Contents

1 Box 2d apuntes

1.1 Conceptos

1.1.1 Shape

Figura 2D Tambien conocido como fixture?

1.1.2 Rigid body

Tambien conocido como body. Cuerpos rigidos?

1.1.3 Fixture

Une un shape con un rigid body y añade propiedades materiales como densidad, friccion y restitucion. Esta es la que pone a una shape en el sistema de colision; asi logrando que pueda colisionar con otros objetos.

1.1.4 Constraint

Coneccion fisica que remueve grados de libertad de cuerpos. **Un cuerpo 2d tiene 3 grados de libertad**.

1.1.5 Contact constraint

Constraint especialmente diseñada para prevenir la penetracion de rigid bodys y para simular **friccion y restitucion**. **Son creadas automaticamente por Box2d**.

1.1.6 Joint

Constraint usada para mantener a dos cuerpos juntos. Hay muchos tipos:

- revolute
- prismatic
- distance
- Y mas

Alguna de estas tienen limit y motor (ver mas abajo)

1.1.7 Joint limit

Limita el rango de movimiento de un joint.

1.1.8 Joint motor

Es el encargado de manejar el moviemiento del joint segun el grado de libertad de la joint.

1.1.9 World

Colleccion de bodys, shapes y constraints que interactuan juntos. Se pueden crear varios mundos, pero no suele ser necesario.

1.1.10 Solver

El world tiene un solver que es usado para avanzar el tiempo y calcular contancto y joint constraints. Es O(n) siendo n la cantidad de constraints.

1.1.11 Continuous collision

Nada un socotroco que evita las cosas se teletransporten.

1.2 Modulos

Box2d esta compuesto de 3 modulos

1.2.1 Common

Codigo para allocacion(?, matematica y configuraicones

1.2.2 Collision

Define shapes, "broad phase"y las funciones de collision

1.2.3 Dynamics

Define el el world, bodies, shapes y joints

1.3 Units

Box2d usa floats. Tiene que trabajar con tolerancias medidas en metros, kilogramos y segundos (GOATed) **Esta optimizado para mover figuras entre 0.1 y 10 metros**. En teoria, puede representar cosas hasta 50 metros. **Usa radianes** para los angulos. NO ANGULOS.

NO es recomendado usar pixeles como unidad de medida. **Los worlds deberian ser de 2km o menos**

1.4 Factories

Si vos queres crear un b2Body o b2Joint tenes que llamar a las funciones factory EN b2World. **NO** crear a los objetos de otra forma.

Funciones de creacion:

```
b2Body* b2World::CreateBody(const b2BodyDef* def);

b2Joint* b2World::CreateJoint(const b2JointDef* def);
```

Funciones de destruccion:

```
void b2World::DestroyBody(b2Body* body);

void b2World::DestroyJoint(b2Joint* joint);
```

Cuando creas un body o joint tenes que dar una definicion. Esta definicion tiene que tener **toda** la infomacion necesaria para crear un body o joint.

Como las shapes (fixtures) estan atadas a un cuerpo, son creadas y destruidas usando una factory.

```
b2Fixture* b2Body::CreateFixture(const b2FixtureDef* def);

void b2Body::DestroyFixture(b2Fixture* fixture);

  //Tambien una funcion mas corta
b2Fixture* b2Body::CreateFixture(const b2Shape* shape, float density);
```