

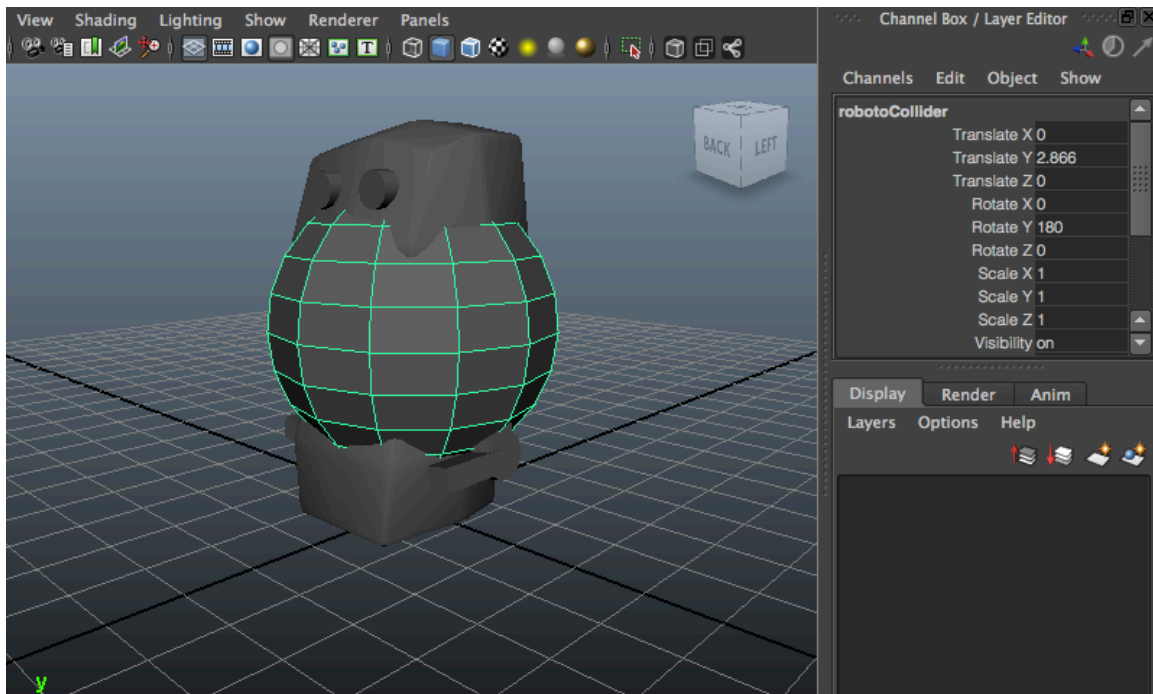
Setting up Panda Collisions in Maya:

In this example I've created robot with two collision spheres. One sphere is a collision sphere for obstacles and walls. The other sphere is for collisions with objects that the robot will pick up.

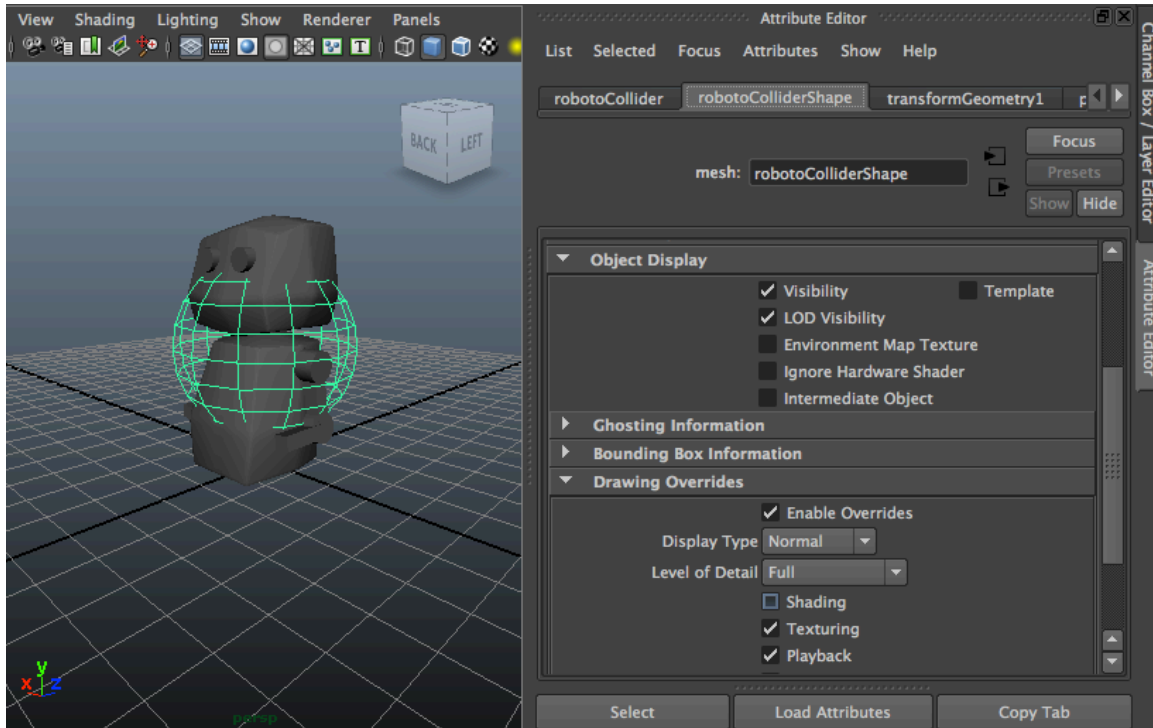
The first thing is start out with a robot model



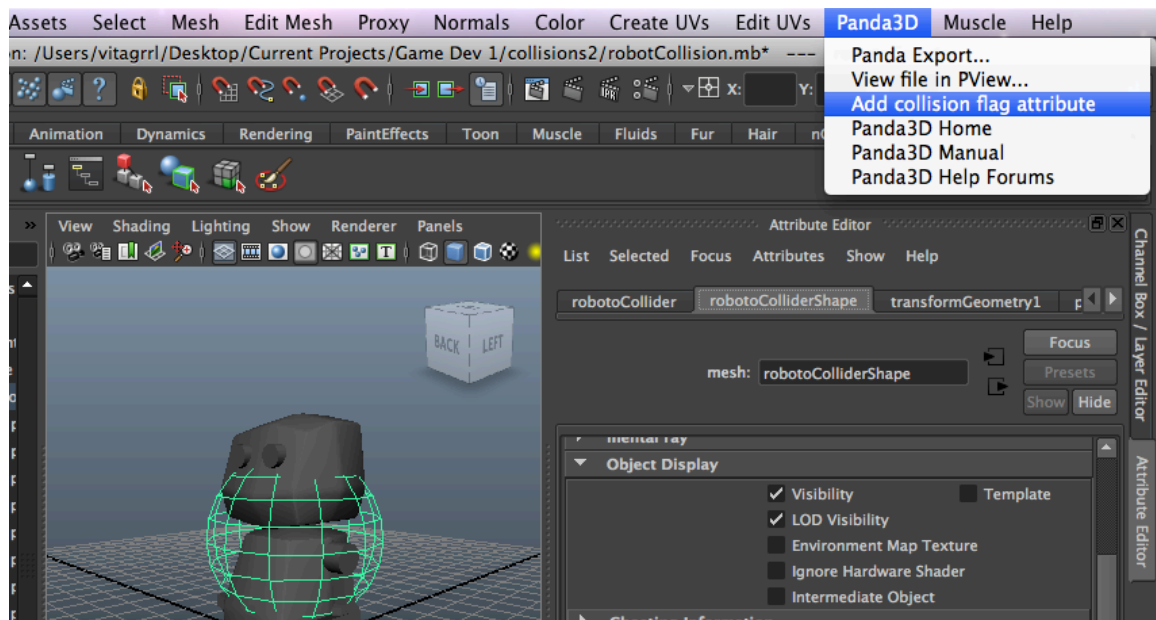
Then you need to create a collision sphere. I've named this sphere robotoCollider.



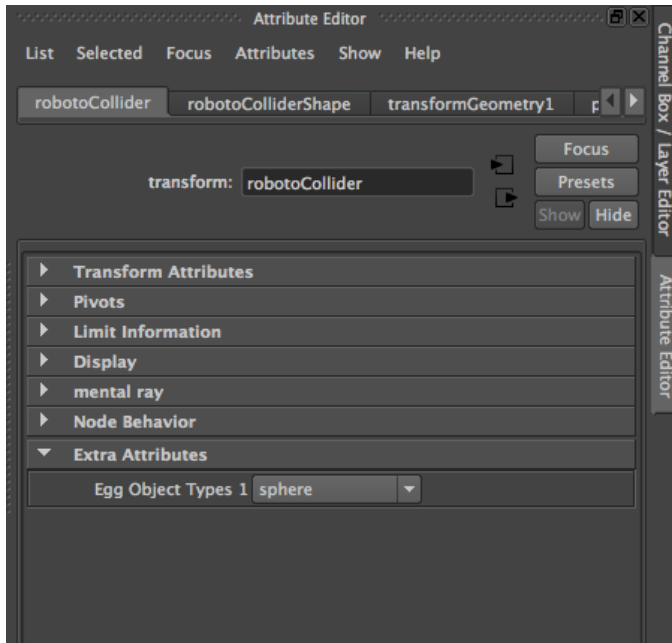
This sphere needs to be invisible so I next need to go into the attribute editor and select the roboColliderShape tab. Next, go into Object Display and in Drawing Overrides, check Enable Overrides and uncheck shading.



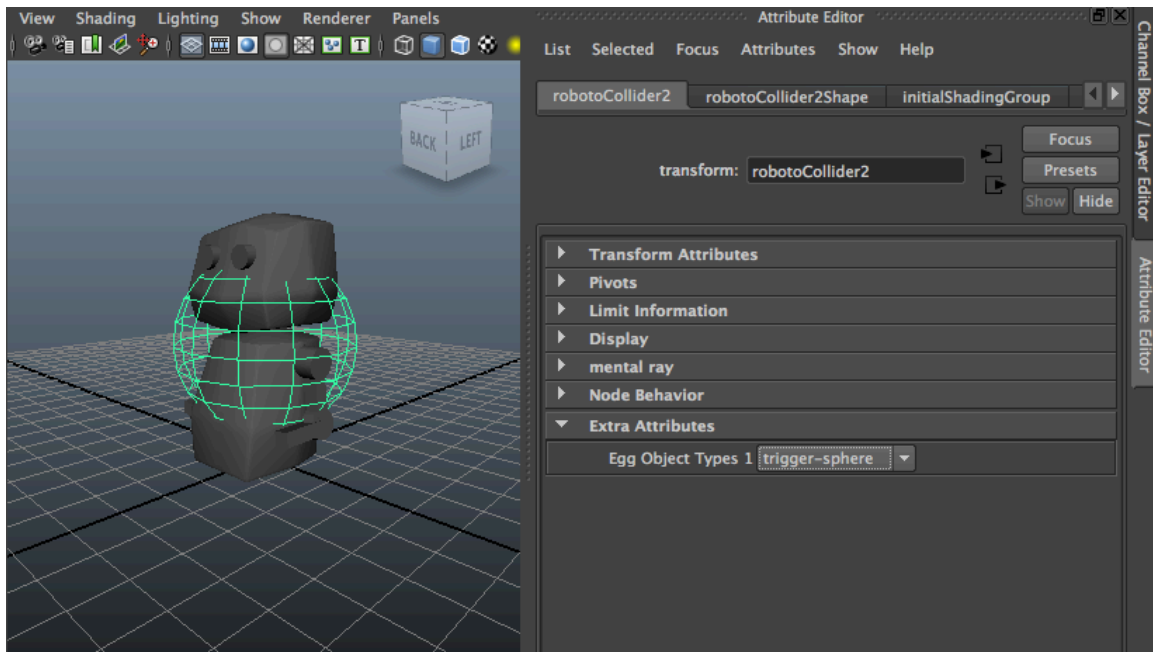
Now go into the Panda3d menu and select Add collision flag attribute.



This will add an extra attribute for the roboCollider. Set this attribute to sphere, if this is for regular collisions and trigger-sphere if this sphere is to trigger an event after collision.



Now that I've completed this for the collision sphere, I will duplicate this sphere for the trigger collision sphere. This sphere is roboCollider2.



Now export from Maya

Panda Exporter

Export Type	Animation Options
<input checked="" type="radio"/> Mesh <input type="radio"/> Actor <input type="radio"/> Animation <input type="radio"/> Both <input type="radio"/> Pose	<input type="radio"/> Full <input checked="" type="radio"/> Custom
Export Options	Start Frame 0
<input type="checkbox"/> Export only selected objects	End Frame 48
<input checked="" type="checkbox"/> Double sided faces	
<input checked="" type="checkbox"/> Overwrite existing files	
<input type="checkbox"/> Run PView after export	
<input type="checkbox"/> Only legacy shaders	
<input checked="" type="checkbox"/> Keep all UV's	
<input checked="" type="checkbox"/> Round UV's	
<input checked="" type="checkbox"/> Tangents+Binormals for all UV's	
Convert units from: cm	
Output File Options: <input type="radio"/> EGG (ASCII) <input type="radio"/> BAM (Binary) <input type="radio"/> Both	
Texture Path Options: <input type="radio"/> Reference textures relative to maya file (default) <input type="radio"/> Reference textures relative to custom path Specify alternate directory: top/Current Projects/Game Dev 1/collisions2/textures Browse	
Output filename options: <input type="radio"/> Export to root directory of source file (default) <input type="radio"/> Export to other directory: grrl/Desktop/Current Projects/Game Dev 1/collisions2 Browse	
<input type="radio"/> Use original filename <input type="radio"/> Use alternate filename roboto2 Browse	
Save transform: <input type="radio"/> all <input type="radio"/> model/DCS flag	
EXPORT	

In the Panda Code:

```
# load the robot model
self.robot=loader.loadModel("roboto2.egg")
self.robot.reparentTo(render)

def setupCollisions(self):
    """
        set up collision system, a Pusher for the walls, and an Event
        handler for things to pick up
    """
    # create a CollisionTraverser and make it show collisions when they
    # happen (for debugging)
    base.cTrav = CollisionTraverser()
    base.cTrav.showCollisions(render)

    # the CollisionHandlerPusher automatically prevents you from walking
    # through walls
    self.collisionPusher = CollisionHandlerPusher()

    # find the two collision sphere nodes in the robot model
    # one is for testing collisions with walls, the other is for testing
    # collisions with the pyramid items

    robotCollisionNode1 = self.robot.find("**/robotoCollider")
    robotCollisionNode2 = self.robot.find("**/robotoCollider2")

    # disable their collide masks: otherwise they will collide with each
    # other
    robotCollisionNode1.setCollideMask(BitMask32(0x00))
    robotCollisionNode2.setCollideMask(BitMask32(0x00))

    # add the first collision sphere to the Pusher handler and tell it to
    # control the robot model
    # if robotCollisionNode1 collides with the walls or the obstacles, the
    # Pusher moves the robot away from the wall
    self.collisionPusher.addCollider(robotCollisionNode1,self.robot)
    base.cTrav.addCollider(robotCollisionNode1, self.collisionPusher)

    # create an event handler to trigger events when you touch a pyramid
    self.notifier = CollisionHandlerEvent()
    self.notifier.addInPattern("%fn-in-%in")
    self.accept("robotoCollider2-in-pyramidTrigger",self.onCollision)

    base.cTrav.addCollider(robotCollisionNode2,self.notifier)
```

These are the names
I gave the collision spheres
in Maya

