Bullet Constraints

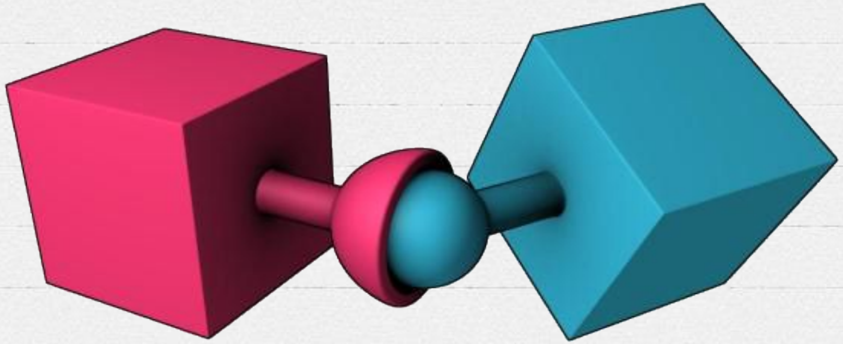
Constraints (or joints)

Bullet presents five types of constraints:

- Point to Point (Distance joint)
- Hinge (Revolute joint) / Spring Hinge
- Slider (Prismatic joint)
- Cone Twist
- Six Degrees-of-freedom / Spring Six Degrees-of-freedom

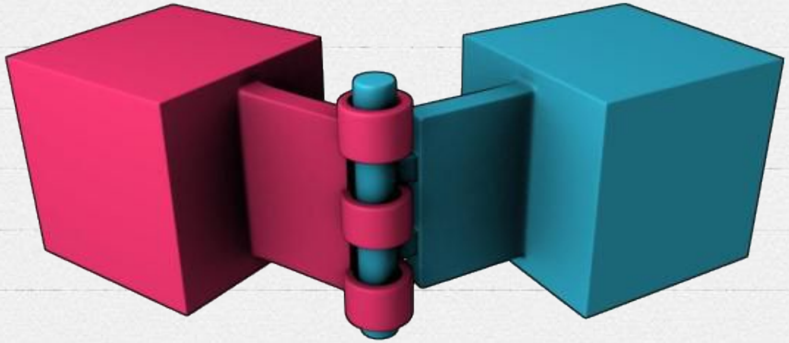
Point to point

The Point constraint (called a point-to-point constraint in the Bullet Physics library) limits the translation so that pivot points between the two rigid bodies match in world space. You can use the Point constraint to create effects, such as a chain-link, or to pin objects together.



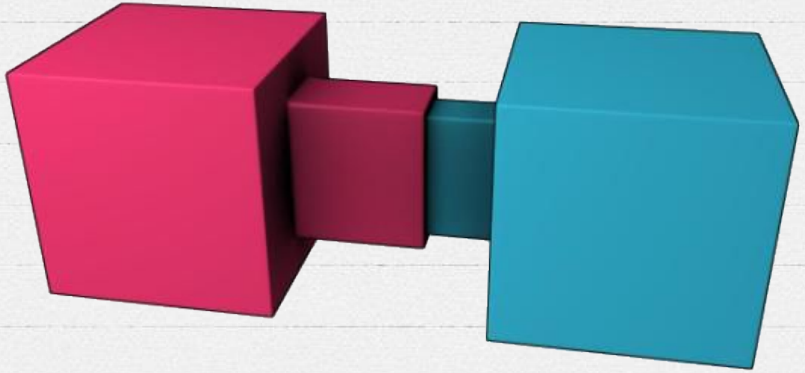
Hinge

The Hinge constraint restricts the translation and two additional angular degrees of freedom, so the body can only rotate around one axis. The hinge axis is defined by the Z axis of the constraint. This constraint is useful for representing doors or wheels rotating around an axis. The user can specify limits and motor settings for the hinge.



Slider

The Slider constraint allows rigid bodies to rotate around one axis and translate along the same axis. The slide axis is defined by the Z axis of the constraint.

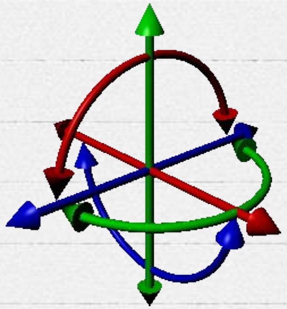


Cone twist

For ragdolls, the Cone-Twist constraint is useful for limbs like the upper arm. It is a special point-to-point constraint that adds cone and twist axis limits. The X axis serves as twist axis.

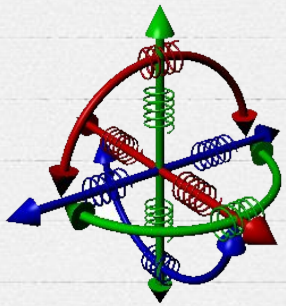
Six Degrees-of-freedom

The Six Degrees-Of-Freedom (SixDOF) constraint can emulate a variety of standard constraints if each of the six Degrees of Freedom (DOF) is configured. The first 3 DOFs axis are linear axis, which represent the translation of rigid bodies, while the latter 3 DOFs axis represent the angular motion. Each axis can be locked, free, or limited. By default, all axes are unlocked.



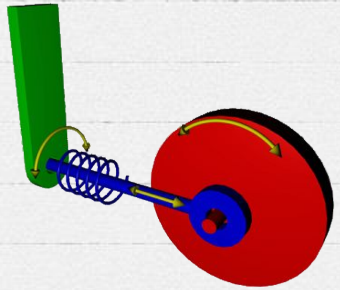
Spring Six Degrees-of-freedom

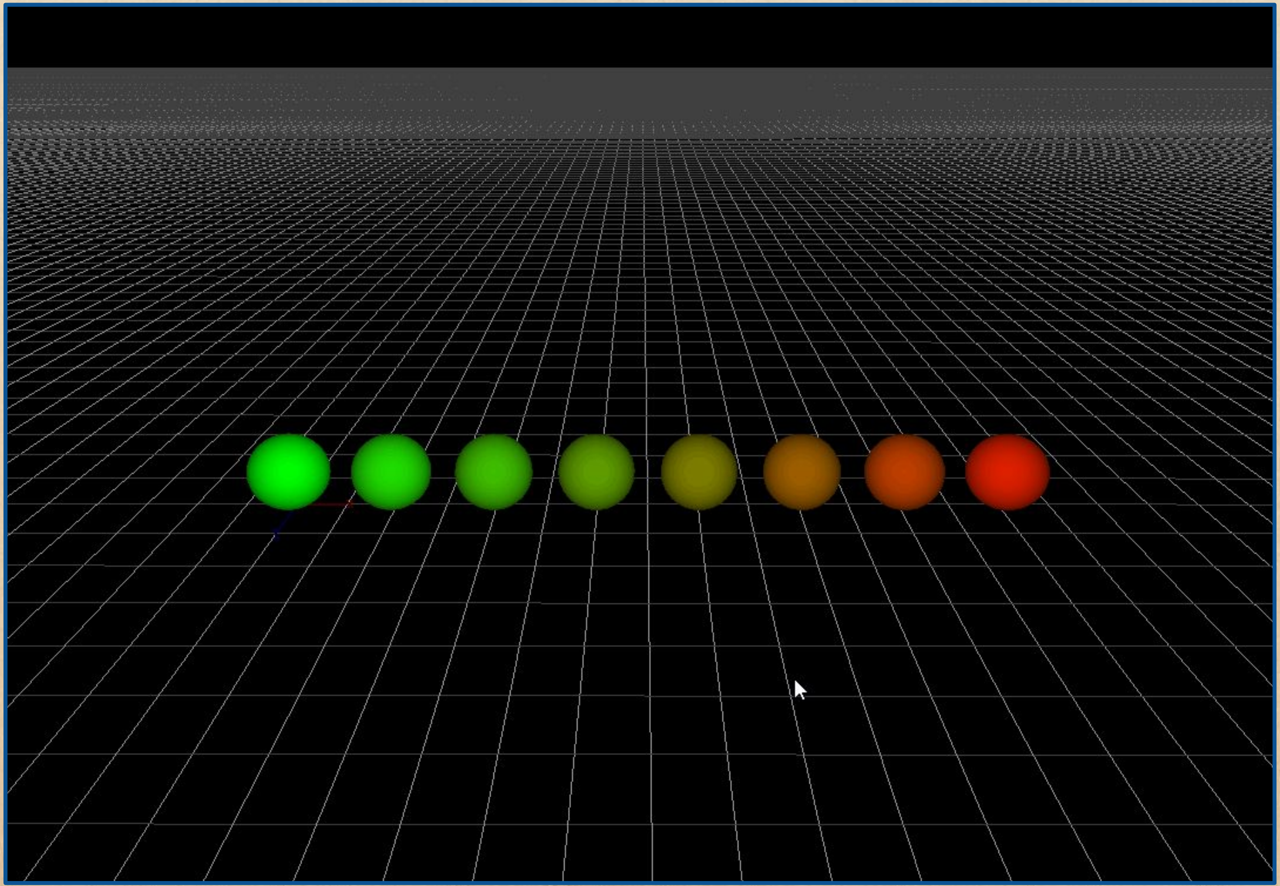
The Spring Six Degrees-Of-Freedom (SpringSixDOF) constraint is a variant of the Six Degrees-of-Freedom constraint that includes the addition of springs for each of the degrees of freedom. Springs and motors cannot be combined on this constraint.



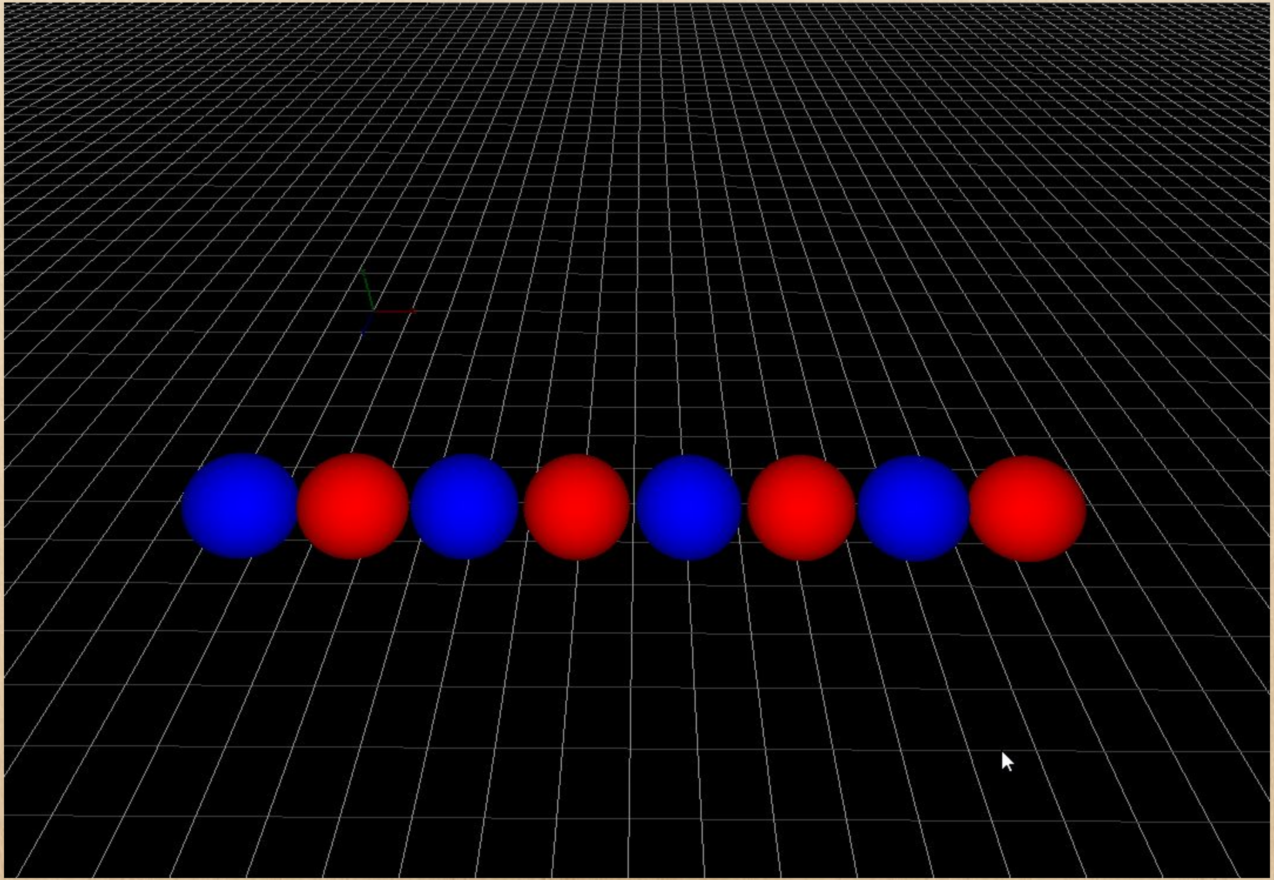
Spring Hinge

The Spring Hinge has three degrees-of-freedom. These include two rotational degrees-of-freedom around Z (Axis 1) and X (Axis 2), and one translational along Z (Axis 1) with a suspension spring. A use-case example of a Spring Hinge constraint is the steering wheel of a car, where one axis allows the wheel to be steered while the other axis allows the wheel to rotate. For this case, you need to rotate the spring hinge -90 in X to align Axis 1 with Y.





Our First goal



Our second goal

YOUR TURN !

TODO Nº 1

```
// TODO 1: Code a method that adds a point 2
point constraint between two bodies
```

The signature of the method could be something similar to:

```
void AddConstraintP2P(PhysBody3D& bodyA, PhysBody3D& bodyB, const vec3& anchorA, const vec3& anchorB);
```

The Bullet call must be:

```
btPoint2PointConstraint(btRigidBody& rbA, btRigidBody& rbB, const btVector3& pivotInA, const btVector3& pivotInB);
```

And then, you will add this constraint with world->addConstraint(). Keep the constraint on a list to delete it afterwards in the CleanUp().

TODO Nº 2

```
// TODO 2: Chain few N spheres together to form a snake
```

Now, create the snake/Marge's pearl necklace. Start with two spheres and link them. Then, use loops and the already created arrays on *ModuleSceneIntro* to built it.

TODO Nº 3

```
// TODO 3: Code a method that adds a hinge
constraint between two bodies
```

The signature of the method could be something similar to:

```
void AddConstraintHinge(PhysBody3D& bodyA, PhysBody3D& bodyB, const vec3& anchorA, const vec3& anchorB,
const vec3& axisA, const vec3& axisB);
```

The Bullet call must be:

```
btHingeConstraint(btRigidBody& rbA,btRigidBody& rbB, const btVector3& pivotInA, const btVector3& pivotInB,
btVector3& axisInA, btVector3& axisInB);
```

And then, you will add this constraint with world->addConstraint(). Keep the constraint on a list to delete it afterwards in the CleanUp().

TODO № 4

```
// TODO 4: Chain few N spheres together to form a bike's
chain. Be sure to put the right axis
```

Use the same procedure as in TODO 2. Experiment with the different axis to match the solution.

Homework (high level)

Create a mouse joint!

Next Week . . .

Vehicle Creation