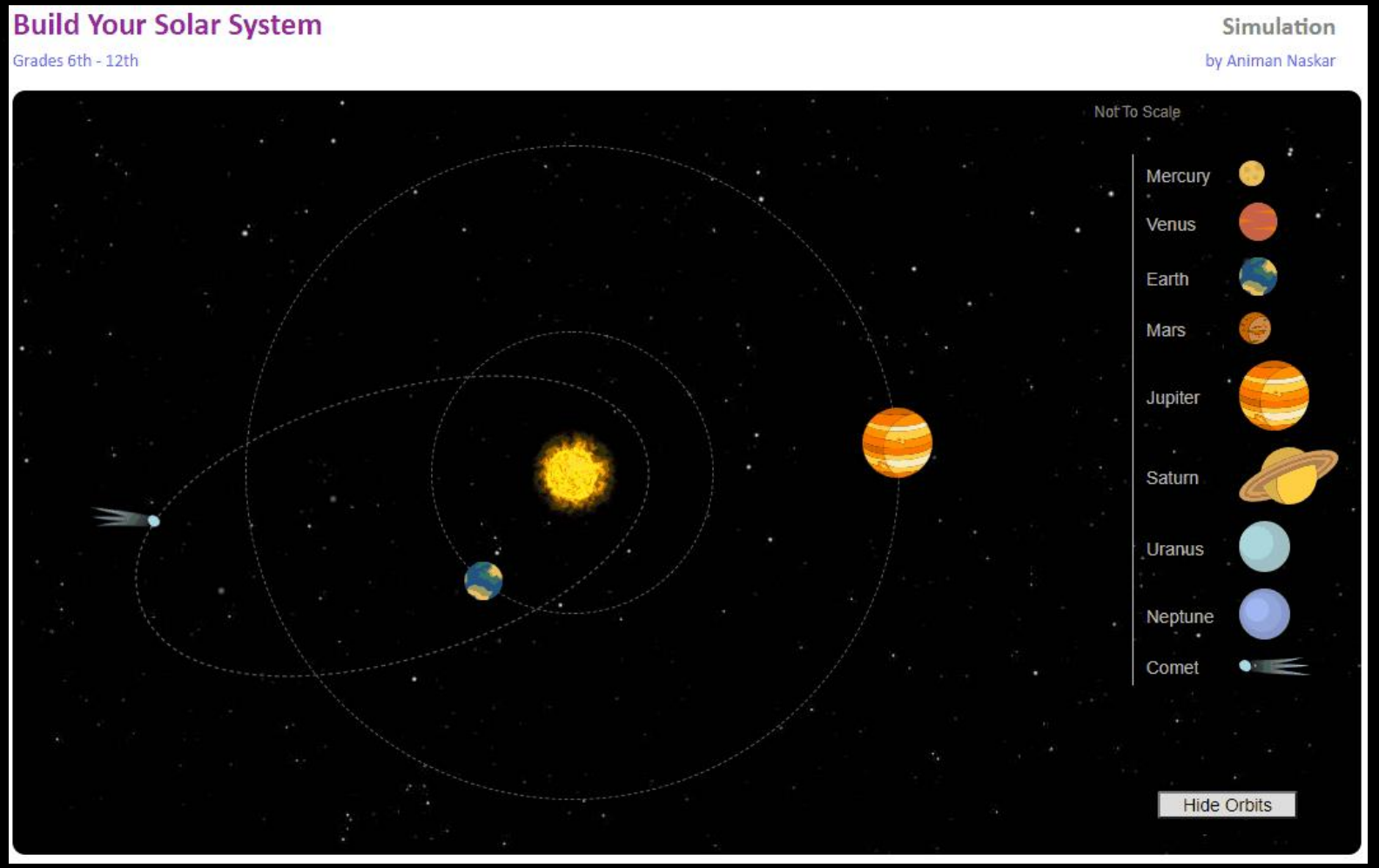
# 太阳系仿真
# Solar System Simulation

0xzhang

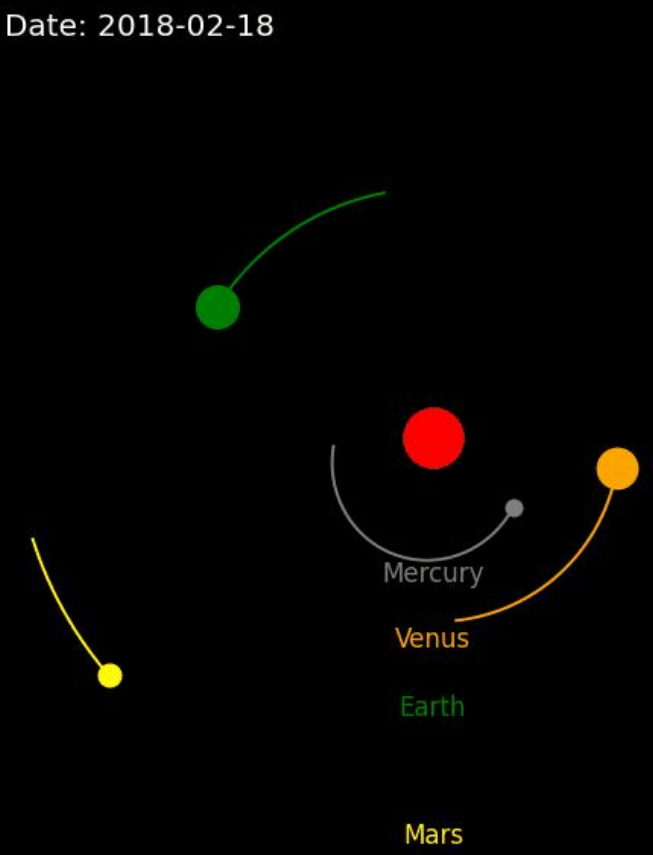
2021/11/11

# Build Your Solar System




https://simpop.org/solar-system/solar-system.htm

# 用86行Python代码模拟太阳系




何崇崇，https://zhuanlan.zhihu.com/p/102375135

# Physics

- Newton's law of universal gravitation

$$F = \frac{GMm}{r^2}$$

where:

$F$ is the force between the objects

$G$ is the gravitational constant

$M$ is the mass of one object

$m$ is the mass of another object

$r$ is the distance between the centers of the masses

# Physics

- Newton's second law of motion

$$F = ma$$

where:

$F$ is the net force acting on an object

$m$ is the mass of the object

$$\vec{a} = \frac{GM}{|\vec{r}|^2}\frac{\vec{r}}{|\vec{r}|}$$

# Physics

SI Units:

$$G = 6.67 \times 10^{-11} m^3 \cdot kg^{-1} \cdot s^{-2}$$

Astronomical unit(AU):

1 AU $\approx$ the distance from Earth to the Sun

- Applying Kepler's 3rd law

$$G = 2.959 \cdot 10^{-4} AU^3 \cdot day^{-2} M^{-1}$$

# Numerical

$$\vec{a} = \frac{GM}{|\vec{r}|^2}\frac{\vec{r}}{|\vec{r}|}$$

Semi-implicit Euler:

$$\vec{r}_{new} = \vec{r}_{old} + \vec{v}_{old}dt$$

$$\vec{v}_{new} = \vec{v}_{old} + \vec{a}_{new}dt$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \cdot dt$$

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = - \begin{bmatrix} x \\ y \\ z \end{bmatrix} \cdot \frac{GM}{(x^2 + y^2 + z^2)^{\frac{3}{2}}}$$

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} + \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \cdot dt$$

# Kernel

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \cdot dt$$

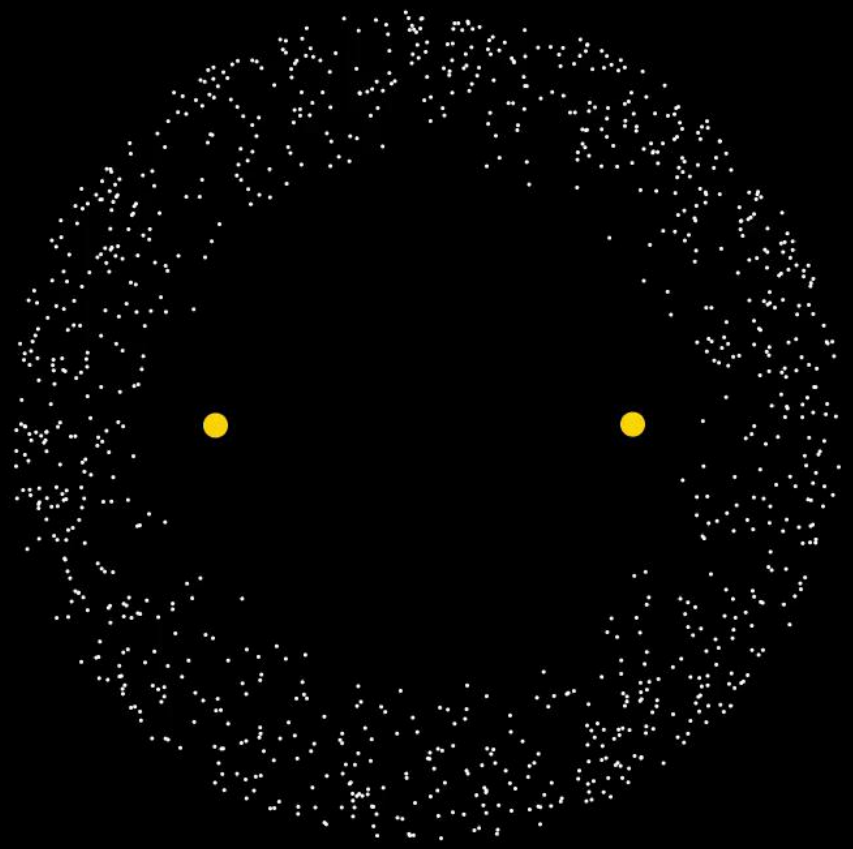
$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = - \begin{bmatrix} x \\ y \\ z \end{bmatrix} \cdot \frac{GM}{(x^2 + y^2 + z^2)^{\frac{3}{2}}}$$

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} + \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \cdot dt$$

```
@ti.kernel
def update(self, dt: ti.f32):
    self.pos[0] += self.vel[0] * dt
    sqr_sum = self.pos[0].norm_sqr()
    # in units of AU-D
    acc = -2.959e-4 * self.pos[0] / sqr_sum**(3. / 2)
    self.vel[0] += acc * dt
```

# ODOP

- SolarSystem
- CelestialObject
  - Sun
  - Planet



taichiCourse01/--Galaxy (github.com)

# Drawing Orbits

## definition

```
# Length of Cycle
LOC = 8000
self.orbit = ti.Vector.field(dim, ti.f32, shape=LOC)
self.orbit_colors = ti.Vector.field(dim, ti.f32, shape=LOC)
self.orbit_radius = self.radius / 8
```

## update

```
self.orbit[step % LOC] = self.pos[0]
```

## display

```
scene.particles(self.orbit,
                self.orbit_radius,
                per_vertex_color=self.orbit_colors)
```

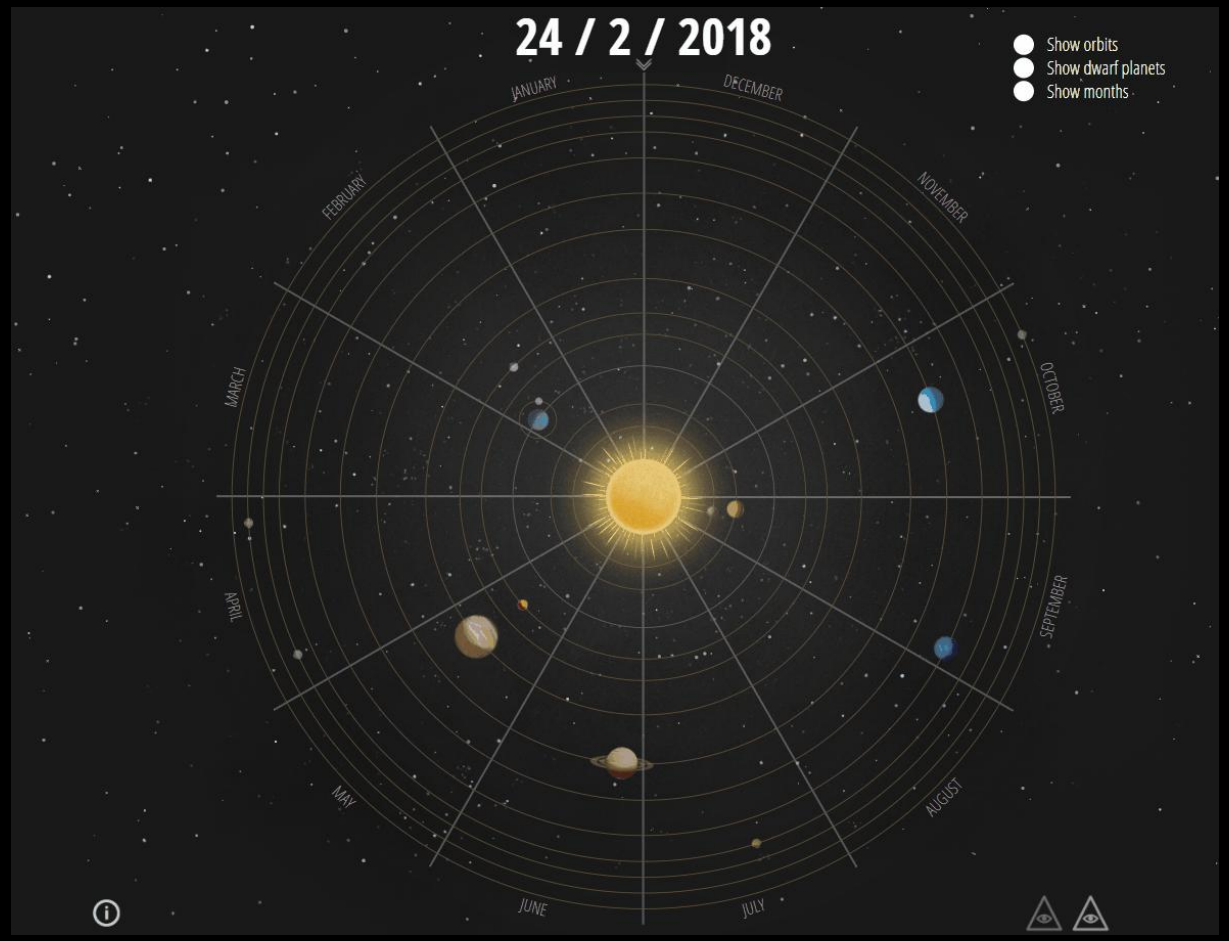
# Real Data, Real Date

- 使用真实数据设置初值，进行实时计算，JPL Horizons

Jet Propulsion Laboratory
California Institute of Technology

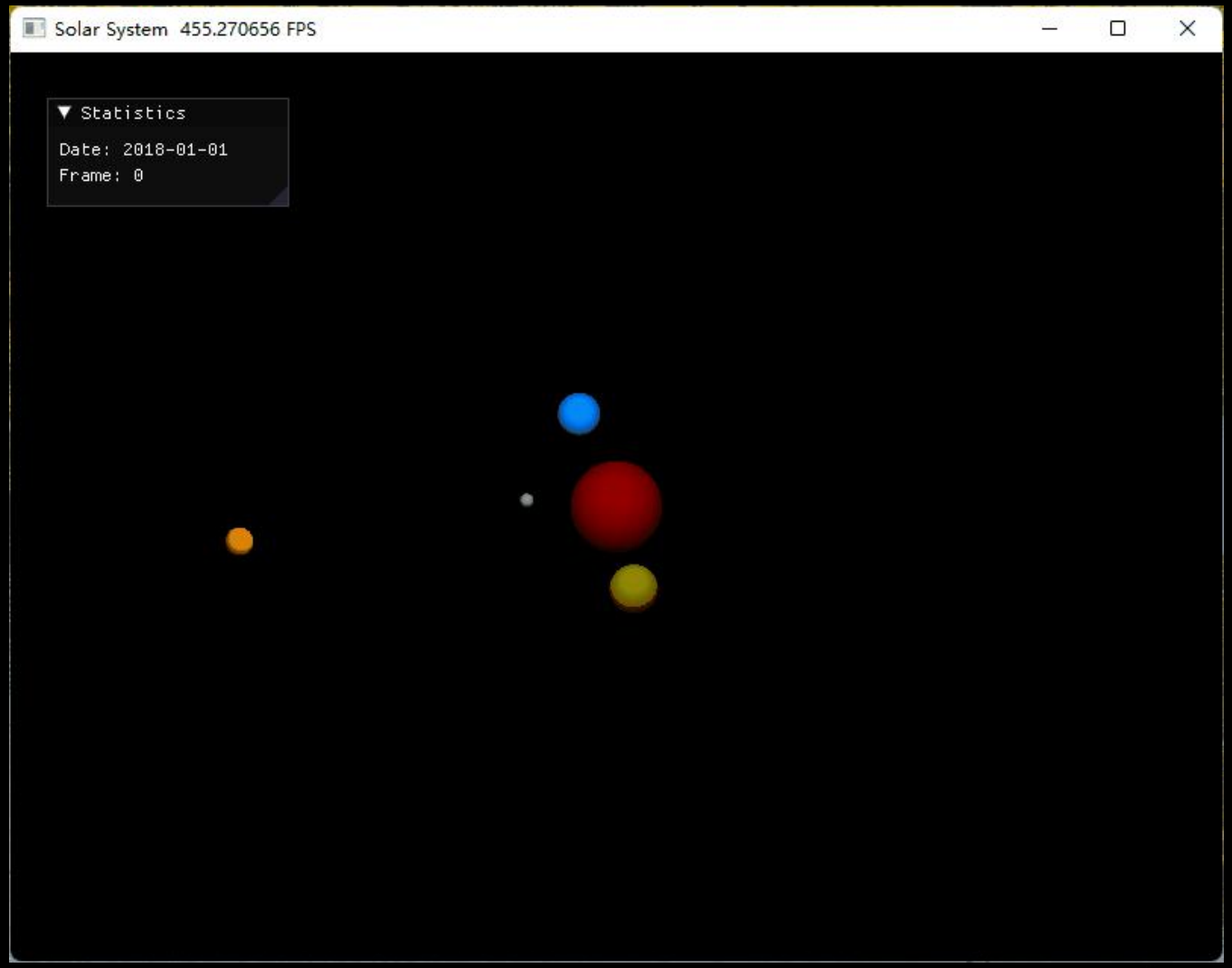
https://ssd.jpl.nasa.gov/horizons/

- 添加文本控件，每累计1天，更新日期




Solar System Orrery by Jeroen Gommers - Atlas Interactive (atlas-digital.nl)

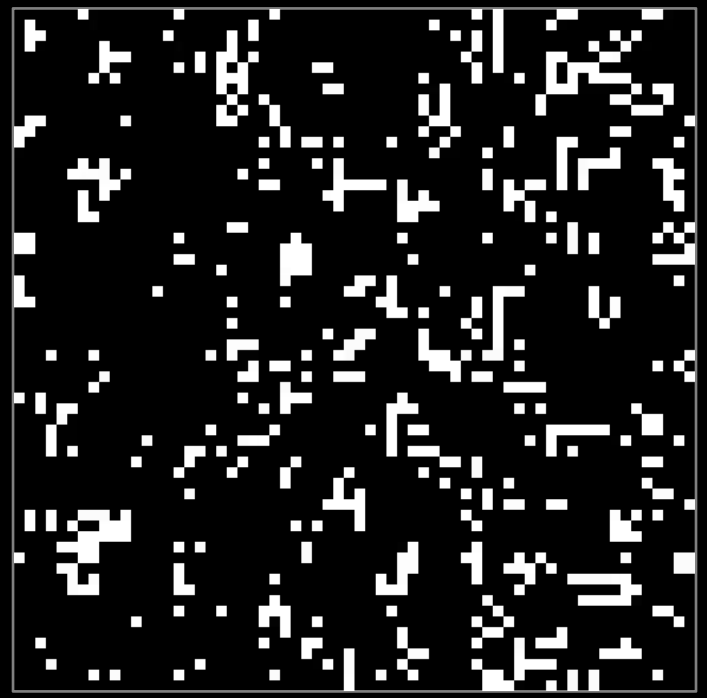
# Solar System



# TODO

- 2D GUI
- 加入地月系统
- 行星轨道只存储一圈
- Accuracy？试试使用其它的数值方法求解
- …

# Gallery



Game of Life


Wall Clock


Digital Clock

# Gifts



# Thanks!

GitHub: https://github.com/0xzhang/taichi-play