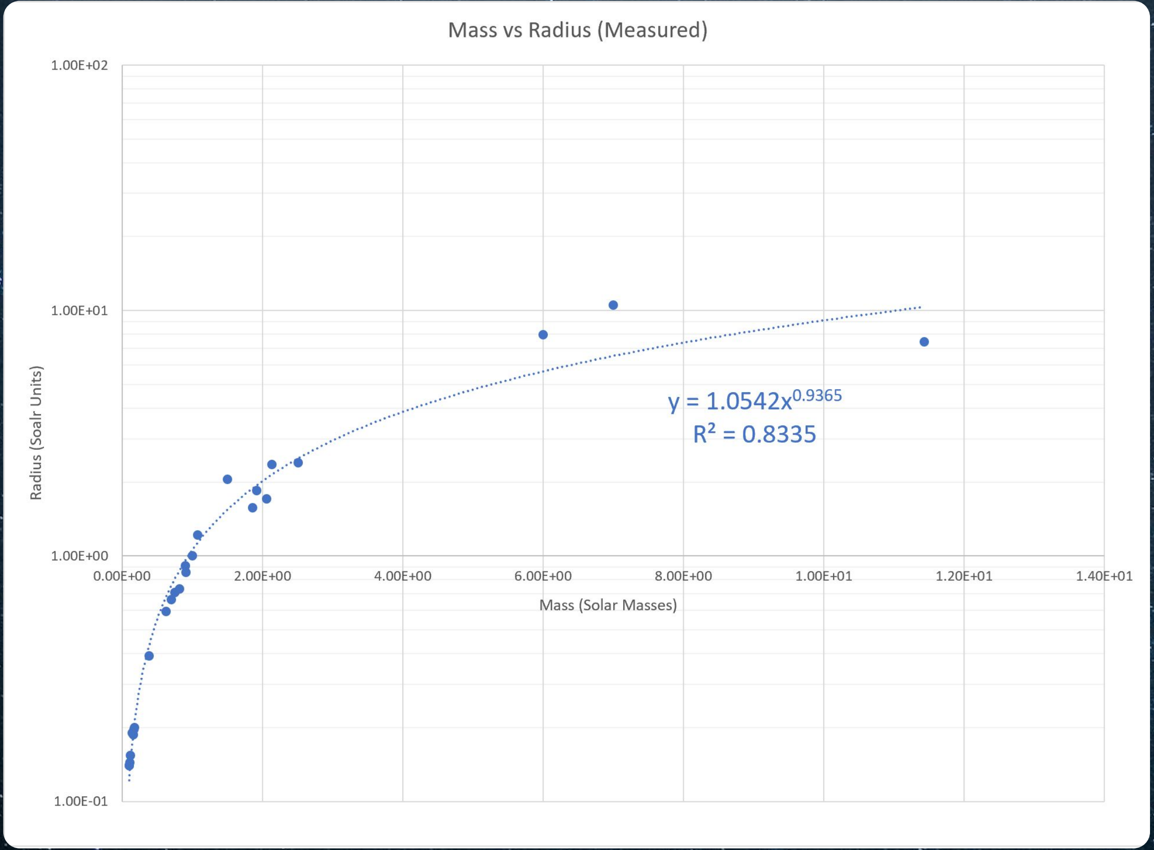


https://www.wwu.edu/depts/skywise/a101_hrdiagram.html

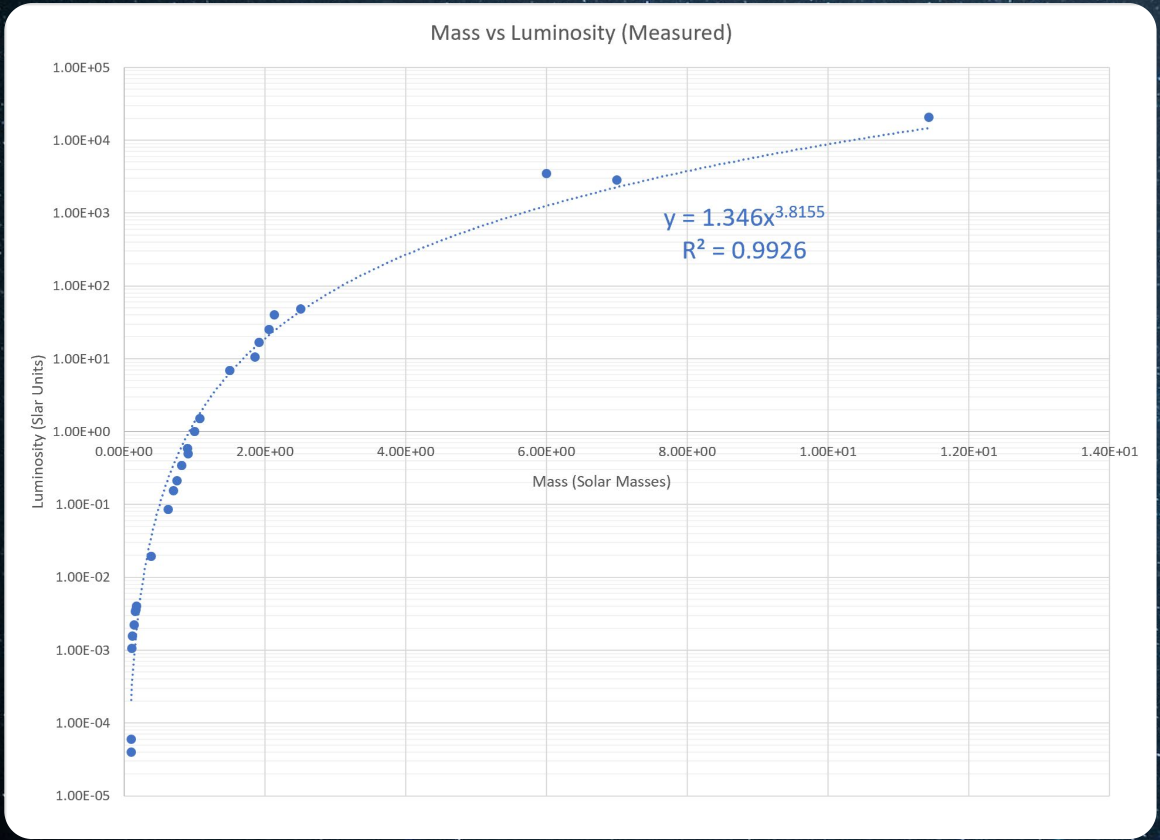
③

More Modeling - Calculating Stellar Radii

Star
Sun
Alpha Centauri A
Alpha Centauri B
Alpha Centauri C
Wolf 359
Lalande 21185
Sirius A
Luyten 726-8 A
Luyten 726-8 B
Ross 154
Ross 248
Epsilon Eridani
Ross 128
61 Cygni A
61 Cygni B
Procyon A
Epsilon Indi
Vega
Achernar
Beta Centauri
Altair
Spica
Delta Aquarii A
70 Ophiuchi A
Delta Persei
Barnard's Star
Luyten 789-6
Alpha Crucis
Fomalhaut
Beta Crucis



Even More Modeling - Calculating Stellar Luminosity/Temp



$$L_{\odot} = R_{\odot}^{2} T_{\odot}^{4}$$

$$R_{\odot} = 1.05(M_{\odot})^{0.937}$$

$$L_{\odot} = 1.346(M_{\odot})^{3.8155}$$

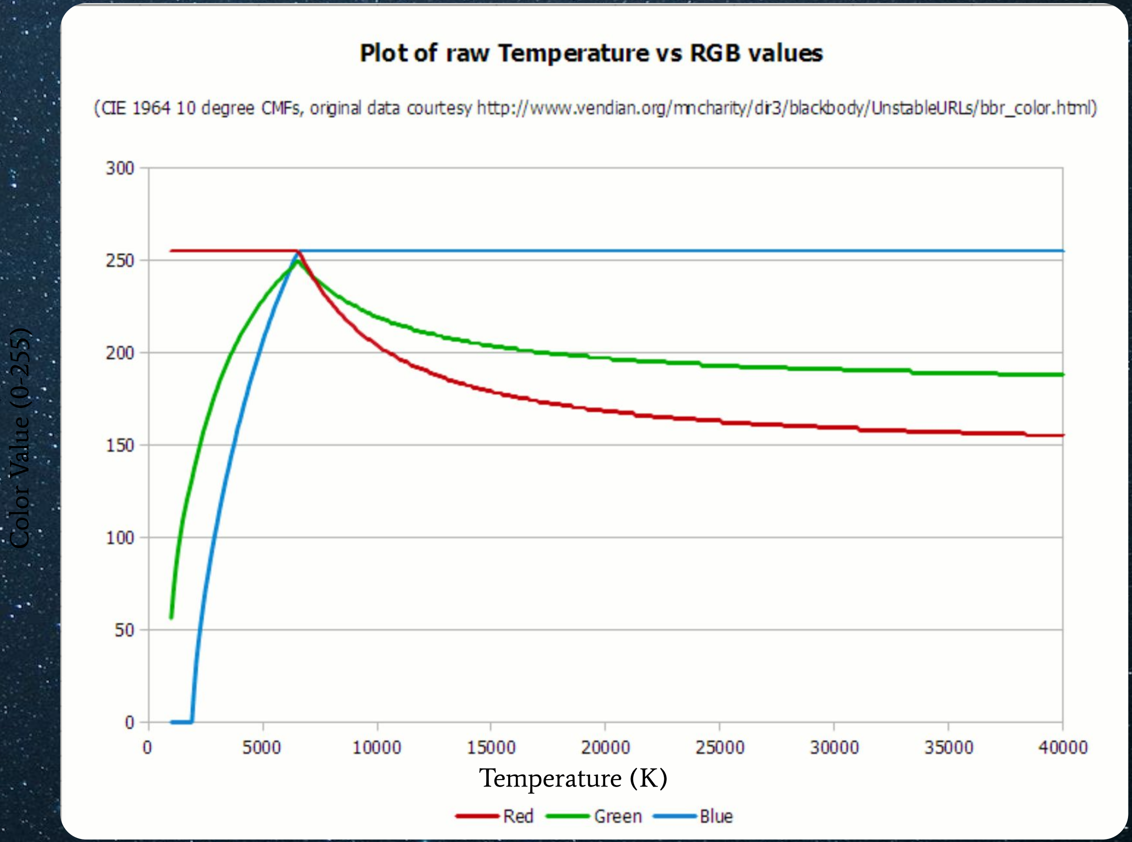
$$T_{\odot} = \sqrt[4]{\frac{L_{\odot}}{R_{\odot}^{2}}}$$

Temperature into RGB

Everything is better in color

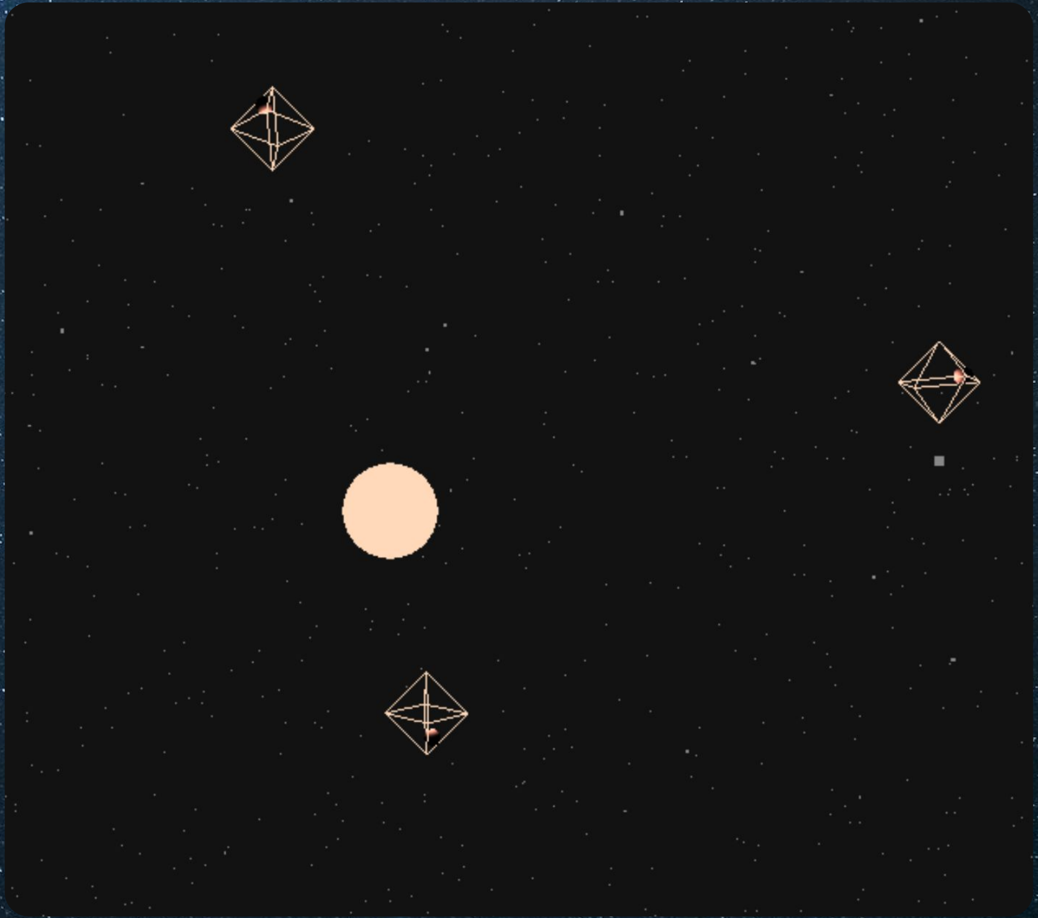
- Black body radiative colors.

- From mass to color via quantitative analysis.



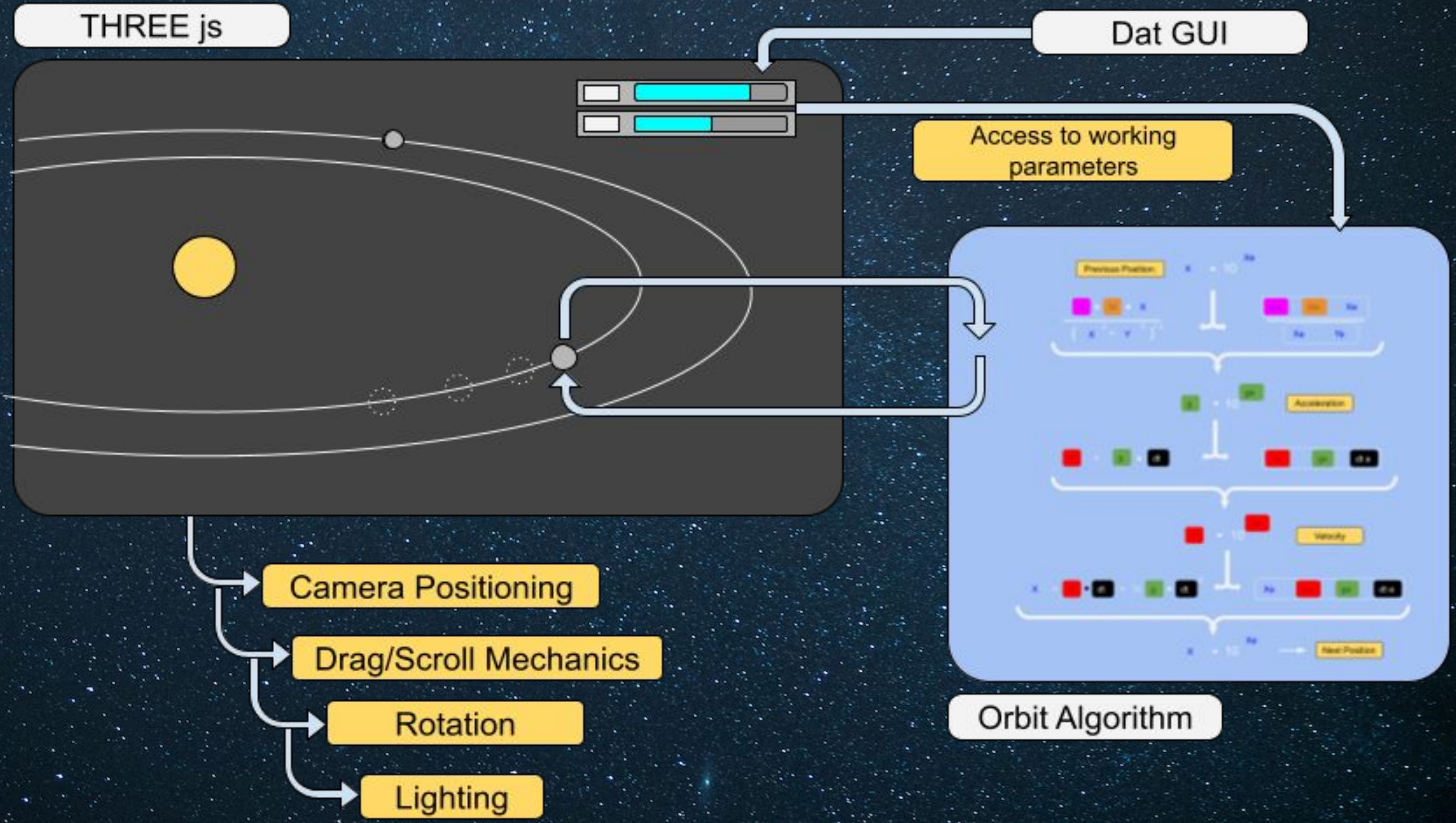
Let There Be Light

- Point lighting
- Directional lighting.
- Realistic visuals
  - Intensity or Color?



System Design

Things are starting to shape up



Extensions and Applications

- Binary star systems.
- Exotic gravitational sources (white dwarfs, black holes, supergiants, etc.)
- Relativistic corrections → black holes.
- Lunar objects.
- Nuclear (Coulomb) Scattering.