第2回Unity講座

コンピュータ知能学コース 3年 福澤 航大







Oビリヤード(の原型みたいなもの)を作ってみる

Oビリヤード版の上で、ボールをぶつけて動



○盤と(発射するほうの)ボールと外枠を作る ○球にスクリプトを導入し、十字キーで球を動かしてみる

○その過程でGameObjectやスクリプトの基礎を 勉強する

下準備~プロジェクト生成

○Unityを起動して新しくProjectを立ち上げよう
 ○スタート画面から「New」を選択
 ○プロジェクト名を入れ「3D」にチェックが入っていることを確認しCreate

下準備~ファイルの用意

○作ったものをまとめるファイルを用意しておくと便利 ○「Project」タブで右クリック->「Create」->「Folder」で フォルダ生成

○とりあえず下の四つがあれば大丈夫かな?

🗄 Console	🛍 Project	
Create *		Q 4 9 *
🔻 😭 Favorites		Assets 🕨
🔍 All Materi	als	
🔍 All Models	s	
🔍 All Prefab	s	
Q All Script	S	
		Material Prefah Scenes Scrint
Assets		
Material		
🚞 Prefab		
🚞 Scenes		
🚞 Script		



・フロジェクトが立ち上がったら、シーンを保存
 ・ク左上の「File」->「Save Scenes as」を選択
 ・の先ほど作った「Scenes」ファイルに保存しておこう

手順1~ボール、盤の生成

○まずはボールを出してみよ 5 O「Hierarchy」タブで右ク $\eta \gamma \gamma$ -> $\lceil 3D Object \rceil$ ->

「Sphere」を選択



手順1~ボール、盤の生成

 ○盤も作ってみよう
 ○「Hierarchy」タブで右ク リック->「3D Object」-> 「<u>Plane」</u>を選択



手順1~ボール、盤の生成

 ・作ったら「Hierarchy」を右の図のように調整しよう

 「Plane」が「Sphere」の中に入っていたらドラッグ&ドロップで外へ







○せめて色がないとわかりづらい

手順2~Materialを使おう

OUnityで色をつけたいときは Materialが便利 「Project」タブで右クリック-> 「Create」->「Material」 で生

「Cleale」->「Malenal」 C. 成

○赤と緑を使うのでそれっぽい名前 に

	Create	>	Folder
	Show in Explorer		C# Script
	Open		Javascript
	Delete		Shader
	Open Scene Additive		Testing
	Import New Asset		Scene
	Import Package	>	Prefab
	Export Package		Audio Mixer
	Find References In Scene		Material
	Select Dependencies		Lens Flare
lsse.	Refresh	Ctrl+R	Render Textu
	Reimport		Lightmap Par
	Reimport All		Sprites
	Run API Updater		Animator Cor
	Open C# Project		Animation
			Animator Ov
			Avatar Mask
			Physic Mater
			Physics Mate
🗎 Mater	ial		GUI Skin
			Custom Font
	Į.	ינ 🤁	Legacy

ameters ntroller erride Controller rial 2D

手順2~Materialを使おう

○赤いMaterialのInspecter を表示 ○「Albedo」を選択->Colorウィンドウで赤色を選ぼう ○緑のMaterialも同様に作ってみよう



手順2~Materialを使おう

○できたら球に赤、盤に緑のMaterialをドラッグ&ドロップ

🚝 Hierarchy	∂	# Scene	C Game	節 Asset Store	🕫 Animator	
Create * (Q*All		Shaded	* 2D	※ 📣) 🖬 💌	Gizmos * (Q*All	
🔻 🚭 Biliyard2*	*≡					
Main Camera						
Directional Light						V Z
Plane						
Sphere						∀ ∪ ▼ π
			1			
E Canaala 🖉 Brajast	_					2
Create T	_					
	A	at a Matarial		C		
	ASS	e 🕨 Material				
			× 🗖			
Q All Scripts						
🔻 🚞 Assets		Green physic	s-ma R	Red		
aterial 🔤 Material						
🚔 Prefab						
a Scenes						
🚞 Script						





○ひとまず見やすくなったので、ここから位置や 向き・大きさの調整

手順3~Transformをいじろう

TransformはUnityの全ての GameObjectに与えられるインスタ ンス

- ここにGameObjectの位置 (Position)・向き(Rotation)・大きさ (Scale)の情報がある
- ○いずれもx,y,zの三つのステータスを持つ



手順3~Transformをいじろう

○球と盤のTransformを右のように変更しよう

○変更するときにViewでオブ
 ジェクトがどう変化するかも
 合わせて確認しよう



Inspector			<u> </u>	
🧊 🗹 Plane			📃 🗌 Static 🔻	
Tag Untagg	ed ≉]Laye	r Default	+	
🔻 🙏 🛛 Transfor	m		ې 🔝	•
Position	× 0 ×	r 0	Z 0	
Rotation	× 0 ×	r 0	Z 0	
Scale	X 3	r 1	Z 5	
Inspector			2	- -
👕 🗹 Sphere	ł		📃 🗌 Statio	•
Tag Untage	ed 🕴 Lay	/er Default		+
🔻 🙏 🛛 Transfo	'm			₿.
Position	× 0	Y 0.5	Z -15	
Rotation	X 0	Y 0	Z 0	
Scale	X 1	Y 1	Z 1	

手順3~Transformをいじろう

ウカメラのTransformも合わせ ておこう

 ・変更したら、一度Game
 ビューでカメラの位置があっ
 ているか確認

↓カメラ

Inspector			≟ •≡
👕 🗹 Main Camera			🗌 Static 🔻
Tag MainCamera	÷ Lay	er Default	*
🔨 🙏 Transform			🔯 🌣,
Position	X 0	Y 5.5	Z -20
Rotation	X 30	Y 0	Z 0
Scale	X 1	Y 1	Z 1







○次は、ボールを動かしてみよう!

「Project」内で右クリック-> 「C# Script」

○適当な名前で作り、ダブルク
 リックするとエディタ(多分
 Visual Studio)が開く

○ちなみにC#で説明しますが JavaScriptも使えます

Main Camera Directional Li	ght		Create	>	Folder	-
Plane Sphere			Show in Explorer		C# Script	
ophere			Open		lavascript	_
			Delete		Shader	>
			Open Scene Additive		Testing	>
			Import New Asset		Scene	
			Import Package	>	Prefab	
			Export Package		Audio Mixer	
			Select Dependencies		Material	
Console	🖨 Project	_	Refresh Reimport	Ctrl+R	Lens Flare Render Texture	
▼ 🚖 Favorites Q All Materia	als	Assets ► Script	Reimport All		Sprites	>
Q All Models Q All Prefab	s	C	Run API Updater		Animator Controller	
Q All Scripts		C#	Open C# Project		Animation	
Assets		RigidScript			Animator Override Controller Avatar Mask	
Scenes					Physic Material	
🚝 Script					Physics Material 2D	
					GUI Skin	
		Script Script			Custom Font	
	_		_		Legacy	>

「Project」内で右クリック-> 「C# Script」

○適当な名前で作り、ダブルク
 リックするとエディタ(多分
 Visual Studio)が開く

○ちなみにC#で説明しますが JavaScriptも使えます

Main Camera Directional Light		Create	>	Folder	li
Plane Sphere		Show in Explorer		C# Script	
		Open		lavascript	
		Delete		Shader	>
		Open Scene Additive		Testing	>
		Import New Asset		Scene	
		Import Package	>	Prefab	
		Export Package Find References In Scene		Audio Mixer	
		Select Dependencies		Material	
🗄 Console 🛛 🗎 Project		Refresh	Ctrl+R	Render Texture	
Create *		Reimport		Lightmap Parameters	
All Materials	Assets ► Script	Reimport All		Sprites	>
Q All Models Q All Prefabs	Cu	Run API Updater Open C# Project		Animator Controller	
Q All Scripts	C#			Animation	
🖲 Assets	RigidScript			Animator Override Controller	
Material				Avatar Mask	
Scenes				Physic Material	
Script 🔤				Physics Material 2D	
				GUI Skin	
	Script			Custom Font	
				Legacy	>

「Project」内で右クリック-> 「C# Script」

○適当な名前で作り、ダブルク
 リックするとエディタ(多分
 Visual Studio)が開く

○ちなみにC#で説明しますが JavaScriptも使えます

Main Camera Directional Light		Create	>	Folder	-
Sphere		Show in Explorer		C# Script	
		Open		Javascript	-
		Delete		Shader	>
		Open Scene Additive		Testing	>
		Import New Asset		Scene	
		Import Package	>	Prefab	
		Export Package		Audio Mixer	
		Select Dependencies		Material	
🗄 Console 🔎 Project		Refresh	Ctrl+R	Render Texture	
Create *		Reimport		Lightmap Parameters	
Favorites	Assets ⊨ Script	Reimport All		Sprites	>
Q All Models Q All Prefabs	C II	Run API Updater		Animator Controller	
Q All Scripts	C#	Open C# Project		Animation	
🔻 🚞 Assets	RigidScript			Animator Override Controller	
Material				Avatar Mask	
Scenes				Physic Material	
aript 🚰 Script				Physics Material 2D	
				GUI Skin	
	Script			Custom Font	
		_		Legacy	>

Oまずはソースを打ち込んでみよう(スクリプト1)



○作ったらセーブ ○その後Unityに戻り、作った Scriptを球にドラッグ&ドロップ(アタッチという)



 Rigidbody.Addforceメ ソッドはRigidbodyがその GameObjectになければ使 えない

「Add Component」->
 「Physics」->「Rigidbody」
 で追加

Inspector	<u>≙</u> .	-=
🔻 ڬ Sphere Collider	🔯 *,	-
	🔏 Edit Collider	Π
Is Trigger		
Material	None (Physic Material) O	
Center	X 0 Y 0 Z 0	
Radius	0.5	
🔻 🖳 🗹 Mesh Renderer	🛐 \$,	
Cast Shadows	On +	
Receive Shadows		
Motion Vectors	Per Object Motion +	
▼ Materials		
Size	1	
Element 0	€Red O	
Light Probes	Blend Probes +	
Reflection Probes	Blend Probes +	
Anchor Override	None (Transform) O	
🔻 🙏 Rigidbody	🛐 * ,	
Mass	1	
Drag	0	
Angular Drag	0.05	
Use Gravity		
Is Kinematic		
Interpolate	None +	
Collision Detection	Discrete +	
▶ Constraints		
Red	[i] \$.	
Shader Standard	•	
	Add Component	
		*

25

手順4~ボールを動かそう(解説)

●UnityのScriptには作った時点でStart関数とUpdate関 数が記述されている

○Start関数はシーン起動時に一度だけ呼ばれる関数

○Update関数は<u>ゲームを実行している間常に毎フレー</u> <u>ムごとに呼び出される</u>関数

○ (とりあえず第2、3回はUpdateしか使わないです)

手順4~ボールを動かそう(解説)

 ORigidbodyは物理演算を行うためのコンポーネント (速度、加速度、重力計算などにもこれが必要)
 ORigidbody.AddForceはGameObjectに「力を加える」メソッド

手順4発展~変数を使うには?

○作ったスクリプトを次のように改変し、実行してみよう



手順4発展~変数を使うには? (解説)

○変数ももちろん使える

 Rigidbody.AddForceの場合は三要素の各値にint、 float、double問わず全て使える

○これでスピードを変えるのがかなり楽に

○もっと簡単にやるには…?

手順4発展~Unityのpublic変数

○作ったスクリプトを次のように改変し、Inspectorを確認

RigidScript.c	s⊣⊨×		
🖙 Assembly	/-CSharp	🗸 🗸 😪 RigidScript	♥ ♥ Update()
1	⊟usi	ng System.Collections;	
2	usi	ng System.Collections.Generic;	
3	usi	ng UnityEngine;	
4			
5	🖃 pub	<pre>lic class RigidScript : MonoBehaviour {</pre>	
6			
7		public float speed;	
8			
9			
10		// Use this for initialization	
11		vold Start () {	
12			
13	-	}	
14		// Undate is called once non frame	
15		// opuate is called once per trame	
17		if (Input GetKey(KeyCode UnAppowl)) //上午印工一押下時	
18			
19		transform.GetComponent <rigidbody>().AddEorce(0, 0, 1):</rigidbody>	
20		}	
21		, if (Input.GetKey(KeyCode.DownArrow)) //下矢印丰一押下時	
22		{	
23		transform.GetComponent <rigidbody>().AddForce(0, 0, -1);</rigidbody>	
24		}	
25		if (Input.GetKey(KeyCode.RightArrow)) //右矢印丰一押下時	
26		{	
27		<pre>transform.GetComponent<rigidbody>().AddForce(1, 0, 0);</rigidbody></pre>	
28		}	
29		if (Input.GetKey(KeyCode.LeftArrow)) //左矢印丰一押下時	
		1	
100 /0 +			

手順4発展~Unityのpublic変数

OlnspectorのScriptコンポー ネントに一つ要素が追加さ れているはず

○この値をいろいろ変えて実 行してみよう

Inspector		_
🔻 😂 🗹 Sphere Collider 👘		🔯 🌣, 🔺
	🚡 Edit Collider	
Is Trigger		
Material	None (Physic Material)	0
Center	X 0 Y 0 Z 0	
Radius	0.5	
🔻 🛃 🗹 Mesh Renderer		🔯 🌣, 👩
Cast Shadows	On	;
Receive Shadows		
Motion Vectors	Per Object Motion	;
▼ Materials		
Size	1	
Element 0	€Red	0
Light Probes	Blend Probes	;
Reflection Probes	Blend Probes	;
Anchor Override	None (Transform)	0
🔻 🙏 Rigidbody		🔯 🌣,
Mass	1	
Drag	0	
Angular Drag	0.05	
Use Gravity		
Is Kinematic		
Interpolate	None	+
Collision Detection	Discrete	;
▶ Constraints		
🔻 📴 🗹 Rigid Script (Scrip	t)	🔊 🍣
Script	© RigidScript	o
Speed	20	U
Red		🗊 ö.
Shadar Standard		

手順4発展~Unityのpublic変数(解説)

○public変数にすると他のスクリプトから値を参照できる

O...が、Unityの場合さらにInspectorから値を直接操作
 できるようになる

○とりあえず特段の事情がなければ全部publicにしとくのもあり

手順5~カメラについてきてもらおう

○以下のソースを追加して実行しよう(スクリプト2)



手順5~カメラについてきてもらおう(解説)

OTransformやGameObjectはスクリプト内で変数として使用できる

Othis.transform.positionについて

(GameObject変数).transform.positionでその GameObjectの位置を三次元ベクトルで返す (thisならこのスクリプトをアタッチした自身が返る)

手順5~カメラについてきてもらおう(解説)

○GameObject.Find("GameObject名")で指定した名 前のGameObjectを検索し、gameObject変数とし て返す

・氏ほどのtransform.positionと組み合わせると…
 「GameObject.Find("Apple").transform.position
 でAppleという名のGameObjectの位置を返す



Oビリヤードを完成させます O枠やターゲットボールの追加、スクリプトもちょいと 変えます

OPhysics materialで弾性もつけます

上して