UNITYの
物理エンジンの機能

高橋 誠史

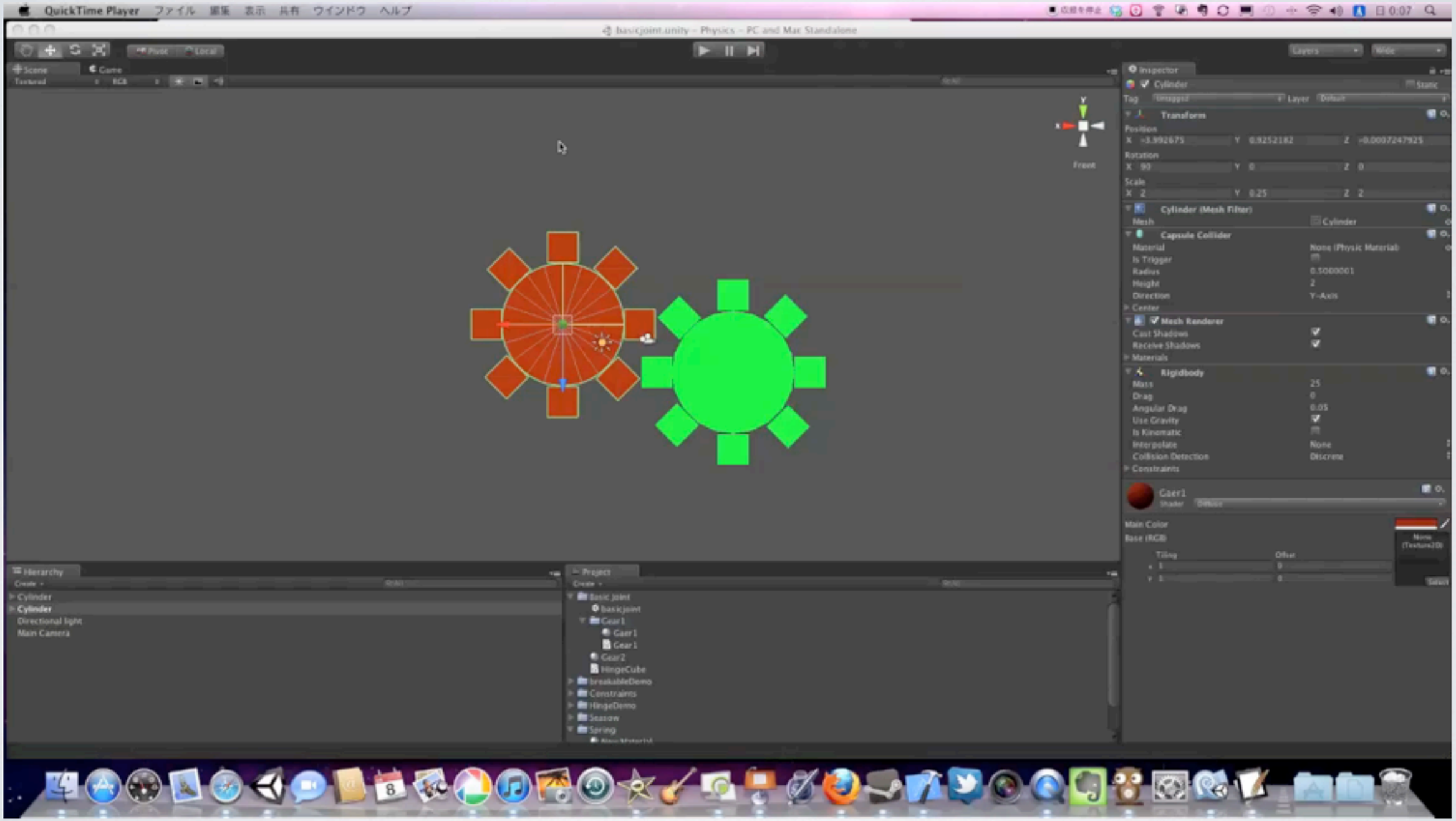
今回の話題

- Unityに乗ってるPhysX物理エンジンの機能を色々と知る
- 使い方を覚えればメカニカルな仕掛けをゲームにとりいれられる！！
- せっかくなので物理エンジンで使われる用語を覚えていこう
- 物理シミュレーションの計算とか難しいので今回は扱いません。
- なので、うまいことUnityでできることを知って、美味しく使いましょう

こんなあなたにオススメ

- ゲームデザイナー
  - これを覚えれば、ゲームやレベルデザインに簡単にメカニカルな物理ベースの仕掛けが取り入れられる
  - 自分で物理エンジンの機能を体験することでゲームデザインの幅を広げられる
- アーティスト
  - Unityの操作感はDCCに近いのでUnityで覚えた物理エンジン機能はMayaでも生かせる→Maya DynamicsやPhysXプラグインやAPEXなどに生かす
  - カットシーンのアニメーションなどに物理エンジンを取り入れられれば手で制御しなくていい
  - エフェクトや演出の道具として物理エンジンを取り入れる練習になる
- プログラマ
  - 物理エンジンの機能を手軽に試せるので実験に良い
  - 物理エンジンを自分のゲームに組み込む際のヒントになる
  - ツールから慣れて物理エンジンSDKのコードに入る

こんなことできます


QuickTime Player ファイル 編集 表示 共有 ウインドウ ヘルプ
basicjoint.unity – Physics – PC and Mac Standalone
Hierarchy
Cylinder
Cylinder
Directional light
Main Camera
Project
Basic Joint
basicjoint
Gear1
Gaer1
Gear1
Gear2
HingeCube
breakableDemo
Constraints
HingeDemo
Seesaw
Spring
Inspector
Cylinder
Transform
Position
Rotation
Scale
Cylinder (Mesh Filter)
Capsule Collider
Mesh Renderer
Rigidbody
Gaer1
Main Color
Base (RGB)

UNITYの物理エンジン

- 物理エンジン自体はNVIDIA PhysXが入ってる
  - 余談…Unreal EngineもPhysXなので用語などに共通なので覚えておいて損はないかも
- Unityでどんな機能が使えるか？どうやってやるか？

今回の話題

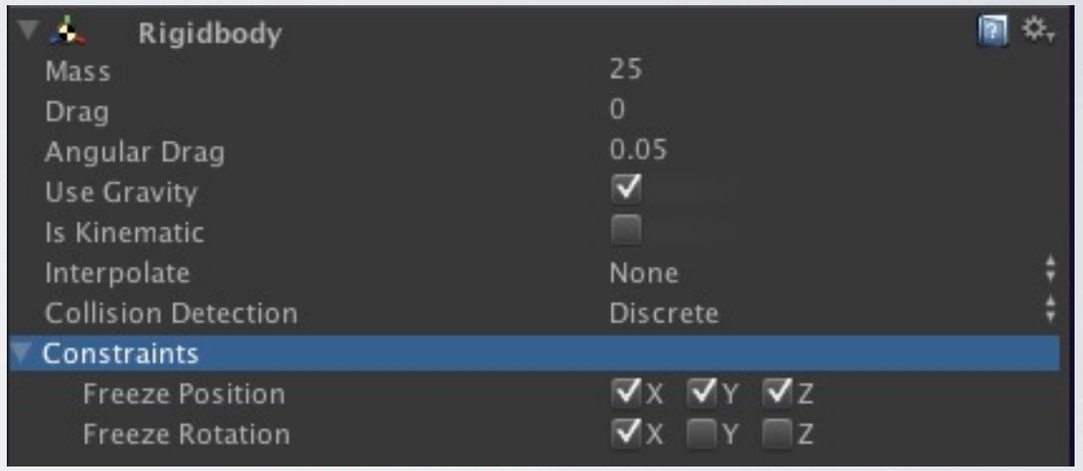
- 剛体(Rigid body)
- コンストレイン(Constraints)
- ジョイント(Joint)
  - Breakable joint(破壊可能なジョイント)
- Constant force
- スクリプトによる制御

剛体(RIGIDBODY)

- ようは、変形しない物体

- 物理シミュレーションの対象となる物体

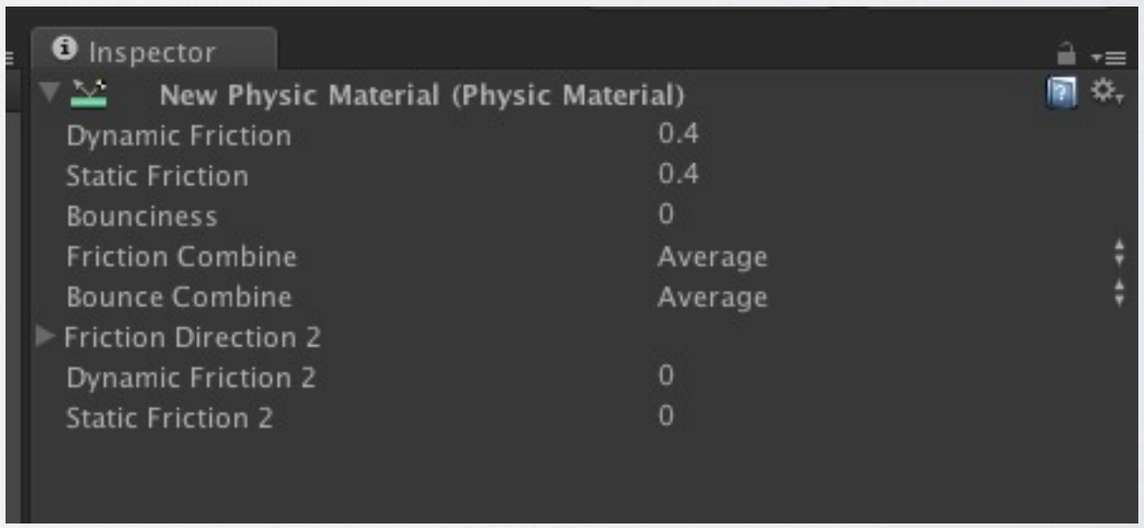
RIGIDBODYのパラメータ



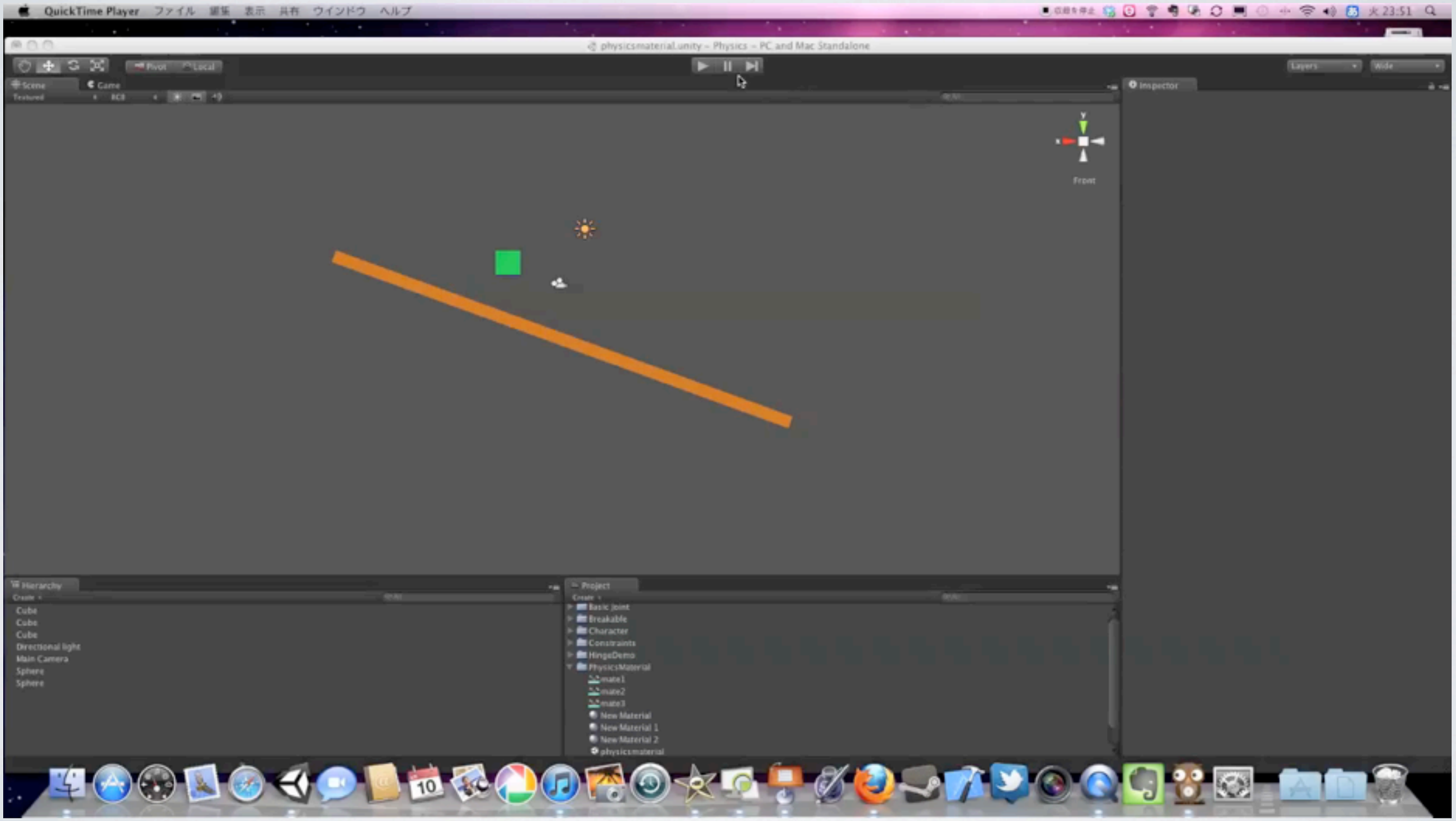
- Mass…質量
- Drag…リニアダンピング係数
- Angular Drag…角ダンピング係数
- Use Gravity…重力の影響の有無
- Is Kinematic…これがtrueだと物理シミュレーションがOFFになる
- Interpolate…シミュレーションのフレーム補間の有無(固定フレーム時)
- Collision Detection…シミュレーションのフレームの固定の有無など
- Constraints…拘束（後述）

物理マテリアル

- 摩擦とか反発係数とか物理的な材質的なパラメータ
  - Friction…摩擦(Dymaicが動摩擦、Staticは静摩擦)
  - Bounciness…反発



マテリアルの違い


QuickTime Player ファイル 編集 表示 共有 ウインドウ ヘルプ
physicsmaterial.unity – Physics – PC and Mac Standalone
Pivot
Local
Scene
Game
Inspector
Front
Hierarchy
Cube
Cube
Cube
Directional light
Main Camera
Sphere
Sphere
Project
Basic Joint
Breakable
Character
Constraints
HingeDemo
PhysicsMaterial
mate1
mate2
mate3
New Material
New Material 1
New Material 2
physicsmaterial

CONSTRAINTS

- コンストレイン（拘束）

- 物理エンジンによってはPhysXのJointと同義(Bulletなんかそう)の場合があるので注意

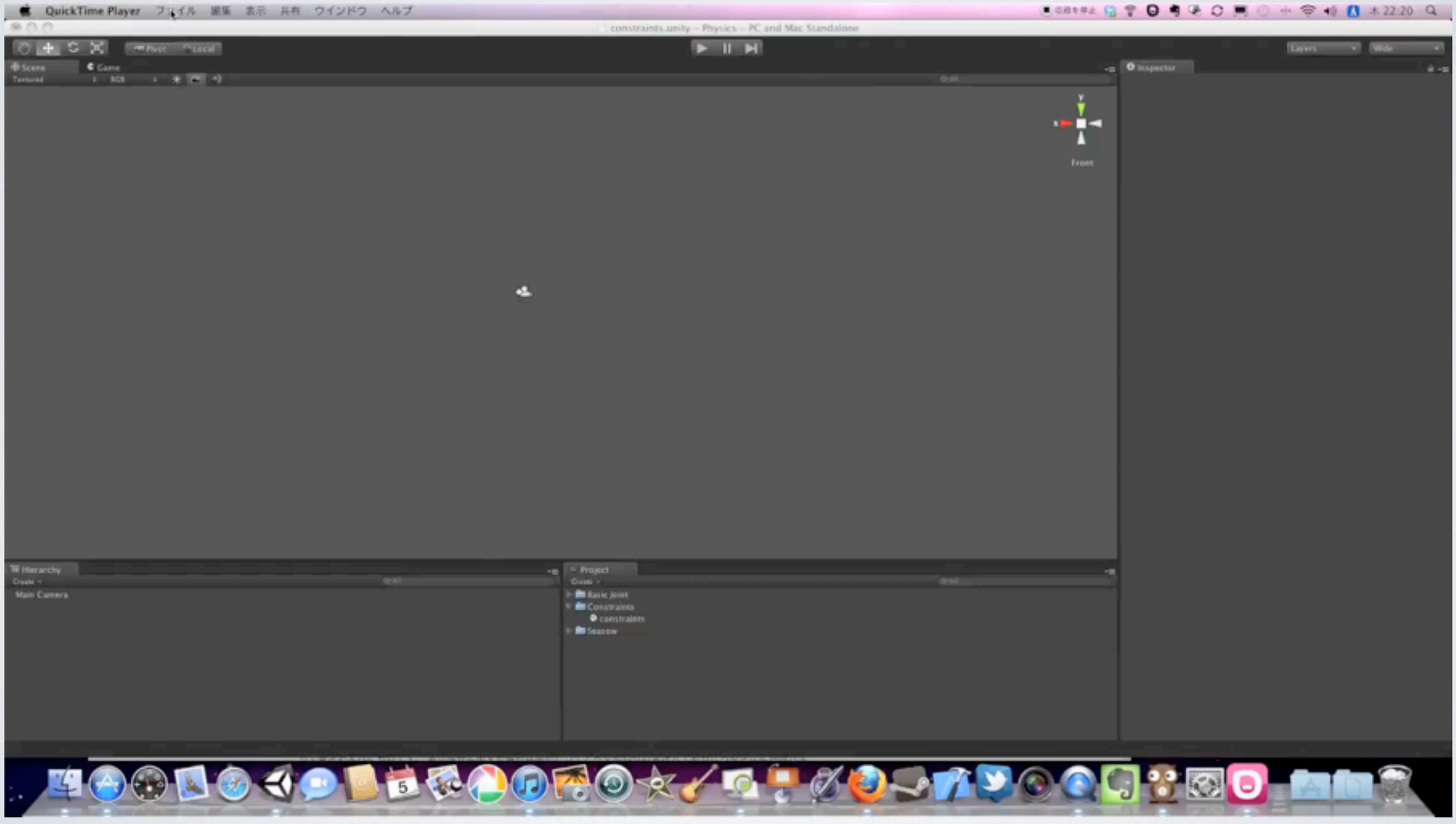
UNITYのCONSTRAINTS

- UnityのConstraintは物理エンジンのシミュレーションの平行移動と回転の処理を軸単位で停止できる（正確にはPhysXのContraintsの仕様）

- Regidbodyのところで設定できる
  - Freeze Position…XYZの任意軸の平行移動を停止
  - Freeze Rotation…XYZの任意軸の回転を停止

- たとえば、Freeze PosionでZを指定すると奥行き方向のシミュレーションが働かなくなるので3Dモデルを使った2Dゲームみたいなのを作ったりするのに便利

CONSTRAINTSデモ



JOINT

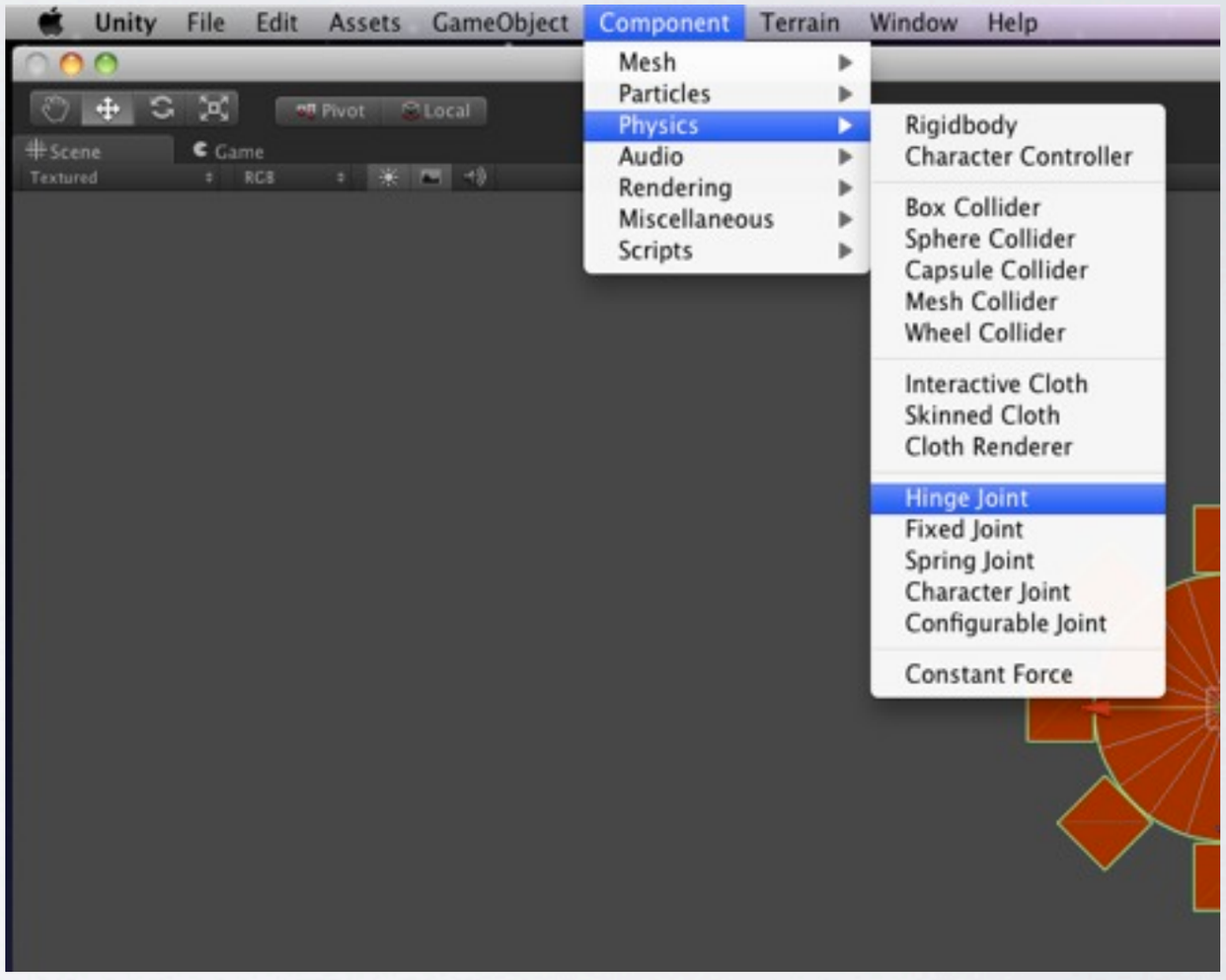
- 剛体同士をつなげます
- 物理エンジンで様々な仕掛けを作りたい場合には必須機能

JOINTの使い方

- メニューのCompornentの下に「～ Joint」という感じで5種類

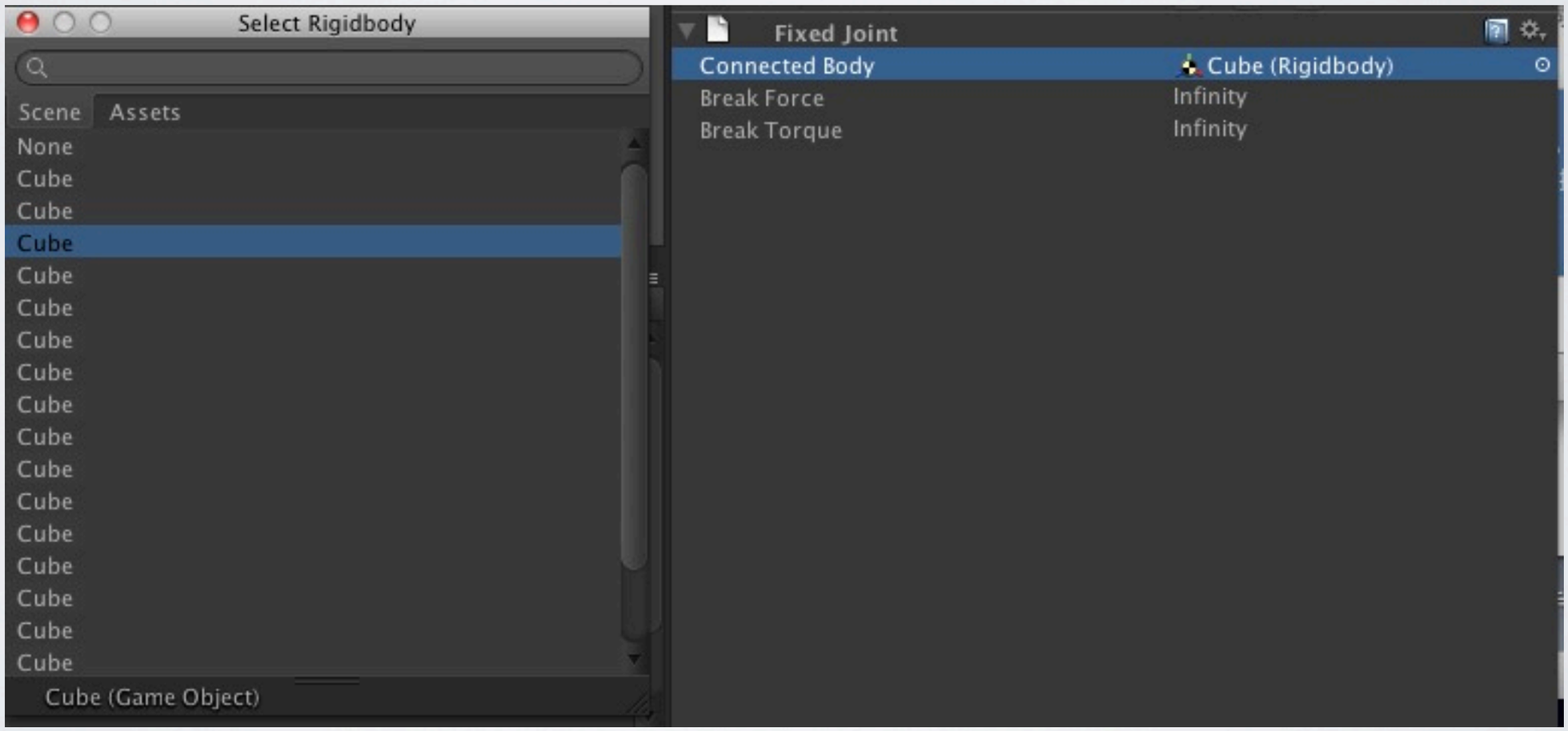
- Rigidbodyのように選択したオブジェクトがjoint化

- jointを指定するとそのオブジェクトはRigidbody化



JOINTの使い方

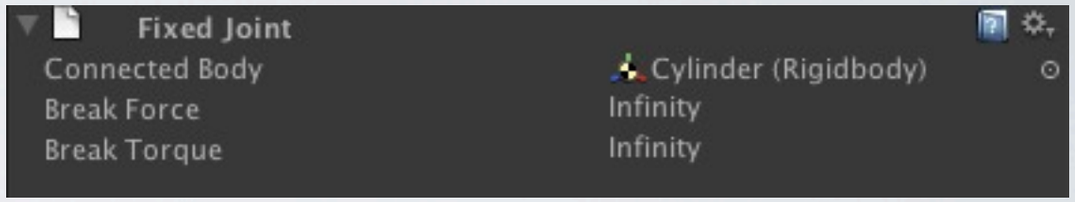
- Inspectorのjointの箇所でConnected Bodyで他のRigidbodyを選択すれば接続
- Conneted Bodyがない場合は、その場に空中で固定かAnchorの場所で固定になる
  - Constraintsと似た感じと言えなくはない？



UNITYで使えるJOINT

- Fixed(固定)
- Hinge(ヒンジ)
- Spring(スプリング)
- Character(キャラクター)
- Configuratable(コンフィギュラブル)

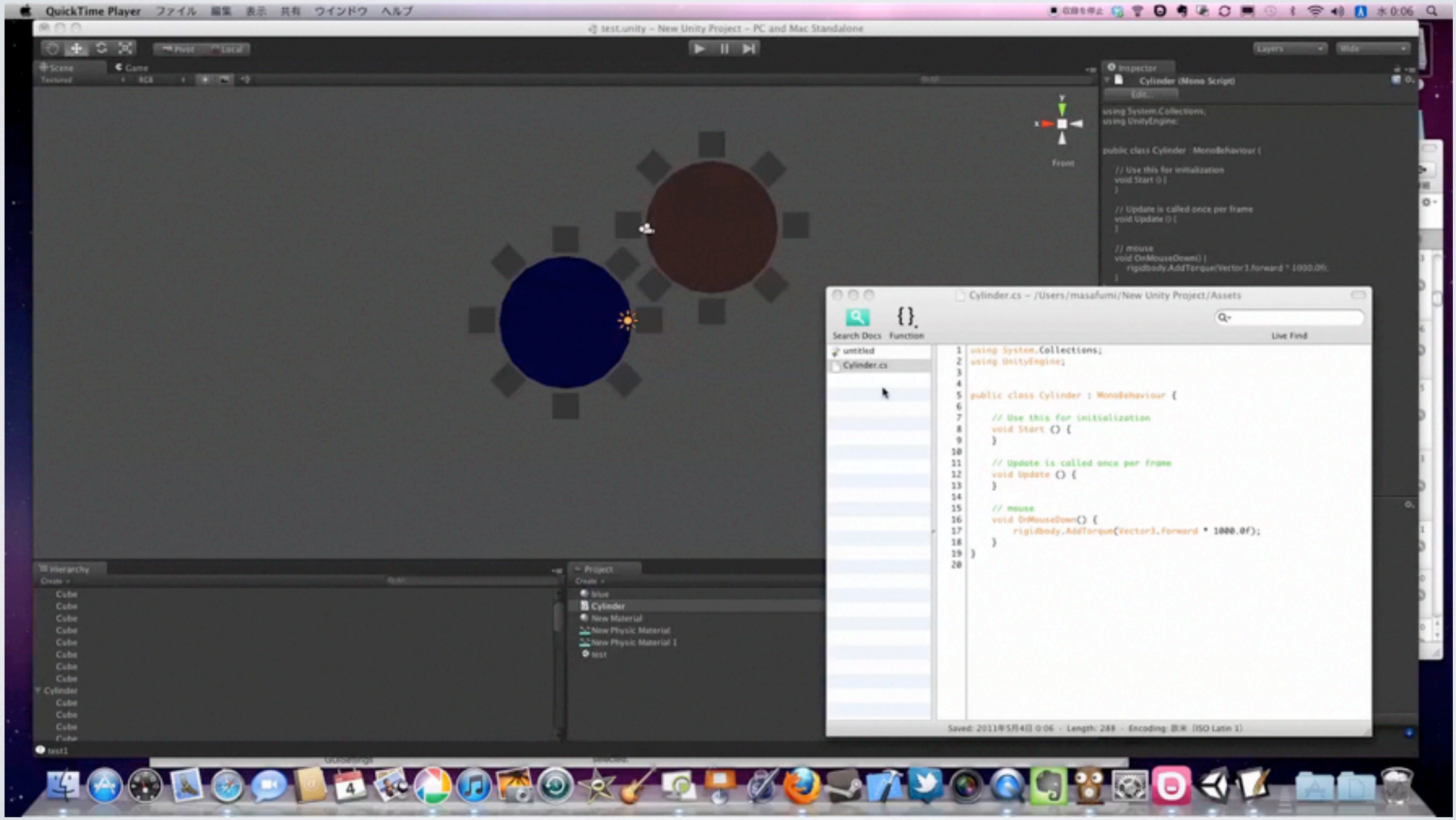
FIXED



- 固定ジョイント
- ジョイント部分が動くことは無いけどあんまり制御もできないので実は動いちゃったりする…

- 項目
  - Connected Body…ジョイントで接続する相手
  - Break Force…指定した力が加わるとjointが壊れる(Infinityなら壊れない)
  - Break Torque…一定のトルクが加わると壊れる

FIXEDデモ


QuickTime Player ファイル 編集 表示 共有 ウインドウ ヘルプ
test.unity – New Unity Project – PC and Mac Standalone
Inspector
Cylinder (Mono Script)
using System.Collections;
using UnityEngine;
public class Cylinder : MonoBehaviour {
// Use this for initialization
void Start () {
}
// Update is called once per frame
void Update () {
}
// mouse
void OnMouseDown() {
rigidbody.AddTorque(Vector3.forward * 1000.0f);
}
Cylinder.cs – /Users/masafumi/New Unity Project/Assets
Search Docs
Function
Live Find
untitled
Cylinder.cs
1 using System.Collections;
2 using UnityEngine;
3
4
5 public class Cylinder : MonoBehaviour {
6
7 // Use this for initialization
8 void Start () {
9 }
10
11 // Update is called once per frame
12 void Update () {
13 }
14
15 // mouse
16 void OnMouseDown() {
17 rigidbody.AddTorque(Vector3.forward * 1000.0f);
18 }
19 }
20
Saved: 2011年5月4日 0:06 · Length: 288 · Encoding: 日本 (ISO Latin 1)
Hierarchy
Cube
Cylinder
Project
blue
Cylinder
New Material
New Physic Material
New Physic Material 1
test
test1

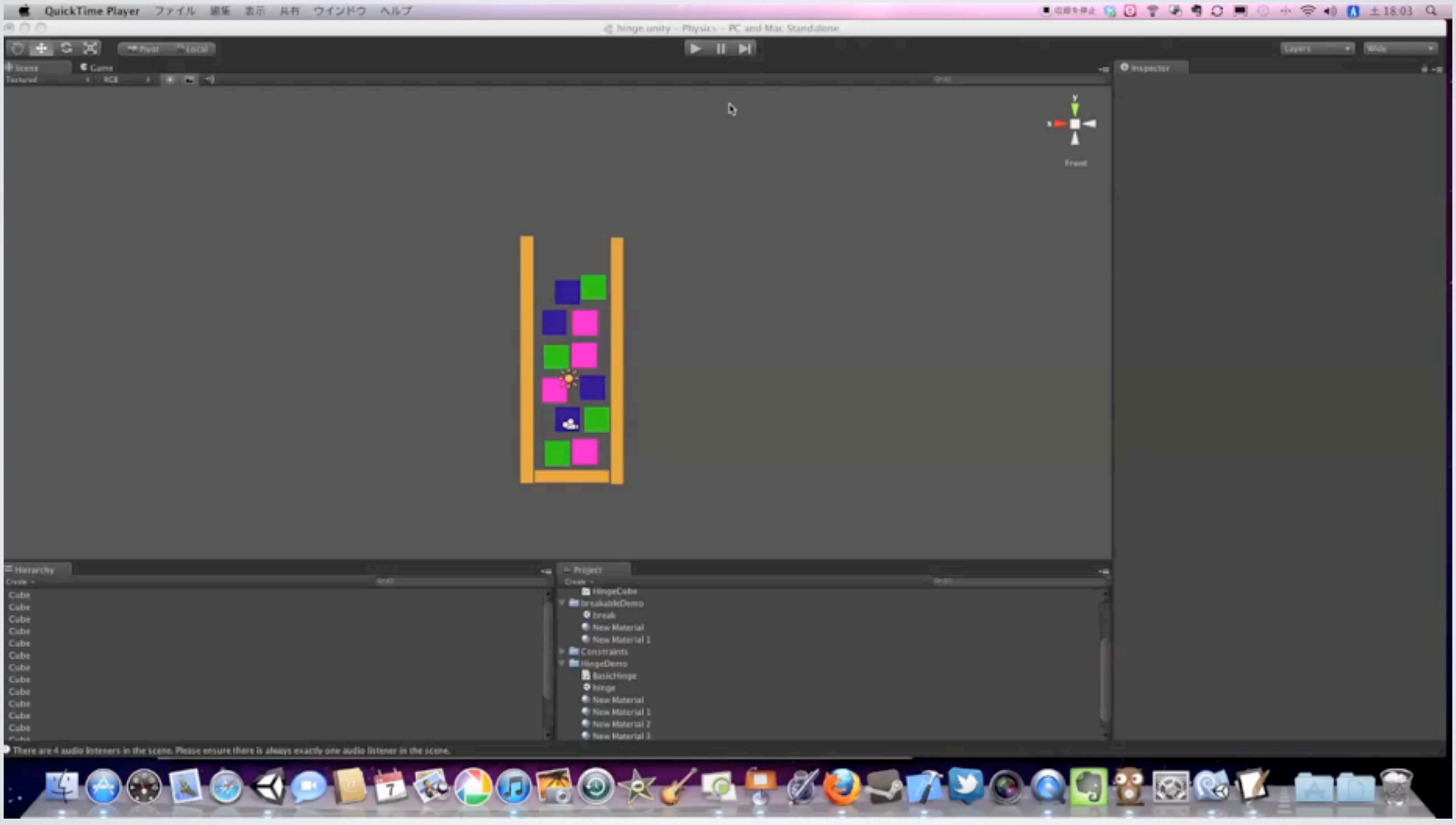
HINGE

- ヒンジ
- 蝶つがい的な動作(ドアなど)
- 回転軸をひとつ持つ（複数ならConfigurableを使う）
- モーター

HINGEの調整項目

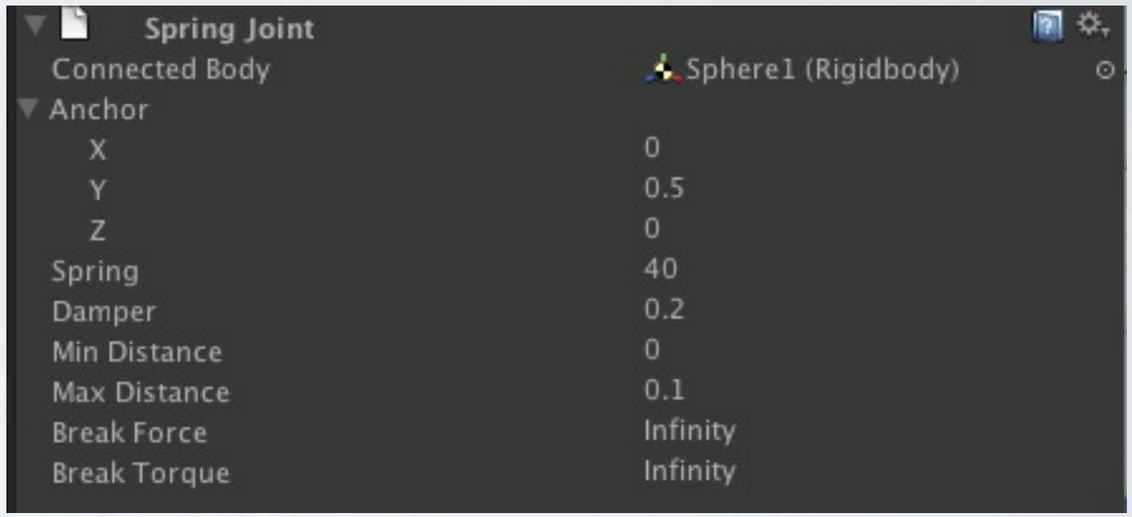
- Anchor(アンカー)
  - Jointの基点位置。オブジェクト相対で指定
- Axis
  - ヒンジの回転軸
- Spring
  - jointにバネ的な動きをさせる場合
- Motor
  - 軸自体が回転する。

HINGEデモ

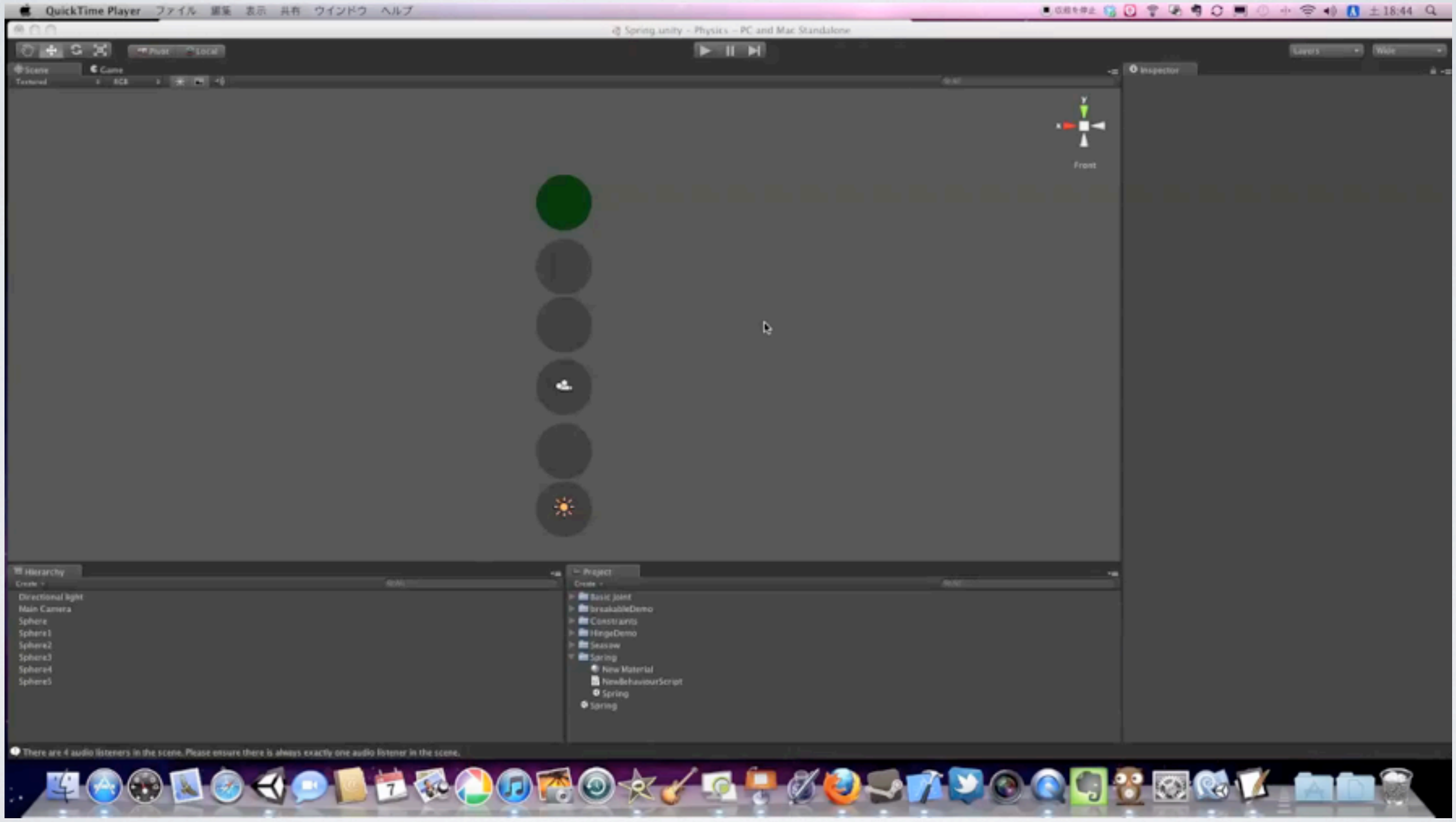

QuickTime Player ファイル 編集 表示 共有 ウインドウ ヘルプ
hinge.unity – Physics – PC and Mac Standalone
Hierarchy
Project
Inspector
There are 4 audio listeners in the scene. Please ensure there is always exactly one audio listener in the scene.

SPRING

- バネ
  - Anchor…Hingeの時と同じ
  - Spring…バネの係数
  - Damper…ダンパー係数
  - Min Distance…Joint間の最小距離
  - Max Distance…Joint間の最大距離



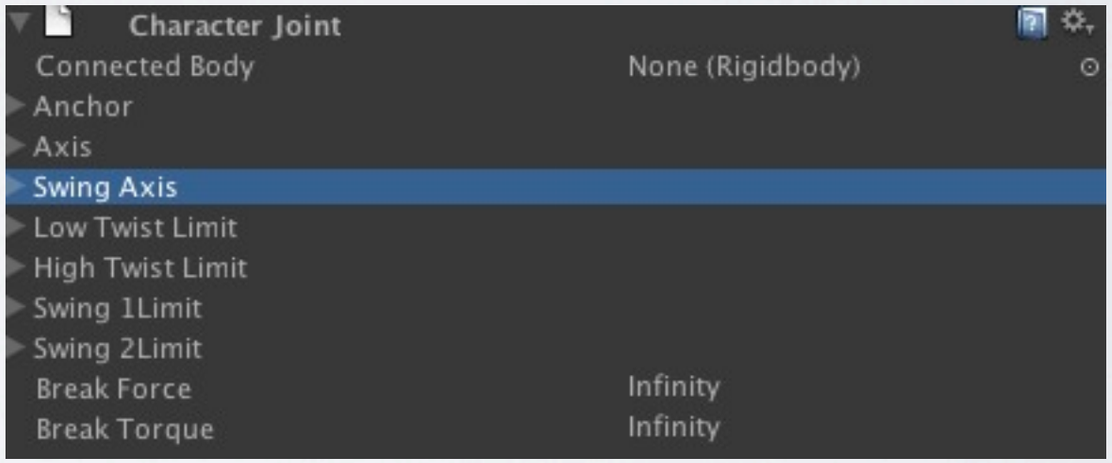
SPRINGデモ


QuickTime Player ファイル 編集 表示 共有 ウインドウ ヘルプ
Spring.unity - Physics - PC and Mac Standalone
Front
Hierarchy
Directional light
Main Camera
Sphere
Sphere1
Sphere2
Sphere3
Sphere4
Sphere5
Project
Basic Joint
BreakableDemo
Constraints
HingeDemo
Seasaw
Spring
New Material
NewBehaviourScript
Spring
Spring
There are 4 audio listeners in the scene. Please ensure there is always exactly one audio listener in the scene.

CHARACTER

- キャラクターコントローラー指定したものを対象にしたJoint

- 今回は割愛



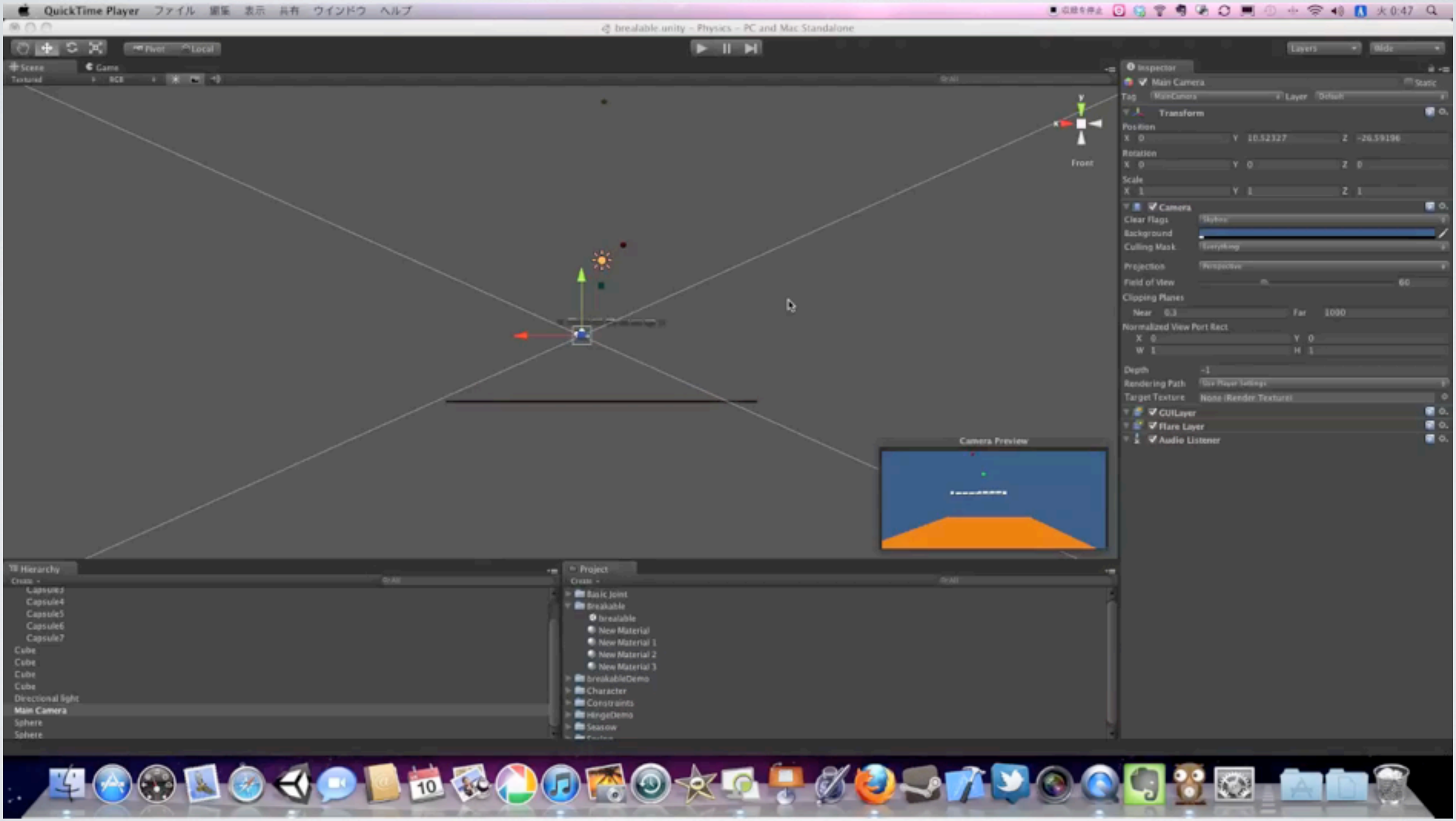
CONFIGURATABLE

- とにかく設定項目が多くてカスタマイズができる
- 軸ごとの拘束とか様々なことができる
- 今回は詳細割愛

BREAKABLE JOINT

- 破壊可能なjoint
- 物理的な力が加わるとjointの接合が壊れる
  - ForceとTorqueそれぞれ設定できる
  - 破壊のような表現に使える

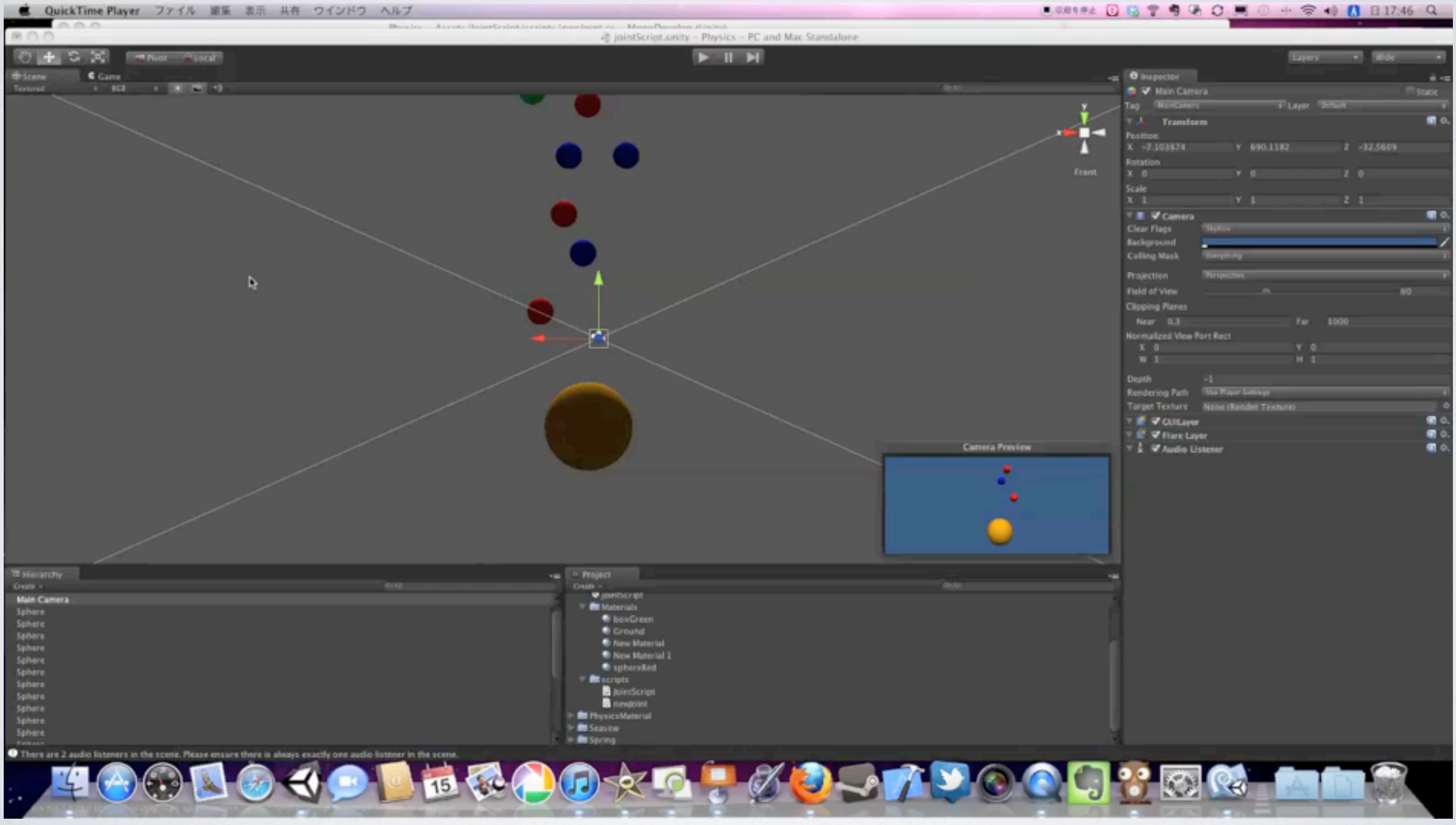
BREAKABLE JOINTデモ



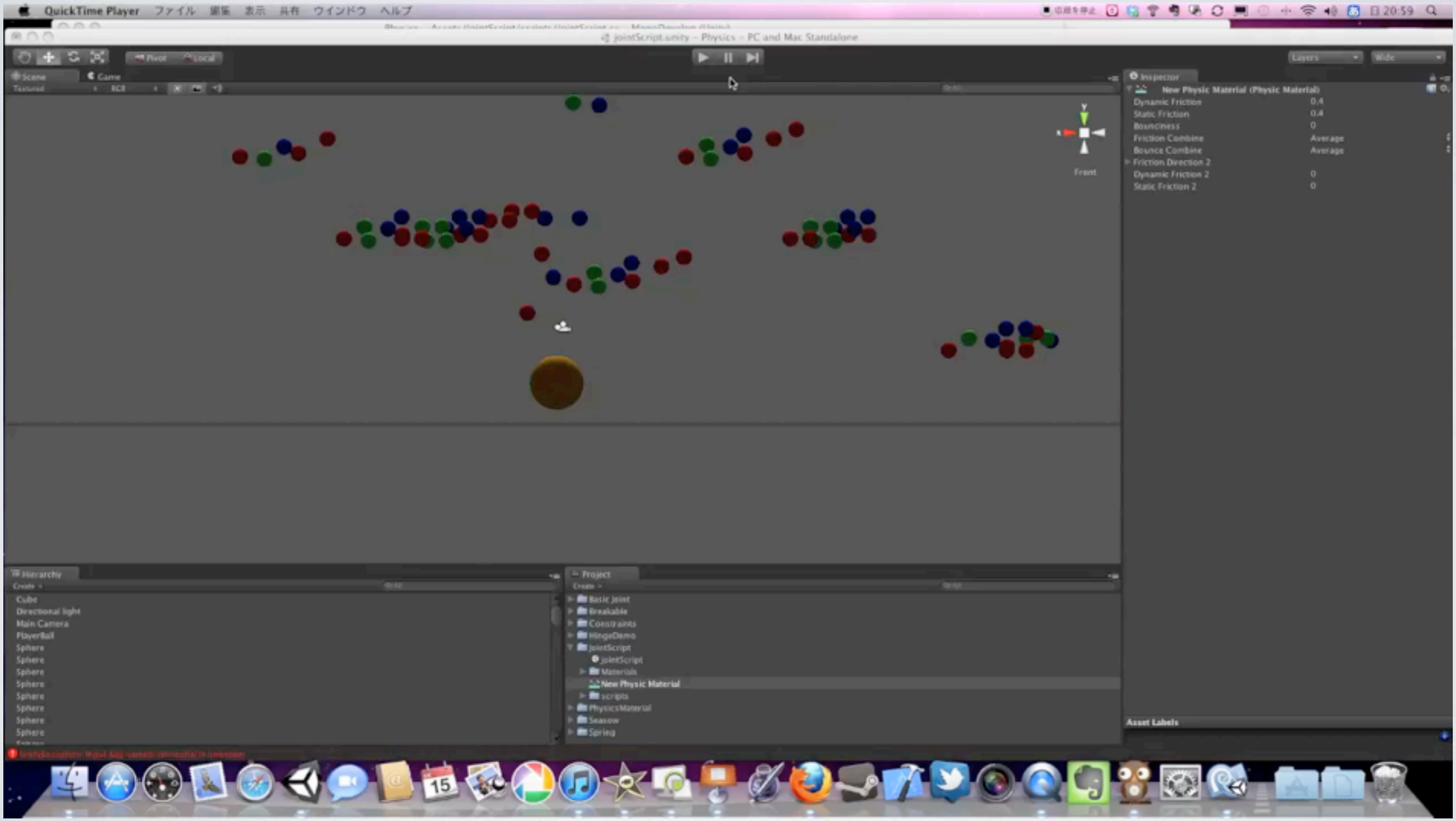
スクリプトと物理エンジン

- 剛体は、Rigidbodyクラス
- Jointは、HingeJointクラス(なぜかHinge以外使えなかったので要調査...)
  - スクリプトリファレンスだとあるけど、エラーになる
- スクリプトならゲームの処理の中で剛体やJoint生成ができる

スクリプトでJOINTを生成してみたデモ



おまけ


QuickTime Player ファイル 編集 表示 共有 ウインドウ ヘルプ
jointScript.unity – Physics – PC and Mac Standalone
Pivot
Local
Layers
Wide
Scene
Game
Inspector
New Physic Material (Physic Material)
Dynamic Friction 0.4
Static Friction 0.4
Bounciness 0
Friction Combine Average
Bounce Combine Average
Friction Direction 2
Dynamic Friction 2 0
Static Friction 2 0
Front
Hierarchy
Cube
Directional light
Main Camera
PlayerBall
Sphere
Project
Basic Joint
Breakable
Constraints
HingeDemo
jointScript
jointScript
Materials
New Physic Material
scripts
PhysicsMaterial
Seasaw
Spring
Asset Labels

剛体に新規にJOINTを追加してみるには？

- Jointの無い剛体にJointを追加する
- JointはGame ObjectのComponentなので以下のコードで追加できる
- Hingeの例

```
gameObject.AddComponent("HingeJoint");
```

スクリプトでJOINT対象を追加したり、変更したりする

- Jointにあらたなターゲットを指定するには？
- Joint系クラスのメンバのconnectedBodyに相手の

RigidBodyをセット

```
rigidbody.hingeJoint.connectedBody = other.rigidbody;
```

衝突があったらJOINT生成と接続例

```
void OnCollisionEnter(Collision other)
{
	if(!rigidbody.hingeJoint)
	{
		gameObject.AddComponent("HingeJoint");

		rigidbody.hingeJoint.connectedBody = other.rigidbody;
		rigidbody.hingeJoint.axis = Vector3.zero;
		rigidbody.hingeJoint.anchor = Vector3.zero;
	}
}
```

まとめ

- jointを使うことでRigidbodyをまとめることができる
- jointを上手に組み合わせるとメカニカルな仕掛けが簡単に作れる

検証が必要と思われること

- Unityの物理エンジンのパフォーマンス
  - Unity側からのオーバーヘッドは？
    - 今回は、BoxやSphereの衝突形状ばかりだったがMeshを多用したらどうなるか？
- より複雑なゲーム用のアセット向けの複合的な衝突形状はどう扱う？

質問&議論

- 答えられれば…