

Survival Guide Box2D

Contactos por cuerpo

Cuando iteran sobre los contactos de un cuerpo determinado, pueden hacerlo como se vio en el apunte teórico:

```
for (b2ContactEdge* ce = myBody->GetContactList(); ce; ce = ce->next)
{
    b2Contact* c = ce->contact;
    ...
}
```

O pueden hacerlo como si fuera una lista (seguramente más adelante verán bien que son):

```
b2ContactEdge* c;
c=body->GetContactList(); //obtenemos la lista de los contactos

while(c!=NULL){ //mientras c no sea nulo hay contacto

    b2Contact* contact=c->contact
    ...
    c=c->next;
}
```

Cuando evaluán los contactos de un cuerpo determinado puede suceder que el cuerpo contenga una lista de contactos a pesar de que no haya determinado algún punto de colisión. Para chequear que efectivamente se posee una normal de colisión y un punto al menos deben mirar el miembro:

```
contact->GetManifold()->pointCount
```

si es 0 entonces no hay puntos de contacto, si es mayor que 0 entonces pueden mirar las normales.

Las normales pueden recuperarlas en coordenadas locales del cuerpo y en coordenadas globales. Para recuperarlas en coordenadas locales:

```
b2ContactEdge* c;
c=body->GetContactList();
b2Manifold *manifold;
while(c!=NULL){
    b2Contact *cont=c->contact;
    manifold=cont->GetManifold();
    b2Vec2 normal2= manifold->localNormal; //normal en coordenadas locales
    //si deseamos convertirlas a globales podemos hacer:
    normal2=body->GetWorldVector(normal2); //usamos el body para convertir a
    //globales
    c=c->next;
}
```

O sino directamente podemos obtener las coordenadas globales de la siguiente manera:

```
b2ContactEdge* c;
    c=body->GetContactList();
    b2WorldManifold manifold;
    while(c!=NULL){
        b2Contact *cont=c->contact;
        cont->GetWorldManifold(&manifold);
        b2Vec2 normal= manifold.normal;
        c=c->next;
    }
```

Listener de contactos

Un listener de contactos es una clase que recibe información de contactos que se producen y permite actuar en consecuencia de estos. Por ejemplo en nuestro juego podríamos necesitar que cuando nuestro proyectil colisiones con un enemigo este pierda vidas, o cuando el player junta un item del escenario. Para implementar esta clase deberemos heredar de la clase de Box2D `b2ContactListener`, luego tendremos que implementar un par de funciones que definirán el comportamiento de nuestro listener. El prototipo mas básico de un listener es:

```
class MiContactListener : public b2ContactListener
{
public:

    void BeginContact(b2Contact* contact)
    { // handle begin event }

    void EndContact(b2Contact* contact)
    { // handle end event }
};
```

Los dos metodos mas importantes que deberemos implementar sera:

Begin Contact

Este método (evento) es llamado cuando dos fixtures comienzan a colisionar.

End Contact

Este método (evento) es llamado cuando los fixtures dejan de colisionar.

Cuando usamos un listener el problema más común es saber qué colisionó con qué, ya que el listener se llama por todas las colisiones de nuestro mundo. Para ello pueden usar el campo

UserData de los cuerpos. En el mismo pueden guardar datos y después leerlos de manera de determinar el tipo de cada objeto. A continuación vemos un ejemplo dónde se guarda un entero y después se lo lee en el listener para determinar que cuerpo chocó:

```
//Primero cuando creamos el cuerpo seteamos el user data, pueden ser clases o cualquier tipo de
//datos
```

```
int *t= new int();
*t=0;
body->SetUserData(t);
```

Y luego en el listener, por ejemplo en el método BeginContact:

```
void BeginContact(b2Contact* contact)
{ // evento que se llama cuando comienza el contacto

    //preguntamos si el tipo del cuerpo correspondiente al fixture A tiene
    //un UserData que nos interese
    int *t=(int*) contact->GetFixtureA()->GetBody()->GetUserData();
    if(*t==0){
        b2Vec2 normal= contact->GetManifold()->localNormal;
    }
    //hacemos lo mismo para el objeto del otro fixture
    t=(int*) contact->GetFixtureB()->GetBody()->GetUserData();
    if(*t==0){
        b2Vec2 normal= contact->GetManifold()->localNormal;
    }
}
```

Para poner a nuestro contact listener a procesar contactos deberemos setearlo a la clase b2World de la siguiente forma:

```
MiContactListener listen;

world->SetContactListener(&listen);
```