自由落体

自由落体运动解析公式：

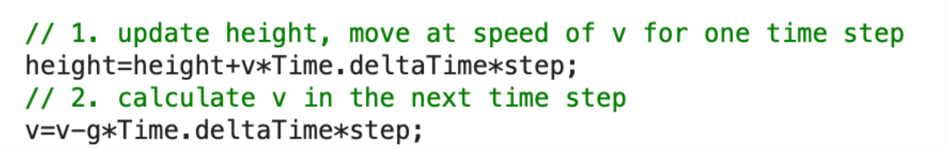
$$h = \frac{1}{2}gt^2$$

$$v_0 = 0$$

$$v = gt$$

根据以下公式计算：

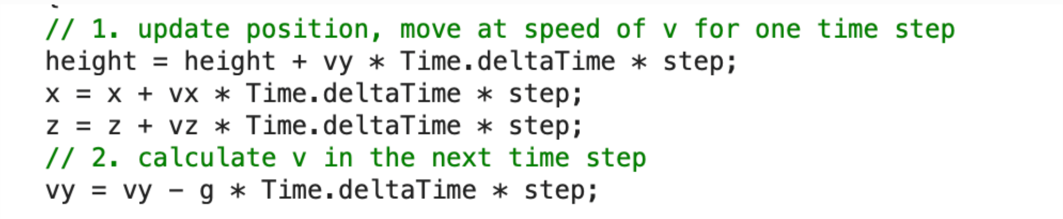
$$\mathbf{x}(t_0 + h) = \mathbf{x}_0 + h\dot{\mathbf{x}}(t_0).$$



```
// 1. update height, move at speed of v for one time step
height=height+v*Time.deltaTime*step;
// 2. calculate v in the next time step
v=v-g*Time.deltaTime*step;
```

抛物运动

与自由落体的不同是要计算3个方向的速度与位置



```
// 1. update position, move at speed of v for one time step
height = height + vy * Time.deltaTime * step;
x = x + vx * Time.deltaTime * step;
z = z + vz * Time.deltaTime * step;
// 2. calculate v in the next time step
vy = vy - g * Time.deltaTime * step;
```

Explicit Euler

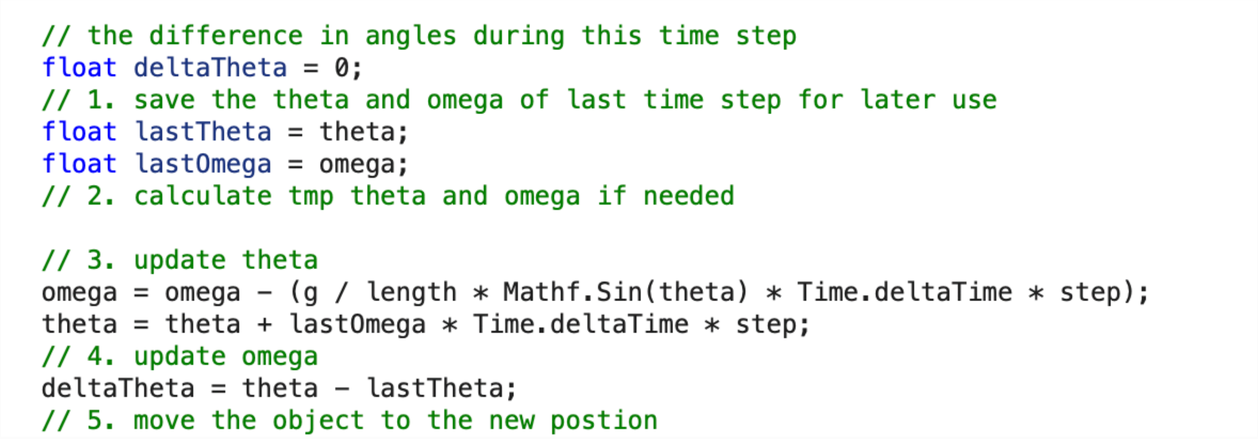
单摆运动公式：

$$\theta^{n+1} = \theta^n + \omega^n \Delta t$$

$$\omega^{n+1} = \omega^n - \frac{g}{l} sin\theta \Delta t$$

Explicit Euler:

$$x(t_0 + h) = x_0 + hf(x_0)$$



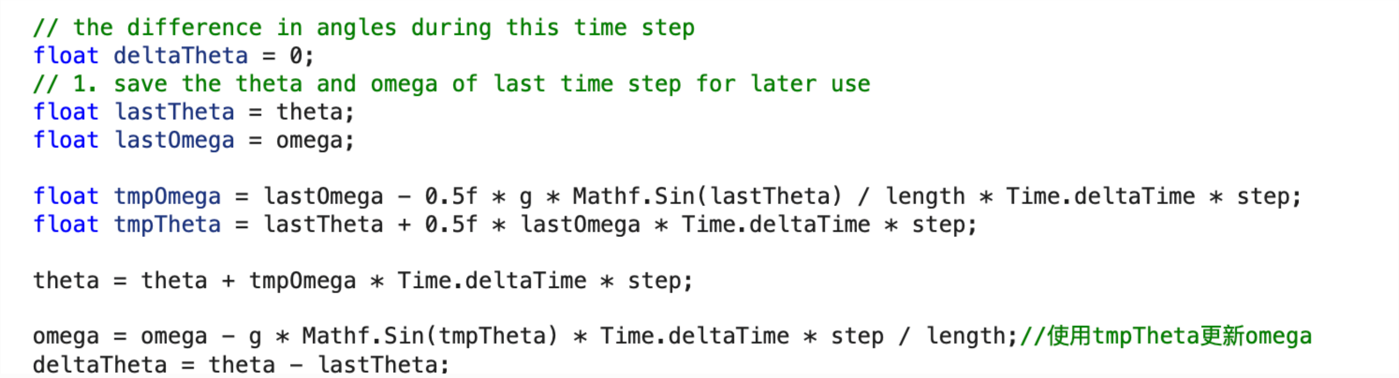
```
// the difference in angles during this time step
float deltaTheta = 0;
// 1. save the theta and omega of last time step for later use
float lastTheta = theta;
float lastOmega = omega;
// 2. calculate tmp theta and omega if needed

// 3. update theta
omega = omega - (g / length * Mathf.Sin(theta) * Time.deltaTime * step);
theta = theta + lastOmega * Time.deltaTime * step;
// 4. update omega
deltaTheta = theta - lastTheta;
// 5. move the object to the new postion
```

Midpoint

使用位置中点的速度

$$x(t_0 + h) = x_0 + hf\left(x_0 + \frac{h}{2} f(x_0)\right)$$



```
// the difference in angles during this time step
float deltaTheta = 0;
// 1. save the theta and omega of last time step for later use
float lastTheta = theta;
float lastOmega = omega;

float tmpOmega = lastOmega - 0.5f * g * Mathf.Sin(lastTheta) / length * Time.deltaTime * step;
float tmpTheta = lastTheta + 0.5f * lastOmega * Time.deltaTime * step;

theta = theta + tmpOmega * Time.deltaTime * step;

omega = omega - g * Mathf.Sin(tmpTheta) * Time.deltaTime * step / length;//使用tmpTheta更新omega
deltaTheta = theta - lastTheta;
```

Trapezoid

使用时间中点的速度

$$x(t_0 + h) = x_0 + h\frac{f(x_0) + f(x_0 + hf(x_0))}{2}$$

```
// the difference in angles during this time step
float deltaTheta = 0;
// 1. save the theta and omega of last time step for later use
float lastTheta = theta;
float lastOmega = omega;

float tmpOmega = lastOmega - g * Mathf.Sin(lastTheta) / length * Time.deltaTime * step;
float tmpTheta = lastTheta +  lastOmega * Time.deltaTime * step;

float midOmega = (lastOmega + tmpOmega) / 2;
float midTheta = (lastTheta + tmpTheta) / 2;
theta = theta + midOmega * Time.deltaTime * step;

omega = omega - g * Mathf.Sin(midTheta) * Time.deltaTime * step / length;
deltaTheta = theta - lastTheta;
```