

Box2DFlash 2.1a Update Notes

Fixtures

One of the biggest changes is how shapes work. Material properties of shapes, such as density, friction, etc, have been separated from geometric properties (the radius, vertices etc). The former are now fixtures, while the latter remains in shapes.

After creating a body, you should create fixture definition for material properties. You provide the shape to the fixture, and call `b2Body.CreateFixture`:

```
1 var fd:b2FixtureDef = new b2FixtureDef();
2 fd.density = 1.0;
3 fd.shape = new b2CircleShape(5.0);
4 var fixture:b2Fixture = myBody.CreateFixture(fd);
```

For convenience, `b2Body.CreateFixture2` allows you to skip creating a `b2FixtureDef`, providing just a shape and (optional) density.

```
1 var fixture:b2Fixture = myBody.CreateFixture2(new
  b2CircleShape(5.0), 1.0);
```

Note that `b2CircleShape` and `b2PolygonShape` are used directly - there is no `b2ShapeDef`.

Body types

In 2.0, there were two sorts of bodies - static and dynamic. The mass determined if a body was static or not. Now a new type has been added, kinematic, for bodies that should move but not be pushed by other bodies. To account for this, you must set the body's type correctly using `b2BodyDef.type` and `b2Body.SetType`:

```
1 var bd:b2BodyDef = new b2BodyDef();
2 bd.type = b2Body.b2_dynamicBody;
```

Events

2.0's collision event system was very hard to use, and has been replaced. Events are now not per-contact-point, but per-contact. They provide a manifold containing information including the contact points. `BeginContact` and `EndContact` are provided for easily determining when two objects begin and end touching, and are the only events raised by sensors. Please see the [reference section](#) for more details.

Initialization

2.1 has a completely rewritten broadphase. It is now not necessary to specify a size for your world, it'll always be large enough.

Game Loop

Though likely to change for the final release, it is now necessary to call `b2World.ClearForces()` immediately after `b2World.Step`. This is to facilitate users who wish have multiple `b2World.Steps` between one game logic step, for the purposes of stability. You may also need to call `b2World.DrawDebugData()` if you are using debug drawing.

Restrictions

Several restrictions have been lifted.

- There are no world boundaries - it's always large enough thanks to a new broadphase.
- Polygons have no limit on the number of vertices. They can be as thin as you like, too, though they must still be convex. Even with two points, they are valid, making good edge shapes.
- Some internal limits have been removed, which means no more baffling assertion failures.

Original Site Design by Andreas Viklund