Planetarium Thingy - By Ethan Powell

A planet and the smaller planets and moons that orbit it. Complete with a camera that partially follows the movement of the mouse.

Things Required:

- Planets
- MousePosition

Planets:

- Central planet or solar mass from which the rotation of all other solar bodies are based on.
- All planets have an empty at the center of the planet they are orbiting. The empty is rotated and the planet orbits the body it is supposed to. (Addendum: Empties were finicky and uncooperative, had to resort to using sprites to anchor planets to an orbit point. Worked fine but still feels a bit sloppy. An additional planet is overlaid onto the central planet to hide all the sprites.)
- This can be repeated to have planets that orbit planets of which are orbiting planets. (Addendum, worked without a hitch.)
- If the player clicks within the bounds of a planet, that planet's orbit is reversed. (Addendum, not only is the orbit reversed but the rotation of the planet itself, I could have either rotation toggled individually by way of splitting both across RMB and LMB, but I am lazy.)

Planet Structure:

- Rotation Point (Sprite/Transform)
- Planet (Sprite, with art.)
- Random Speed (Random Float)
- Rotation direction. (Addendum, clockwise by default.)

Mouse Position:

- The camera slides around the screen depending on where the mouse is. (Done! See below for details.)
- An animation curve is used to cap how far the camera can move from the center of the screen. (Getting the animation curve right took 20+ minutes of trial and error between the curve itself and the AnimationCurve, go too short or far in the animation curve and the camera starts to jitter, the slider is now in a comfortable position.)
- The camera is lerped between the mouse position and the center of the screen. (The distance between the center of the game and the mouse is fed into and evaluated against the animation curve, and then that value is fed into a lerp between the center and the mouse, divided by two.)