# irrBullet

```
 _____ _____ _____ 
|   __|  _  |     |
|   __|     |  |  |
|__|  |__|__|__  _|
               |__|
```

## Index:

1.0 Beginner:

- 1.1 My GImpactMeshShape doesn't collide with other objects. What's the problem?
- 1.2 How do I change the position or rotation of one of my objects?
- 1.3 Is it possible to have sub-objects as children of another rigid body?
- 1.4 Which collision shapes should I use for mesh scene nodes?
- 1.5 How do I apply a local force? (for instance, in the direction it's pointing)
- 1.6 How do I apply a local angular force? (for instance, yaw, pitch, and roll)
- 1.7 How do I use constraints?
- 1.8 Sometimes high-speed objects go right through other objects. How do I prevent this?
- 1.9 When my raycast vehicle sits in the same spot for a couple of seconds, it won't move anymore. Why does this happen?
- 1.10 My raycast vehicle's wheels get stuck under the ground sometimes when it lands. How can I stop it from doing this?
- 1.11 Why is there no wrapper interface for constraints?
- 1.12 When I try to use irrBullet in my project, I get undefined references to Bullet functions. What am I doing wrong?
- 1.13 In what order do I have to link the libraries?
- 1.14 How do I prepare my project to use irrBullet?
- 1.15 How do I use collision masking?
- 1.16 As advised, I put the hardpoint of my raycast vehicle's

wheels inside the rigidbody chassis, but the debug drawer shows them as colliding with the chassis itself and so they never touch anything outside of the chassis. How do I solve this?

2.0 Intermediate:

- 2.1 When a collision occurs, how do I find which two objects it was between?
- 2.2 How do I make ragdolls?
- 2.3 How do I make my own object affectors?

3.0 Advanced:

# Content:

## 1.0 Beginner:

1.1

Q: My GImpactMeshShape doesn't collide with other objects. What's the problem?

A: If your GImpactMeshShape doesn't collide with other objects, it could be that you set the second parameter of the function createIrrBulletWorld to false.

If you want to use GImpactMeshShapes, you must set the second parameter of the function to true, which tells irrBullet that it should register the GImpact collision algorithm.

If this doesn't work and you are using collision masking/filters, make sure your filter system is setup to allow the objects to collide with eachother.

1.2

A: How do I change the position or rotation of one of my objects?

Q: Changing the world transform of an object is easy with irrBullet.

Here is an example:

```
irr::core::matrix4 mat;

mat.setTranslation(irr::core::vector3df(0,1000,1000));
mat.setRotationDegrees(irr::core::vector3df(0,0,0));

object->setWorldTransform(mat);
```

1.3

Q: Is it possible to have other objects as children of a rigid body?

A: Yes. This should not be confused with the compound shape, which allows multiple shapes to be added into one shape.

The method used in the game irrBullet was created for uses a base class that handles the child objects of a main rigid body and keeps them together with Bullet constraints.

Because each game might need a different way to handle child objects, irrBullet doesn't have an interface for this. It should be implemented by the game developer to fit his/her game's needs.

See section 1.11 in this FAQ list about constraints for more information on how to use Bullet constraints.

1.4

Q: Which collision shapes should I use for mesh scene nodes?

A: The mesh collision shape you should choose varies depending on the object's role in the game:

If an object is dynamic (for instance, a player's vehicle), you could choose:

- a compound collision shape with cubes, spheres, and other primitives to approximate the shape.

- an IConvexHullShape for convex mesh shapes (cubes, spheres, cylinders and certain other shapes are convex)

- an IGImpactMeshShape for concave mesh shapes (this could be useful for a lot of different things)

- If an object is meant to be static, you can use the IBvhTriangleMeshShape, which was created solely for static triangle mesh shapes like terrain and level geometry.

1.5

Q: How do I apply a local force? (for instance, in the direction it's pointing)

A: Every method in IRigidBody that deals with velocities and impulses has a parameter to set the space in which to apply it (world space or local space).

For example, this will apply a central force of 1000 in the direction a rigid body is pointing:

```
body->applyCentralForce(vector3df(0,0,1000.0f), ERBTS_LOCAL);
```

This will apply a central force of 1000 upward on a rigid body no matter which direction it is pointing:

```
body->applyCentralForce(vector3df(0,0,1000.0f), ERBTS_WORLD);
```

By default, the second parameter is ERBTS_WORLD.

1.6

Q: How do I apply a local angular force? (for instance, yaw, pitch, and roll)

A: Applying a local angular force is just as easy as applying a local linear force.

In this code snippet, a rigid body is receiving an angular force of 10 on the X axis. If this were an aircraft, it would change the attitude of it by torque (the same effect of elevons and elevators).

```
body->applyTorque(vector3df(0,0,10), ERBTS_LOCAL);
```

This, on the other hand, will apply an angular force of 10 on the X axis in world space.

```
body->applyTorque(vector3df(0,0,10), ERBTS_WORLD);
```

By default, the second parameter is ERBTS_WORLD.

1.7

Q: How do I use constraints?

A: irrBullet does not provide an interface for constraints for a couple reasons: unnecessary overhead and memory usage.

irrBullet handles the de-allocation of the memory of referenced constraints in the destructor of each IRigidBody, so the developer only has to add the constraint to the IRigidBody's btRigidBody pointer (retrieved through IRigidBody::getPointer()).

This might seem to add complications, but it's really very simple.

This way was chosen because each game (including the one irrBullet was designed for) would typically need its own classes for objects and their children, and with irrBullet's helper functions, mixing pure Bullet functions with irrBullet functions is quite simple and straight-forward.

Here is an example that adds a generic 6DoF spring constraint between two rigid bodies:

```
btTransform frameInA, frameInB;
    frameInA = btTransform::getIdentity();
    frameInA.setOrigin(irrlichtToBulletVector(location));
    frameInB = btTransform::getIdentity();
    frameInB.setOrigin(irrlichtToBulletVector(rotation));

btRigidBody* bodyA = parentObject->getRigidBody()->getPointer();
btRigidBody* bodyB = obj->getRigidBody()->getPointer();
btGeneric6DofSpringConstraint* pGen6DOF = new
btGeneric6DofSpringConstraint(*bodyA, *bodyB,
        frameInA, frameInB, true);

pGen6DOF-
>setLinearUpperLimit(irrlichtToBulletVector(linearUpperLimit));
pGen6DOF-
>setLinearLowerLimit(irrlichtToBulletVector(linearLowerLimit)); /
/vector3df(0,0,-0.5)

pGen6DOF-
>setAngularUpperLimit(irrlichtToBulletVector(angularUpperLimit));
pGen6DOF-
>setAngularLowerLimit(irrlichtToBulletVector(angularLowerLimit));


DynamicsWorld->getPointer()->addConstraint(pGen6DOF, true);
```

Example for a hinge constraint:

```
btTransform frameInA, frameInB;
    frameInA = btTransform::getIdentity();
    frameInA.setOrigin(irrlichtToBulletVector(location));
    frameInB = btTransform::getIdentity();
    frameInB.setOrigin(irrlichtToBulletVector(rotation));

btRigidBody* bodyA = parentObject->getRigidBody()->getPointer();
btRigidBody* bodyB = obj->getRigidBody()->getPointer();

btVector3 axis = irrlichtToBulletVector(hingeAxis); //
vector3df(0,1,0)

btHingeConstraint* pGen6DOF = new btHingeConstraint(*bodyA,
*bodyB, frameInA.getOrigin(),
        frameInB.getOrigin(), axis, axis, true);

// If you want to use motors
//pGen6DOF->enableAngularMotor(true, 0.2f, 30.0f);

//pGen6DOF->setMaxMotorImpulse(30.0f);
//pGen6DOF->setMotorTarget(180, 100);

DynamicsWorld->getPointer()->addConstraint(pGen6DOF, true);
```

Constraints added to a rigidbody can be retrieved like so:

```
btTypedConstraint* constraint = body->getPointer()-
>getConstraintRef(index);
```

They can also be removed like so:

```
body->getPointer()->removeConstraintRef(constraint);
```

1.8
Q: Sometimes high-speed objects go right through other objects. How do I prevent this?

A: This problem occurs in all physics engines.

It can be prevented by custom user methods, for instance, ray-casting from the last position to the new update position and checking for collision with geometry. But, fortunately, Bullet also provides us with a way to solve this using its own techniques.

It's called motion clamping.

In order to avoid missing collisions for fast objects, you need to set the motion clamping sphere radius and the motion threshold.

In irrBullet you do this with IRigidBody::setCcdValues(), which takes two floating point values as parameters.

It will usually take a bit of experimentation to get it working as desired without "tunneling" effects. The values may need changed based on mass, speed, size, etc.

Also, keep in mind that it depends how dense the geometry is for a certain object's collision shape.

For example, a box shape is more likely to tunnel than a sphere or cylinder shape, because the IBoxShape geometry includes on four vertices on which to test penetrations, so once the object is over half-way penetrated into a surface, it'll go through the other side because there are no central vertices to test.

1.9
Q: When my raycast vehicle sits in the same spot for a couple of seconds, it won't move anymore. Why does this happen?

A: The reason this happens is if you didn't set your "chassis" (the rigidbody that the vehicle affects) activation state properly.

When using a rigidbody as a vehicle chassis, you must disable deactivation.

Here is an example:

```
body->setActivationState(EAS_DISABLE_DEACTIVATION);
```

1.10
Q: My raycast vehicle's wheels get stuck under the ground sometimes when it lands. How can I stop it from doing this?

A: This could happen for more than one reason.

The most common reason is that your raycast vehicle's wheels are not coming from "within" the rigidbody chassis. This is to prevent the wheels from going beneath (or inside) other objects, preventing it from further movement (or a lot of "jittery" movement around the place the wheel is stuck in).

The other reason may be a simple penetration problem from faces of your collision mesh (in the case of a triangle mesh shape like GImpact) penetrating the level geometry, causing the object to get stuck in that position.

So, the simple answer is to make sure no such protruding faces exist in your rigidbody's collision shape, and make sure all wheels have some way to prevent them from penetrating the level geometry (either by having them come from the rigidbody chassis or by using a custom constraint to some other custom method).

1.11
Q: Why is there no wrapper interface for constraints?

A: irrBullet does not provide an interface for constraints for a couple reasons: unnecessary overhead and memory usage.

irrBullet handles the de-allocation of the memory of referenced constraints in the destructor of each IRigidBody, so the developer only has to add the constraint to the IRigidBody's btRigidBody pointer (retrieved through IRigidBody::getPointer()).

This might seem to add complications, but it's really very simple.

This way was chosen because each game (including the one irrBullet was designed for) would typically need its own classes for objects and their children, and with irrBullet's helper functions, mixing pure Bullet functions with irrBullet functions is quite simple and straight-forward.

1.12
Q: When I try to use irrBullet in my project, I get undefined references to Bullet functions. What am I doing wrong?

A: Refer to sections **1.13** and **1.14**.

1.13
Q: In what order do I have to link the libraries?

A: libirrBullet.a, libbulletdynamics.a, libbulletsoftbody.a
   libGIMPACTUtils.a (if you use GImpact),
   liblinearmath.a, libbulletcollision.a

1.14
Q: How do I prepare my project to use irrBullet?

A: From readme.txt:

To begin using basic irrBullet features in your project,
you must follow these steps:

1. Add these search directories to your project:
```
%irrBullet%/source
%irrBullet%/source/bheaders
%irrBullet%/source/bheaders/Bullet
```

2. add "**#include <irrBullet.h>**" to the top of your project
with the other includes.

3. add these files to your linker (available in the libs/
folder):
**libirrBullet.a, libbulletdynamics.a, libbulletsoftbody.a**
**libGIMPACTUtils.a (if you use GImpact),**
**liblinearmath.a, libbulletcollision.a**

Make sure the linker files are in that order for irrBullet
or your project will not compile.

4. Extra:

The header files to the Bullet Physics library are included under **source/bheaders/Bullet**. A Code::Blocks project file to compile
irrBullet is under
the source/ folder, an MSVC 2010 project is included in
source/msvc/2010,
and an MSVC 2008 project is included in source/msvc/2008

1.15
Q: How do I use collision filtering?

A: Collision masking is very easy to use, thanks to Bullet's
interface.

Here is an example:

First, you should start with enumeration values of types, like
so:

```
enum ECollisionGroupTypes
{
    COL_NOTHING = 0, //<Collide with nothing
    COL_WORLD_OBJECT = BIT(1),
    COL_CHILD_OBJECT = BIT(2),
    COL_WEAPON = BIT(3),
    COL_TERRAIN = BIT(4),
    COL_VEHICLE_RAYCAST = BIT(5)
};
```

NOTE: The #define BIT is included with irrBullet

Then you need to make groups that an object will be able to collide with:

```
int worldObjectCollidesWith = COL_WORLD_OBJECT | COL_TERRAIN |
COL_WEAPON;
int childObjectCollidesWith = COL_TERRAIN | COL_WEAPON;
int weaponCollidesWith = COL_TERRAIN | COL_WORLD_OBJECT;
int terrainCollidesWith = COL_WORLD_OBJECT | COL_WEAPON |
COL_CHILD_OBJECT | COL_VEHICLE_RAYCAST;
int vehicleRaycastCollidesWith = COL_TERRAIN;
```

Then you use both the group and mask in the constructor parameters for your rigidbody ( the second parameter is the group, the third is the mask):

```
IRigidBody* body = DynamicsWorld-
>addRigidBody(shape,COL_WEAPON,weaponCollidesWith);
```

For raycast vehicles it's slightly different. You use masking and groups in the same way, but making the vehicle's raycaster realise them takes different methods:

```
vehicle->getVehicleRaycaster()->setUseFilter(true);
vehicle->setCollisionFilterMask(vehicleCollidesWith);
vehicle->setCollisionFilterGroup(COL_VEHICLE);
```

A raycast vehicle's group and mask can be changed at any time. A collision object's group can not be changed after it's constructed.

It is as simple as that!

1.16
Q: As advised, I put the hardpoint of my raycast vehicle's wheels

inside the rigidbody chassis, but the debug drawer shows them as colliding with the chassis itself and so they never touch anything outside of the chassis. How do I solve this?

A: You may need to use collision filtering to prevent this.

In the mask you use for your vehicle, simply don't put the group that your chassis' rigidbody belongs to. (and make sure all other objects have their groups and masks setup properly or nothing will collide)

Refer to section 1.15 on collision filtering to find out how.

## 2.0 Intermediate:

2.1
Q: When a collision occurs, how do I find which two objects it was between?

A: This is done (currently) by iterating through contact manifolds and collision points.

Here is an example:

```
for(int i=0; i < DynamicsWorld->getNumManifolds(); i++)
{
    ICollisionCallbackInformation *info = 0;
    info = DynamicsWorld->getCollisionCallback(i);

    int numContacts = info->getPointer()->getNumContacts();
    for(int j=0; j < numContacts; j++)
    {
        if(verifyCollisionCallback(info))
        {
            if(info->getContactPoint(j).getDistance()<1.5f &&
info->getContactPoint(j).getLifeTime() < 2.0f)
            {
                handleCollision(info, j);
            }
        }
    }
    info->getPointer()->clearManifold();
 }
```

Here is an example that shows particle effects where a collision happens (this would be placed inside handleCollision()):

```
if(info->getContactPoint(contactNumber).getAppliedImpulse() >
2.0f)
{
     createExplosion(info-
>getContactPoint(contactNumber).getPositionWorldOnA(), info-
>getContactPoint(contactNumber).getNormalWorldOnB());
}
```

2.2
Q: How do I make ragdolls?

A: There is nothing special about ragdolls. They simply use constraints to simulate animal/machine joints.

What you need to do to use ragdolls is make your own system that positions each node (usually IBoneSceneNode from an IAnimatedMeshSceneNode) to its corresponding rigidbody (which would be constraints to another rigidbody, with the torso perhaps being the base object) each update.

For most "joints" you should use a btConeTwistConstraint (it's good for shoulder joints, head, elbows, etc.)

For more help, see sections 1.7 and 1.11 on constraints.

2.3
Q: How do I make my own object affectors?

A: Make your own class derived from ICollisionObjectAffector and overload the function affectObject().

Then you create an instance of your object affector and use ICollisionObject::addAffector().

NOTE:
Since both IRigidBody and ISoftBody are derived from ICollisionObject, affectors can be used for both of them, so make sure you check (from within affectObject()) whether the object that's being affected is an IRigidBody or an ISoftBody to prevent problems and/or handle each according to its type.

## 3.0 Advanced: