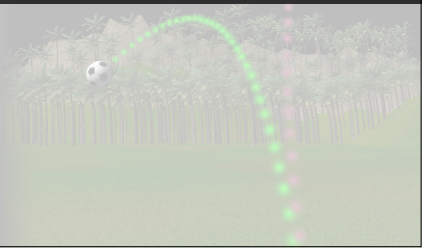
Introduction To Section 3

In this section you will learn about…

- About “moments of inertia”.
- The rotational equivalent of $F = ma$
- Parallel and Intermediate Axis Theorems.
- Calculating rotation from torques.
- Tensors and Einstein notation.

Unstable Rotation In Unity 5



In this video...

- The limitations of spinning objects in Unity 5.
- Why unstable rotation doesn’t evolve.
- How to model unstable rotation in game physics.

Spin a phone-shaped cuboid

- Create a cuboid with scale (70, 140, 7).
- Rotate it 2 degrees about y-axis.
- Add a rigid body, ignore gravity, 0 angular drag.
- Give it **angularVelocity = (4f, 0, 0)**
- Observe that the resulting spin is stable.

Introducing Inertia Tensors

In this video...

- Roughly what is an inertia tensor (**I**)?
- Unstable intermediate axis
- How Unity Calculates **I**
- Calculate **I** for your phone
- Predict it's unstable axis

Unstable Intermediate Axis

- Look at the three components of I*
- Does one axis have an *intermediate* value?
- If so, the rotation will be unstable about this axis.

** Strictly there are 9 components as we'll see later.*

How Unity Calculates I

- Mass is specified by the parent's rigidBody.
- No child may have a rigid body.
- Child objects' colliders distribute mass evenly.
- Local axis must be aligned with symmetry.
- (Inertia tensor calculated about C.O.M.)

Inertia Tensor Of Your SmartPhone

- Lookup the dimensions and mass.
- Assume it's an exact cuboid (no rounding).
- Write down the value Unity gives you.
- Which axis does it suggest is unstable?

Calculating Moment Of Inertia

```
void ResolveForces (Vector3[] forces) {
    Vector3 combinedForce = Vector3.zero;
    foreach (Vector3 force in forces) {
        combinedForce += force;
    }
    float deltaT = Time.fixedDeltaTime;
    acc = combinedForce / mass;
    vel += acc * deltaT;
    pos += vel * deltaT;
    if (isEnabled) transform.position = pos;
}
```

In this video...

- How about if two axis are equal?
- Try it with a spinning cylinder.
- Calculate the moment of inertia.
- Render vs. collider mesh influence.

Try spinning a cylinder

- Spin a cylinder around each axis.
- Use **SpinRite.cs** to see if it's stable in all 3.

Calculate your cylinder's value

- Calculate the moment about x (or z)*
- See if it matches Unity's value.
- If not, try and explain why.

* *http://en.wikipedia.org/wiki/List_of_moments_of_inertia*

Parallel Axis Theorem

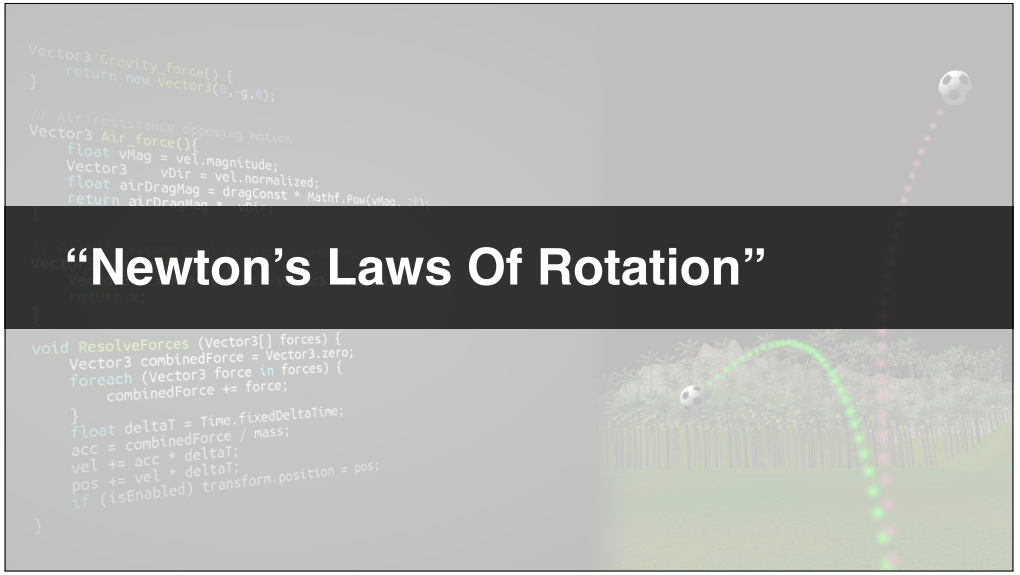
In this video...

- NOT the same as Intermediate Axis Theorem.
- How to calculate I for compound objects.
- Think of a discus thrower.

http://en.wikipedia.org/wiki/Parallel_axis_theorem

Work out I for two balls

- Use Unity to calculate I for these two balls.
- One with mass 100 at (0,0,0).
- Another with mass 10 at (1,1,0).
- What's the value of I Unity calculates?
- What's the Centre Of Gravity of the system?



In this video...

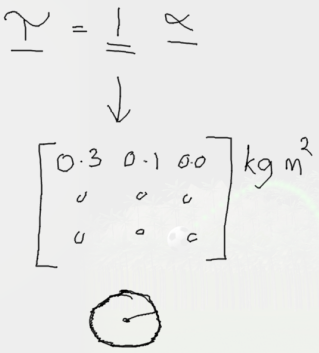
- A recap of Newton's laws of motion.
- How these relation to rotation.
- Important differences.
- Why we're jumping in the deep end!

Draw your own parallels

Substitute with torque, angular velocity, angular acceleration and inertia tensor…

1. With no net **force, velocity** remains constant.
2. **Force** = **mass** * **acceleration**
3. Every **force** has an equal and opposite **force**.

Rotational equivalent of F = ma



Applying Torque To Rigidbodies

In This Video...

- How to use Unity's **AddTorque()** method.
- How Inertia Tensor affects rotation.
- Comparing the rotation of two objects.

Match the Inertia tensor

- Scale the 2nd sphere so it has the same I value.
- Apply the same torque to both spheres.
- Observe the relative rotation.

Vector Cross Products

In This Video...

- What a cross product is.
- Why they are so useful.
- Using WolframAlpha to calculate.

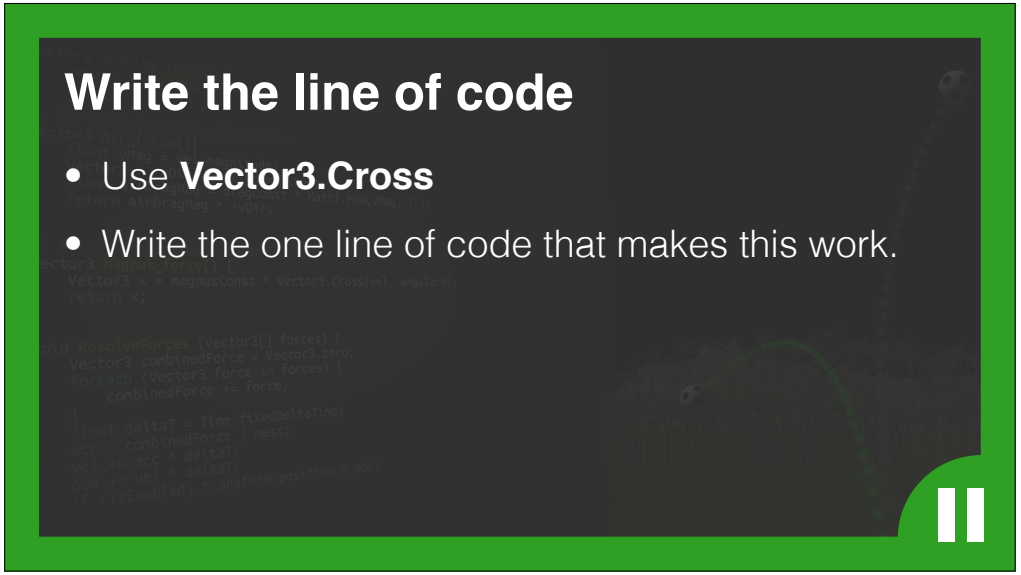
The Magnus Effect

In This Video...

- What the Magnus Effect is.
- How we can approximate ball flight with it.
- Using **Vector3.Cross** in Unity.

Write the line of code

- Use **Vector3.Cross**
- Write the one line of code that makes this work.



Section Wrap-Up