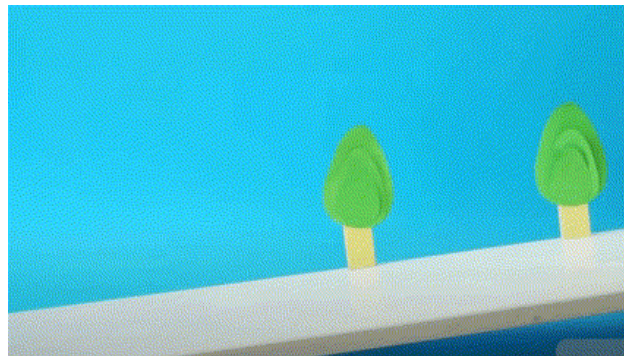




Physics trong Unity

nguyenkimlong@savameta.com

Định luật Newton



Định luật I:

Một vật không chịu tác dụng của lực hoặc hợp lực bằng 0, sẽ giữ nguyên trạng thái đứng yên hoặc chuyển động thẳng đều

Định luật II:

Gia tốc của vật tỉ lệ thuận với lực và tỉ lệ nghịch với khối lượng của vật: $\vec{F} = m\vec{a}$

Định luật III:

Khi một vật tác dụng lực lên vật thứ hai, vật thứ hai tác dụng một lực cùng độ lớn và ngược chiều về phía vật thứ nhất



Lực

Trong tự nhiên, mọi vật muốn di chuyển được phải có lực tác dụng lên nó

Lực có thể được tính bằng nhiều cách

$$\vec{F} = m\vec{a} = m\frac{\vec{v}}{t} = \frac{\vec{p}}{t}$$

Cách đơn giản nhất là lấy khối lượng nhân với vector vận tốc và chia cho thời gian

Qua công thức trên, ta có thể thấy rằng nếu vật càng nặng, mà lực tác dụng lại không đổi, thì vận tốc của vật cũng không tăng

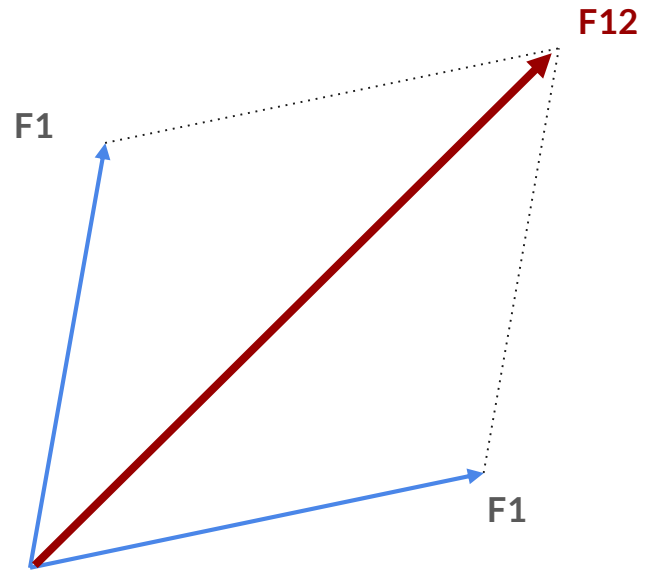
Hợp lực

Lực là đại lượng vector nên chúng có thể cộng lại với nhau - hợp lực

Hợp lực của chúng tuân theo quy tắc hình bình hành (như hình bên)

Ngoài ra, hướng của hợp lực chính là hướng di chuyển của vật

$$\vec{v} = \frac{\vec{F}t}{m}$$

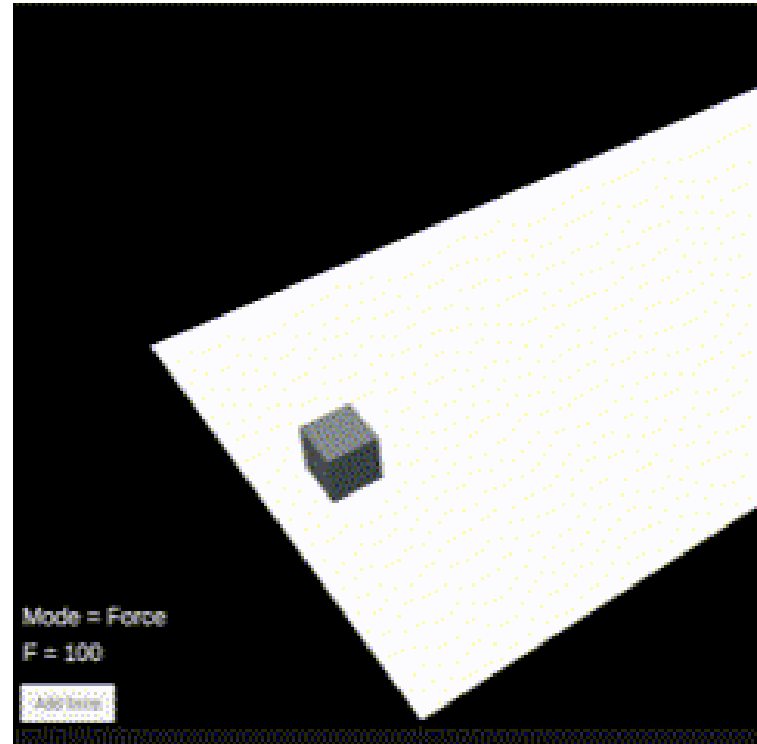


AddForce

Trong Unity, để thêm lực vào vật ta dùng phương thức **Rigidbody.AddForce**

Ngoài ra, đối tượng được thêm lực cần có một số yêu cầu sau:

- Phải chứa **Rigidbody**
- Rigidbody không được là **Kinematic**
- Phải chứa ít nhất 1 **Collider**



ForceMode

AddForce dựa trên thời gian và khối lượng

Unity cung cấp 4 chế độ để người dùng tiện sử dụng

Nhìn chung, Force và Acceleration được viết trong Update(), còn Impulse và VelocityChange thì không

ForceMode

Know the Difference

for `rigidBody.AddForce(f, ForceMode)*`:

	time dependent	time independent	
mass dependent	Force same as: $v += f * dt / m$	Impulse same as: $v += f / m$	Use these when force depends on mass. E.g. the same amount of gunpowder fires a small bullet at a high velocity, and a big one at a low velocity.
mass independent	Acceleration same as: $v += f * dt$	VelocityChange same as: $v += f$	Use these when the mass doesn't affect the change of velocity. An example is gravity, which accelerates all objects equally.
	Use these when you want to apply a force continuously, each frame, like a rocket engine, jetpack, magnet..	Use these when you want to apply an instantaneous force, such as on collision reponse, shooting a bullet, kicking a ball..	

* The same rule applies to all related functions that use ForceMode as a parameter

PID

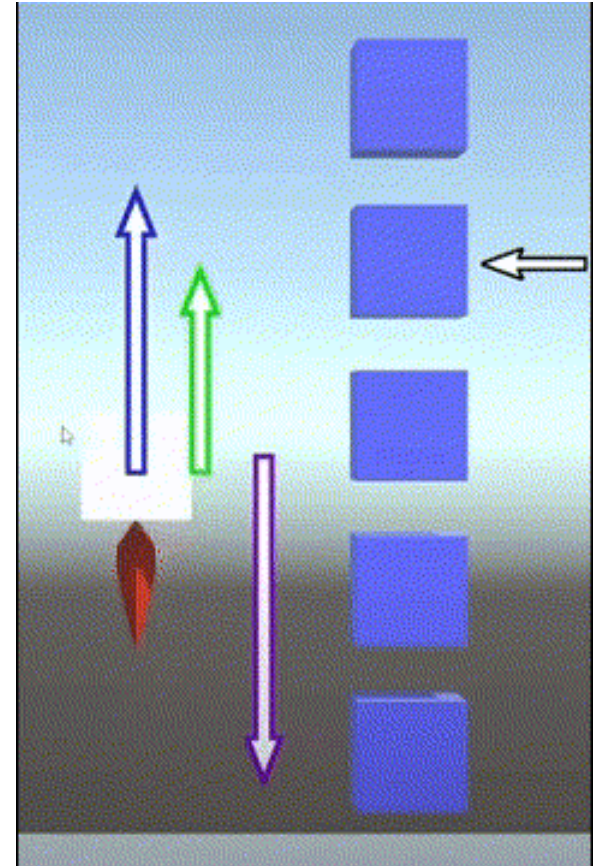
PID = Proportional Integral Derivative

PID là cách để giảm tối đa sai số bằng việc điều chỉnh giá trị đầu vào

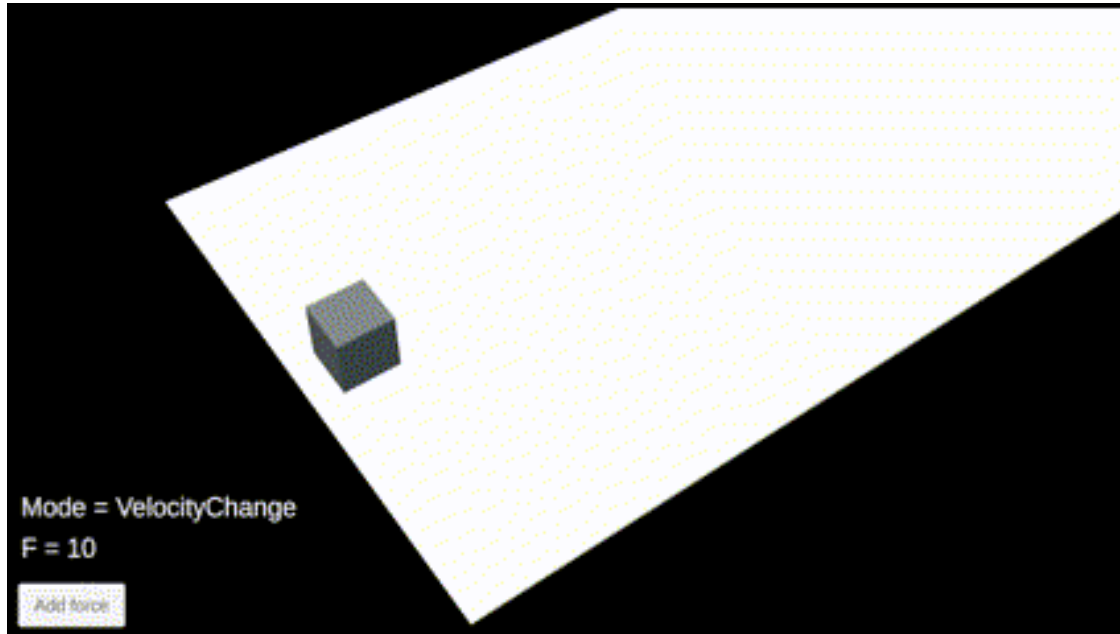
PID có rất nhiều ứng dụng trong Unity: máy bay phản lực, tên lửa đuổi, xoay đầu xe tăng, quay người khi chạy nhanh,...

* Một video giải thích rất hay về PID:

<https://www.youtube.com/watch?v=y3K6FUgrgXw>



Velocity



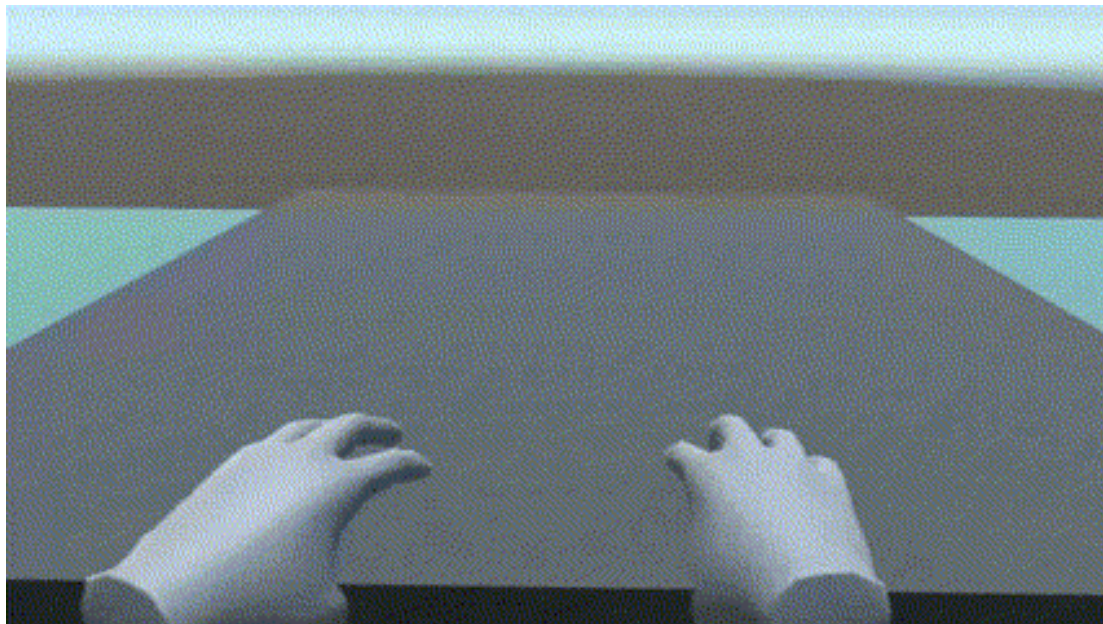
Điều khiển việc di chuyển vật thể bằng `AddForce` có vẻ khó khăn, liệu có cách nào đơn giản hơn mà vẫn giữ được tính vật lý của Unity?

Thật may mắn, kết quả cuối cùng của các phép tính `AddForce` là `Velocity`, nên ta có thể truyền trực tiếp `Velocity` vào trong `Rigidbody`

Tuy nhiên, cách này sẽ không tính đến khối lượng của vật nên chỉ dùng cho các trường hợp như lực hút trái đất hoặc cầm nắm vật thể trong VR

Velocity follower with Hand VR

Chỉ cần thay target thành controller và follower thành bàn tay trong VR là được



Grab with Hand VR

Ta cần lưu lại offset giữa tay và vật

Nếu ta di chuyển vật theo tay thì sẽ xảy ra trường hợp vật xuyên qua tường

Do vậy, giải pháp là ta lưu lại offset giữa tay và vật, rồi cho tay di chuyển theo vật, vật di chuyển theo controller

