Game Physics - Jayden L.

SOLAR SYSTEM SIMULATION

ORBITAL MECHANICS

01

Elliptical Orbit

02

Varying Velocities

03

Orbital Period

KEPLER 1ST

```
//calc position on the ellipse
//a = semi-major axis | b = semi-minor axis
//Keplers 1st law -> keeping planets on elliptical orbits around the sun
float x = a * Mathf.Cos(currentAngle);
float y = b * Mathf.Sin(currentAngle);
Vector3 desiredPosition =
new Vector3(
    particle.target.transform.position.x + x,
    particle.target.transform.position.y + y,
    particle.transform.position.z
);
```

Parametric Equation of an Ellipse

Sun Pos used as Offset

KEPLER 2ND

```
//calc velocity magnitude using vis-viva equation
//v = sqrt(G * M * (2/r - 1/a))
//Keplers 2nd law -> varying velocity on the ellipse
//  = faster when closer to sun, slower when further YET same area convered
float velocityMagnitude = Mathf.Sqrt(gravityConstant * particle.sunMass * (2 / r - 1 / a));
```

Higher orbital velocity nearing Major Vertex

Lower orbital velocity nearing Minor Vertex

Same Area!

$$v = \sqrt{G \cdot M \cdot \left(\frac{2}{r} - \frac{1}{a}\right)}$$

KEPLER 3RD

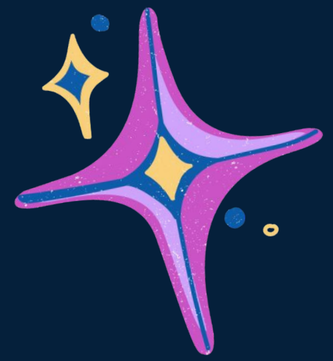
```
//keplers 3rd law -> T = 2π * sqrt(a^3 / (G * M))
//the further the planet from the sn, the slower its orbitalperiod
float gravityConstant = Planet.gravConstant;    // gravity constant
float a = particle.semiMajorAxis;               // semi-major axis
float orbitalPeriod = 2 * Mathf.PI * Mathf.Sqrt(Mathf.Pow(a, 3) / (gravityConstant * particle.sunMass));
```

Furthest Orbit to sun = Slowest Orbital Period

Closest Orbit to sun = Fastest Orbital Period

$$T = 2\pi\sqrt{\frac{a^3}{G \cdot M}}$$

GRAVITY



$$F_g = G\frac{m_1 m_2}{r^2}$$

where

- F_g is the force
- G is the gravitational constant (6.674×10^{-11} $m^3.kg^{-1}.s^{-2}$)
- m_1 and m_2 are the masses of the objects
- r is the distance between the centers of the objects

GRAVITY

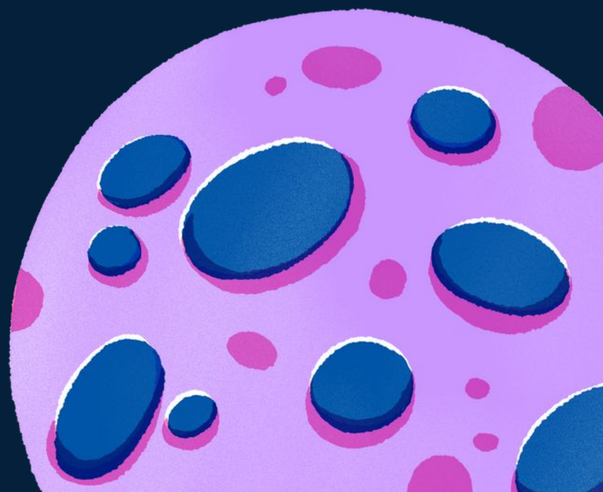
```
float velocityMagnitude = Mathf.Sqrt(gravityConstant * particle.sunMass * (2 / r - 1 / a));
```

Velocity altered through vis-viva equation

```
Vector3 tangent = new Vector3(-Mathf.Sin(currentAngle), Mathf.Cos(currentAngle), 0).normalized;
```

Velocity direction altered through tangent vector

TECHNICAL DEMO



SCALING

Factor of: 2.8748 * 10^-8

Sphere	Real Dist. From Sun	Scaled Dist.
Earth	1.496 * 10^11 M	4228.69
Mercury	5.79 * 10^10 M	1322.41
Neptune	4.503 * 10^12 M	128533.85

Due to floating-point precision limitations, it is recommended to bring the world coordinates of the GameObject within a smaller range.

PROBLEMS

- Scaling
- LineRenderer
- Gravitational Calculation
- All 3 Laws

QUESTIONS

THANKS :3

-JAYDEN

CREDITS

Slides Carnival

This presentation template is free for everyone to use thanks to the following:

SlidesCarnival for the presentation template

Pexels for the photos

HAPPY DESIGNING!