

基礎輪講 第2週

Unity講習

杉本研究室4年
谷 直人



内容

前半はUnityの紹介, 後半は今日やってもらうことの説明

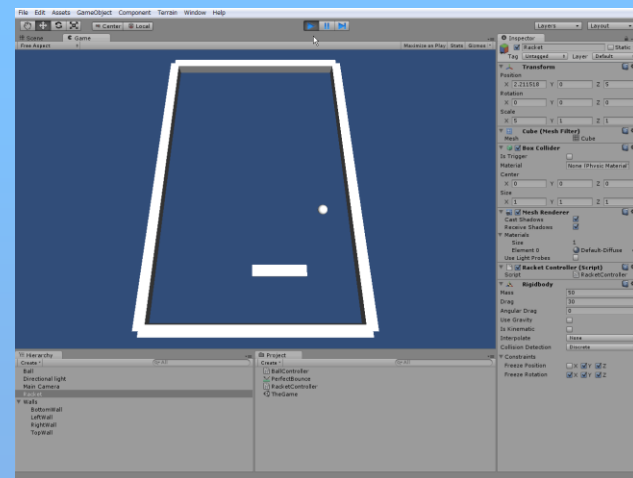
Unityの概要

- Unityとは？
- Unityを用いた作品例
- Unityでできること
- Unityで使用する言語



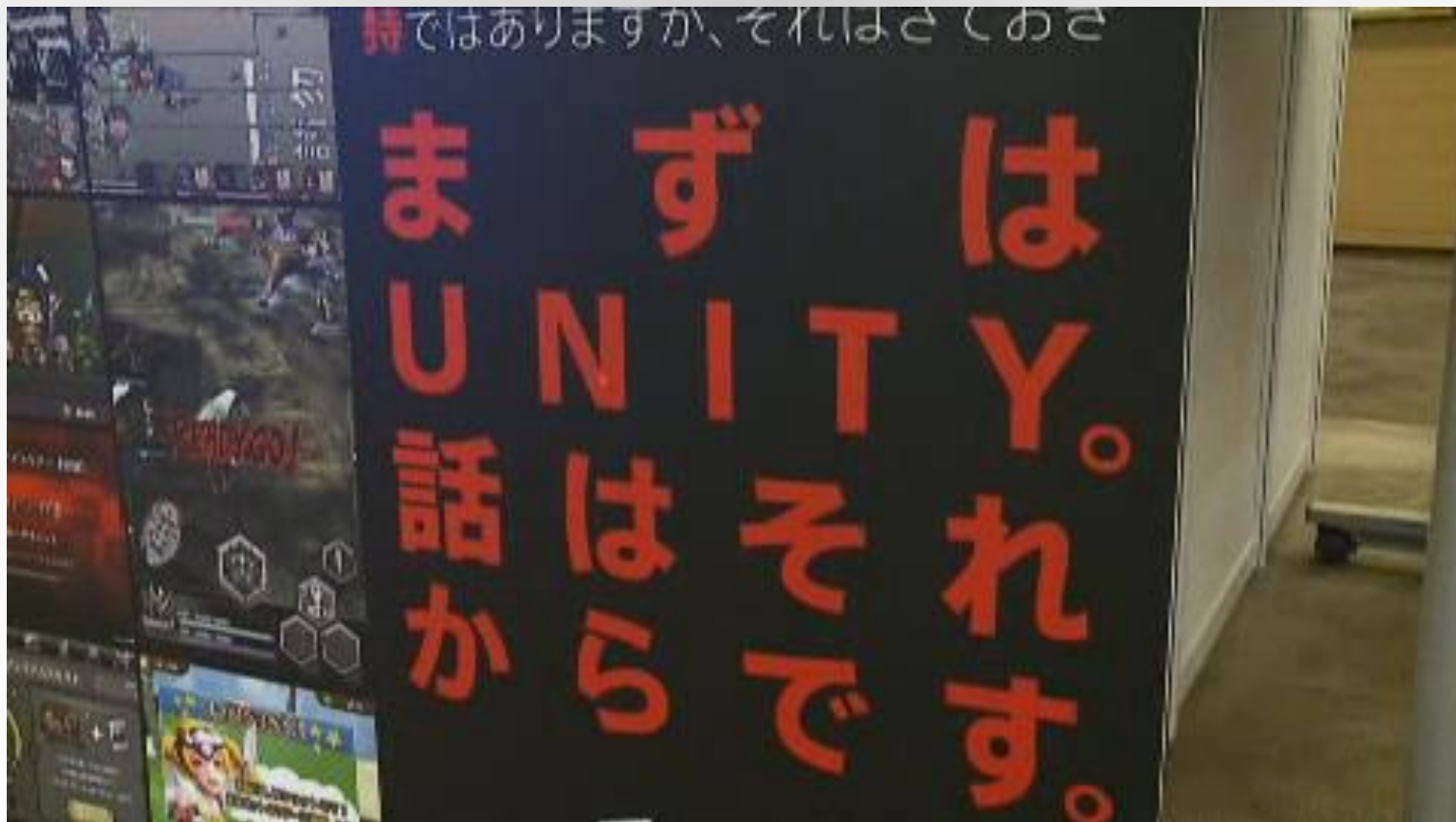
Unityを使ってみよう

- Unityのダウンロード, インストール
- はじめてのUnity
- スクリプトの記述, 物理エンジンの設定
- ボールの跳ね返しゲームの作成
- UnityとOpenCVの融合
- 顔検出でボール跳ね返しゲーム



Unityとは？

3Dアプリケーションを制作できるゲームエンジン



Unityを紹介する番組^[1]

[1] 賢者の選択 第24回 2013年9月15日放映

<http://kenja.jp/company/uov.php?tarC=businesslab1&tar=122>

Unityを使用した作品例



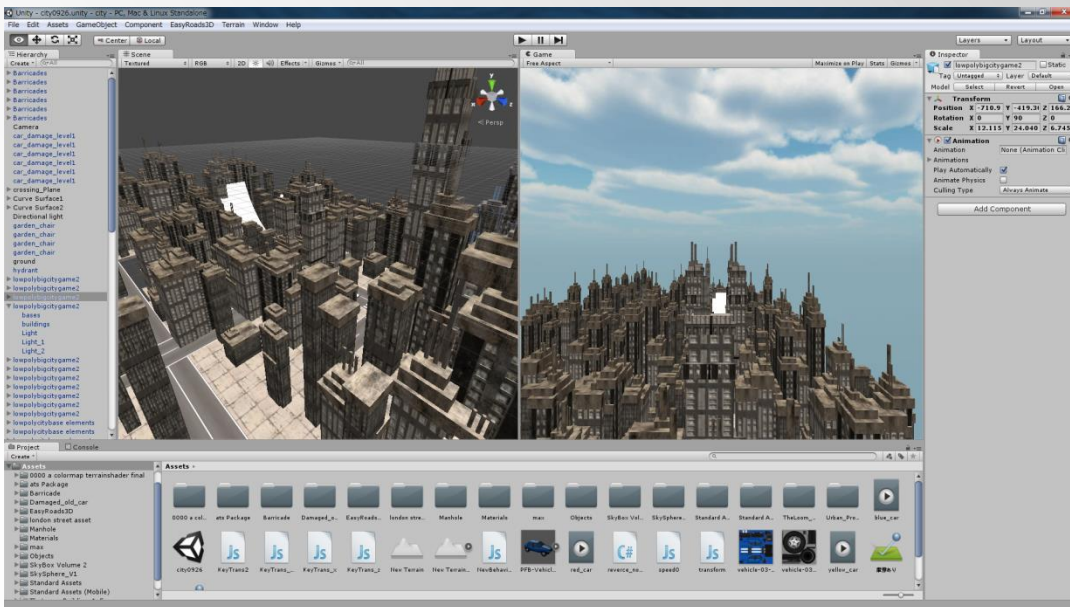
Unityを使用している作品の例[2]

[2] Unity ホームリスト <http://japan.unity3d.com/gallery/made-with-unity/game-list>

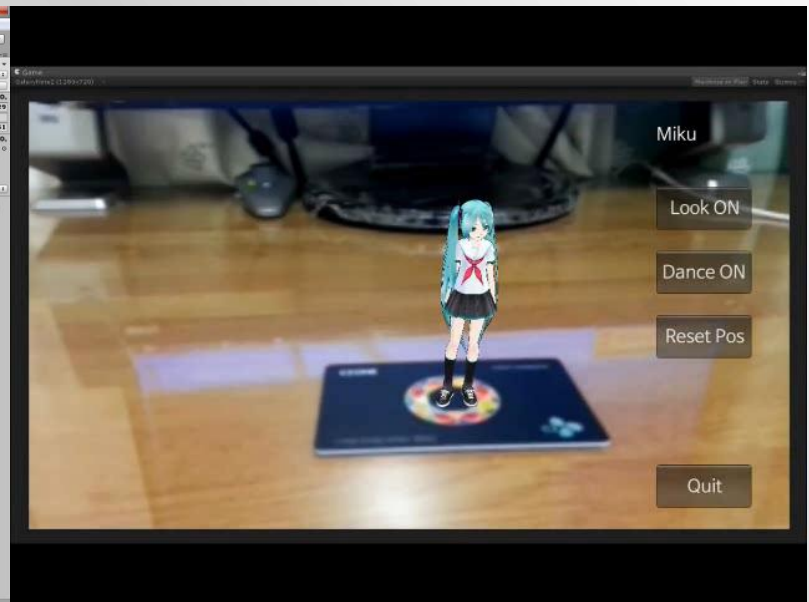
慶應義塾大学理工学部情報工学科杉本研究室 バーチャルロープスライダー <http://iu-lab.net/virtualropeslider/>

Unityでできること

- 物理エンジンを考慮したシミュレーション
- Assetと呼ばれるCGモデルの利用, 共有
- iOSやAndroid向けのアプリ開発
- Vuforiaというライブラリを用いてARアプリケーションの開発



Unityの開発画面



Unity+VuforiaでAR[3]

ゲームだけでなくインタラクティブなコンテンツを作成することが可能

Unityで使用する言語

Unityでは、物体の挙動制御やGUI設定のためのコードを適用できる。

```
gui.cs
No selection
1 using UnityEngine;
2 using System.Collections;
3
4 public class gui : MonoBehaviour {
5
6     // Use this for initialization
7     void Start () {
8
9     }
10
11     // Update is called once per frame
12     void Update () {
13
14     }
15
16     void onGUI () {
17         if ( GUI.Button(new Rect(0, 0, 100, 20), "ぼたん" ) )
18         {
19             Debug.Log("ぼたんがおされました");
20         }
21     }
22 }
```

C#で書かれたコード

```
Test.js
Test ▶ OnGUI
1 #pragma strict
2
3 function Start () {
4
5 }
6
7 function Update () {
8
9 }
10
11 function OnGUI () {
12
13     if ( GUI.Button (Rect (0, 0, 100, 20), "ぼたん"))
14     {
15         Debug.Log("ぼたんがおされました");
16     }
17 }
```

JavaScriptで書かれたコード

どちらで書いても基本的に変わらないので、使いやすい方を使おう。

本日の輪講でやってもらいたいこと

Unityのダウンロード, インストール

11:00

「はじめてのUnity」に沿って作業

11:30

スクリプトの記述法や物理演算に慣れる

12:00

ボールの跳ね返しゲームが完成

12:30

UnityとOpenCVの融合

14:30

顔検出でラットを操ってみる

16:30

お昼ご飯

本日の輪講の流れ

Unityのダウンロード, インストール

では実際にUnityを触ってみましょう！

「はじめてのUnity」というページにあるダウンロードをクリック



アカウント確認画面が出たら, 右のCreate Accountでアカウント作成

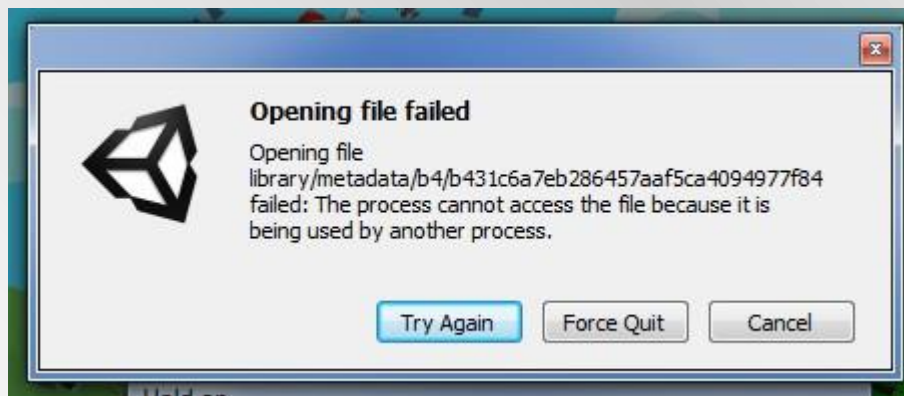
アカウント確認画面

ライセンス確認画面は真ん中をチェック

「初めてのUnity」に沿って作業

基本的にページ通りに進めていけば問題なし

たまに“～ failed”といって何かに失敗することがある。
→ Try Again を押し続けるとなんとかなることが多い。



Try Again を押しまくれ！

マウス操作に慣れるとScene画面操作の効率アップ

- 右クリックしながら動かすとカメラのRotation
- ホイールを押しながら動かすとカメラのTransform

スクリプトの記述や物理演算に慣れる

ページにはJavascriptで作れと書いてあるけど, C#でもOK.

C#の場合, ファイル名とクラス名は一致させる必要がある.



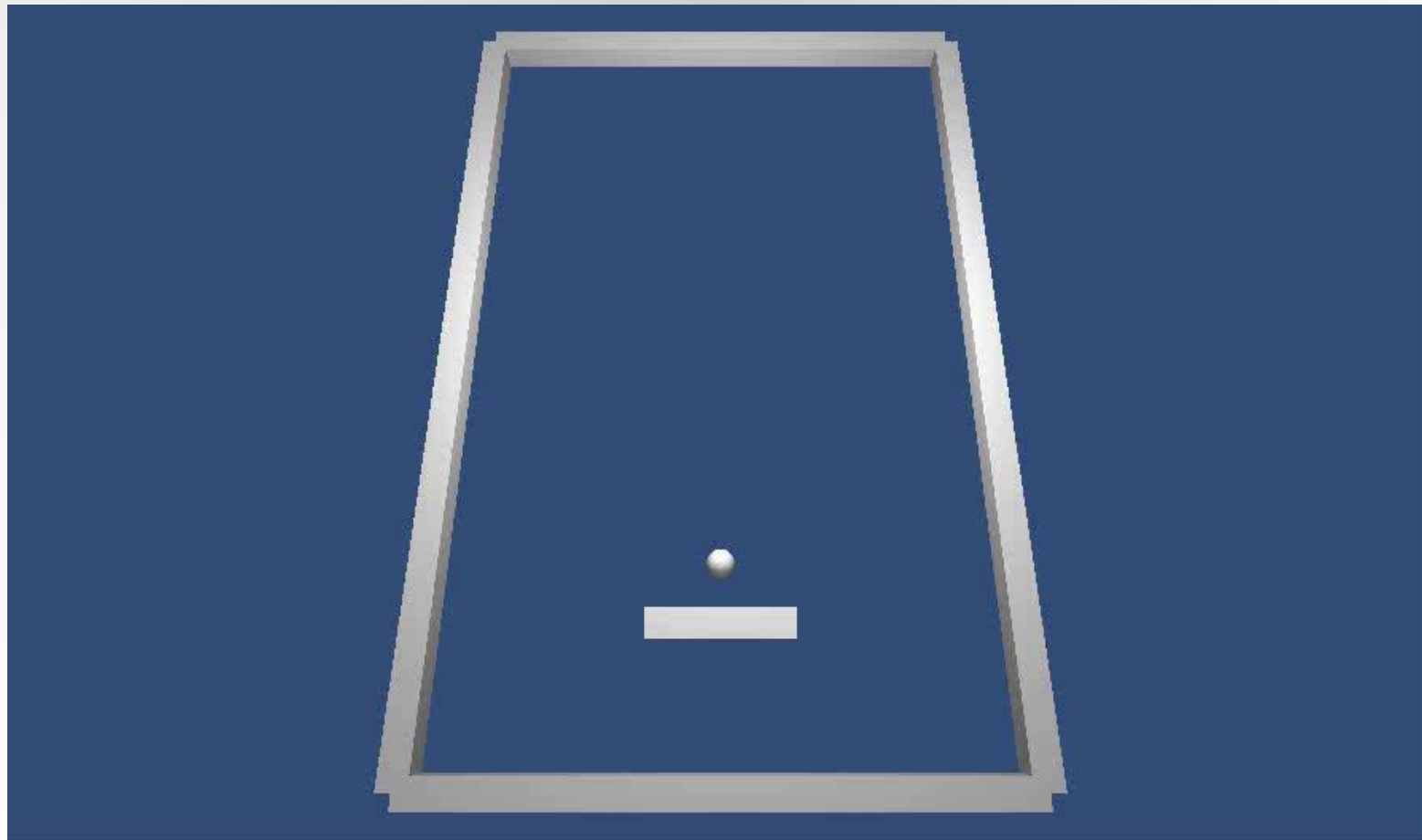
```
1 using UnityEngine;
2 using System.Collections;
3
4 public class BallController : MonoBehaviour {
5     double Speed = 20.0;
6
7     // Use this for initialization
8     void Start () {
9         rigidbody.AddForce(
10             ( transform.forward + transform.right ) * Speed,
11             ForceMode.VelocityChange );
12     }
13
14     // Update is called once per frame
15     void Update () {
16
17     }
18 }
19
```

C#ではクラス名とファイル名を一致させる

(※ RacketControllerだけはC#で書いておいた方が後で楽できるかも…)

ボールの跳ね返しゲームが完成

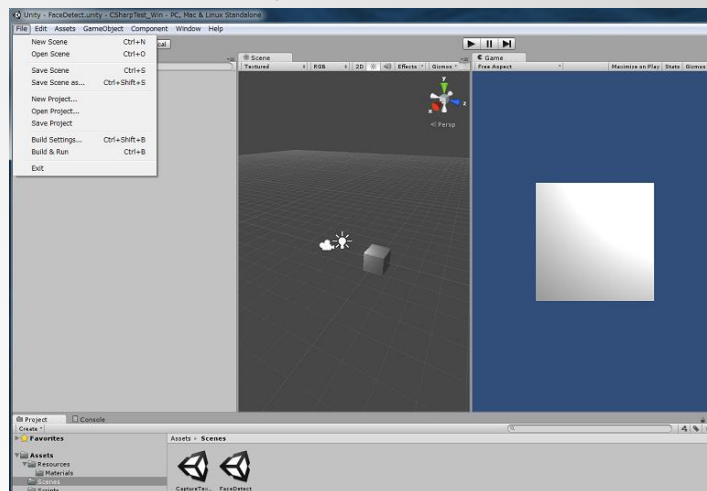
下のデモ動画のようなゲームが完成すればOK！



ボールの跳ね返しゲーム

UnityとOpenCVの融合 (確認)

1. まずは基礎輪講2014のページからサンプルコードをダウンロード
2. UnityのFile -> Open Project -> Open Other でDLしたフォルダを選択
3. Webカメラがつながっていることを確認
4. Scenesの“FaceDetect”をダブルクリック
5. Unity画面中央上の再生ボタンをクリック
6. 人間の顔が検出されて、円が描かれているウィンドウが起動
7. 検出された顔に追従してUnity内のボックスが動けば成功！



UnityとOpenCVの融合 (スクリプト)

顔検出をするコードと、カメラを動かすコードの2つ

```

36
37 // Use this for initialization
38 void Start()
39 {
40     // カスケードファイル読み込み
41     cascade = CvHaarClassifierCascade.FromFile(@"Assets/haarcascade_frontalface_alt.xml");
42     // Webカメラを使うという宣言
43     capture = Cv.CreateCameraCapture(0);
44     // 縦横とフレームの設定
45     Cv.SetCaptureProperty(capture, CaptureProperty.FrameWidth, CAPTURE_WIDTH);
46     Cv.SetCaptureProperty(capture, CaptureProperty.FrameHeight, CAPTURE_HEIGHT);
47     // Webカメラからフレーム取得
48     IplImage frame = Cv.QueryFrame(capture);
49     // Unity上に縦横とフレームのサイズをコンソール出力
50     Debug.Log("width:" + frame.Width + " height:" + frame.Height);
51     // おなじみのウィンドウの名前設定
52     Cv.NamedWindow("FaceDetect");
53
54     // SVM = サポートベクタマシン 機械学習の1つ
55     CvSVM svm = new CvSVM ();
56     // 目標精度 (第3引数) に向かって反復アルゴリズムを行う
57     CvTermCriteria criteria = new CvTermCriteria (CriteriaType.Epsilon, 1000, double.Epsilon);
58     // SVM用のパラメータ設定
59     CvSVMParams param = new CvSVMParams (CvSVM.C_SVC, CvSVM.RBF, 10.0, 8.0, 1.0, 10.0, 0.5, 0.1, null, crit
60 }
61
62 // Update is called once per frame
63 void Update()
64 {
65     IplImage frame = Cv.QueryFrame(capture);
66
67     using (IplImage img = Cv.CloneImage(frame))
68     using (IplImage smallImg = new IplImage(new CvSize(Cv.Round(img.Width / Scale), Cv.Round(img.Height / Sc
69     {
70         // イメージの準備
71         using (IplImage gray = new IplImage(img.Size, BitDepth.US, 1))
72         {
73             Cv.CvtColor(img, gray, ColorConversion.BgrToGray);
74             Cv.Resize(gray, smallImg, Interpolation.Linear);
75             Cv.EqualizeHist(smallImg, smallImg);
76         }
77
78         using (CvMemStorage storage = new CvMemStorage())
79         {
80             storage.Clear();
81
82             // ここで顔検出!
83             CvSeqCvAvgComp faces = Cv.HaarDetectObjects(smallImg, cascade, storage, ScaleFactor, MinNeighb
84
85             // 顔が検出できたら円を描きます
86             for (int i = 0; i < faces.Total; i++)
87             {
88                 CvRect r = faces[i].Value.Rect;
89                 CvPoint center = new CvPoint
90                 {
91                     X = Cv.Round((r.X + r.Width * 0.5) * Scale),
92                     Y = Cv.Round((r.Y + r.Height * 0.5) * Scale)
93                 };
94                 int radius = Cv.Round((r.Width + r.Height) * 0.25 * Scale);
95                 img.Circle(center, radius, colors[i % 8], 3, LineType.AntiAlias, 0);
96             }
97         }
98     }
99 }

```

```

CameraScript.cs
selection
1 using UnityEngine;
2 using System.Collections;
3
4 public class CameraScript : MonoBehaviour {
5
6     public GameObject faceDetect;
7     public FaceDetectScript faceDetectScript;
8
9     // Use this for initialization
10 void Start () {
11     faceDetectScript = faceDetect.GetComponent<FaceDetectScript>
12 }
13
14 // Update is called once per frame
15 void Update () {
16     Vector2 facepos = faceDetectScript.facepos;
17     Debug.Log("facepos:(" + facepos.x + " " + facepos.y + ")");
18     float alpha = facepos.y * Mathf.PI;
19     float beta = facepos.x * Mathf.PI;
20     Vector3 pos;
21     pos.x = -3.0f * Mathf.Sin(alpha) * Mathf.Cos(beta);
22     pos.y = 3.0f * Mathf.Cos(alpha);
23     pos.z = -3.0f * Mathf.Sin(alpha) * Mathf.Sin(beta);
24     transform.position = pos;
25     Vector3 relativePos = -transform.position;
26     Quaternion rotation = Quaternion.LookRotation(relativePos);
27     transform.rotation = rotation;
28 }
29 }
30

```

顔検出用コード(FaceDetect.cs)

カメラ用コード(CameraScript.cs)

顔検出の方は、どんなことをしているのかコメントでチェック!

顔検出でラケットを操る (ファイル統合1)

先ほど作った跳ね返りゲームと顔検出を統合してみよう。

→ まずはコードやファイルの準備から

1. まずはサンプルコードのファイルのAssetsフォルダ(A)を開く。
2. 先ほどのゲームを作ったファイルのAssetsフォルダ(B)も開く。
3. (B)の2つのコードを(A)の“Scripts”フォルダに移動する。
4. (B)のシーンファイルを(A)の“Scenes”フォルダに移動する。
5. (B)の物理マテリアルを(A)のAssetsフォルダ直下に移動する。
6. Unity上でScenesフォルダを開き, FaceDetectシーンを開く。
7. 画面左のHierarchyのFaceDetectを右クリックしてCopy
8. TheGameシーンを開き, Hierarchy上でしてPasteする。

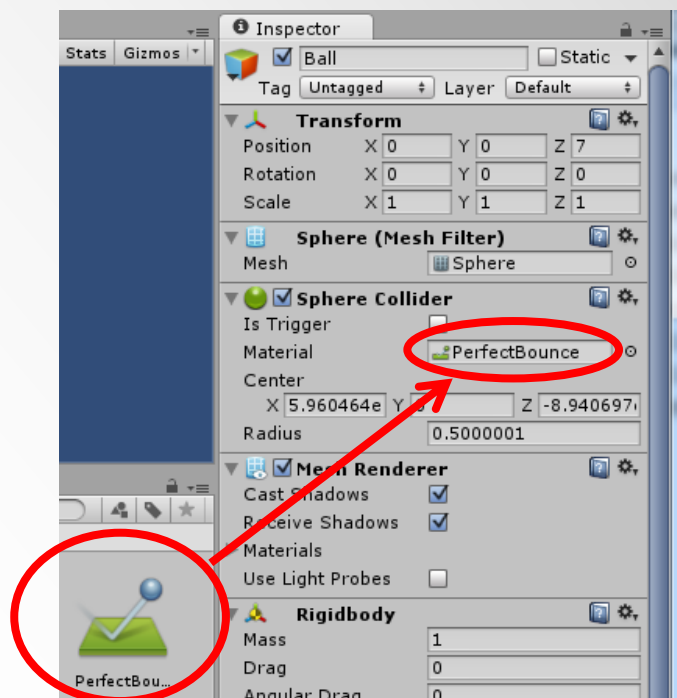


もうちょい続きます

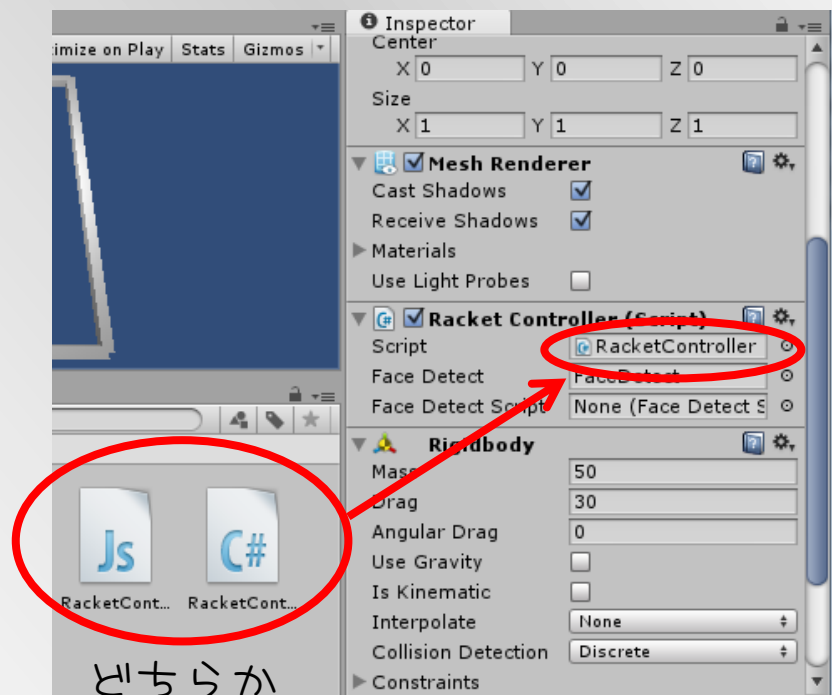
顔検出でラケットを操る (ファイル統合2)

9. Ballをクリック, 右画面のSphere ColliderのMaterialに物理マテリアルをドラッグアンドドロップ

10. 同じようにRacketをクリックして右画面のRacket ControllerのScriptにRacketController.js(cs)をドラッグアンドドロップ



物理マテリアルの適用



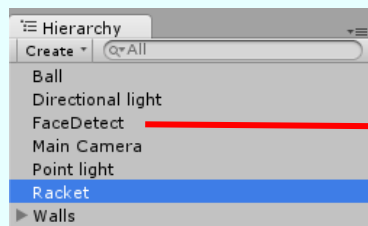
スクリプトの適用

顔検出でラケットを操る (プログラミング)

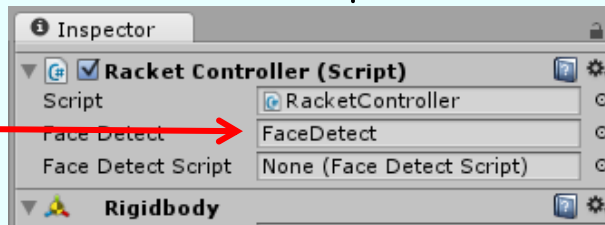
では実際に跳ね返りゲームと顔検出を組み合わせてみよう。
→顔を動かすとラケットが動くようにプログラミングしよう！

コードやゲーム作成のヒント

- 主にいじるのはRacketController.csだけ
- サンプルコードのカメラ用コードから顔の位置を記述している部分を探そう！(C#だとそのままコピーできる)
- 顔の位置の値は, サンプルコードを実行したときに Hierarchyをクリックすると右側に表示される。
- 「はじめてのUnity」のページに大事なヒントあり。

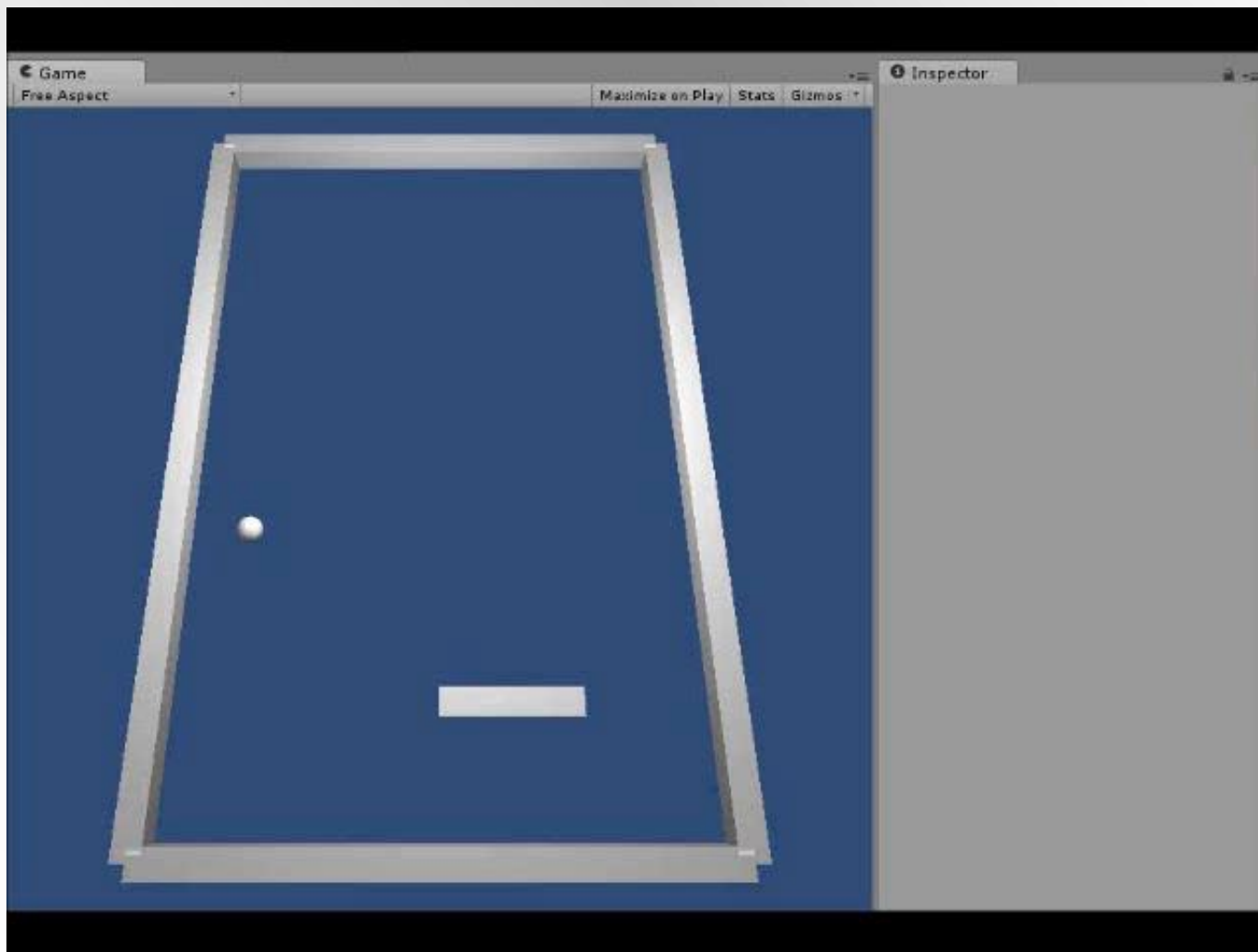


Drag&Drop



顔検出でラケットを操るゲームが完成

下のデモ動画のようなゲームが完成すれば本日の基礎輪は終了！



顔検出を利用したボールの跳ね返しゲーム

おつかれさまでした

このクソゲーの作成を通して, Unityにはインタラクティブな活用方法もあるということを知っていただけたらと思います.

参考文献

- はじめてのUnity <http://japan.unity3d.com/developer/document/tutorial/my-first-unity/O1.html>
- Unity+OpenCvSharp <http://www.youtube.com/watch?v=fRAPaLEIQao>
- No hack, no work <http://warapuri.tumblr.com/post/70283352060/unity-opencv-mac>

Special Thanks

- あくびんフォント <http://pandachan.jp/>
- ターザンオープン愛好会
- 初音ミク