NOTAS ACERCA DE BULLET

-- --

CREAR UN CUERPO RÍGIDO ESTÁTICO

```
shape.reset (new btBoxShape(halfExtents));

btTransform transform;
transform.setIdentity();
transform.setOrigin(origin);

state.reset (new btDefaultMotionState(transform));
RbConstructionPointer info(0, state.get(), shape.get());
body.reset (new btRigidBody(info));

dynamicsWorld.addRigidBody(body.get());
```

-- --

CREAR UN CUERPO RÍGIDO DINÁMICO

```
shape.reset (new btBoxShape(halfExtents));

btTransform transform;
transform.setIdentity();
transform.setOrigin(origin);

btVector3 localInertia(0, 0, 0);

shape->calculateLocalInertia (mass, localInertia);

state.reset (new btDefaultMotionState(transform));
RbConstructionPointer info(mass, state.get(), shape.get(), localInertia);
body.reset (new btRigidBody(info));

dynamicsWorld.addRigidBody(body.get());
```

-- --

EVITAR QUE BULLET DESACTIVE EL CONTROL DE CIERTOS CUERPOS RÍGIDOS

```
player->setActivationState(DISABLE_DEACTIVATION);
```

-- --

LIMITAR EL MOVIMIENTO EN CIERTOS EJES

```
platform->setLinearFactor  (btVector3(0, 1, 0));
platform->setAngularFactor (btVector3(0, 0, 0));
platform->setGravity       (btVector3(0, 0, 0));
```

-- --

CONSULTAR LOS ATRIBUTOS DE TRANSFORMACIÓN

```
btTransform transform;

platform.getMotionState ()->getWorldTransform (transform);

btScalar x = transform.getOrigin().getX()
```

-- --

APLICAR MOVIMIENTO A CUERPOS RÍGIDOS

```
platform.setLinearVelocity (btVector3(1, 0, 0));

player.applyImpulse
(
      btVector3(player.getLinearVelocity().getX (), impulse_y, 0),
      btVector3(0, 0, 0)
);
```

-- --

APLICAR LA TRANSFORMACIÓN FÍSICA A UN MODELO GRÁFICO

```
btTransform physics_transform;
physics_object.getMotionState ()->getWorldTransform (physics_transform);

glm::mat4 graphics_transform;
physics_transform.getOpenGLMatrix (glm::value_ptr(graphics_transform));

graphics_object->set_transformation (graphics_transform);
graphics_object->scale (graphics_scale.x, graphics_scale.y, graphics_scale.z);
```

-- --

EVITAR LAS RESPUESTAS A COLISIONES ENTRE CUERPOS RÍGIDOS

```
object->setIgnoreCollisionCheck(object_a, true);
```

-- --

DETERMINAR LAS COLISIONES QUE SE HAN PRODUCIDO
(de un modo trivial)

```
int manifold_count = dynamicsWorld.getDispatcher ()->getNumManifolds ();

for (int i = 0; i < manifold_count; i++)
{
    btPersistentManifold * manifold =
dynamicsWorld.getDispatcher()->getManifoldByIndexInternal (i);
    btCollisionObject * object_a = (btCollisionObject *)(manifold->getBody0 ());
    btCollisionObject * object_b = (btCollisionObject *)(manifold->getBody1 ());

    int numContacts = manifold->getNumContacts ();

    for (int j = 0; j < numContacts; j++)
    {
        btManifoldPoint & point = manifold->getContactPoint(j);

        if (point.getDistance() < 0.f)
        {
            if
             (
                    (object_a == player_body   && object_b == platform_body) ||
                    (object_a == platform_body && object_b == player_body  )
             )
            {
                // DO SOMETHING
                // object_a->setIgnoreCollisionCheck (object_b, true);
            }
        }
    }
}
```