

# 2020年度10月START Unity講座



16回目

講師：幸田 将伍 (@MagurodonDev)

# 今回の講義の目的

2

- ▶ プログラムを自分で読めるようになる
- ▶ Unityを使って自分が実現したいことをできるようになる
- ▶ 自分一人でもゲームを作成できるレベルになる
- ▶ Unityの活用事例を学び、自分の進路に役立てる
- ▶ 実際のエンジニアがどういった仕事の進め方をしているかを知る
- ▶ ゲーム会社のクライアントエンジニアとして就職できるレベルになる

一緒にレベルアップして行きましょう！

# Unity

3

## 2Dパズルゲーム

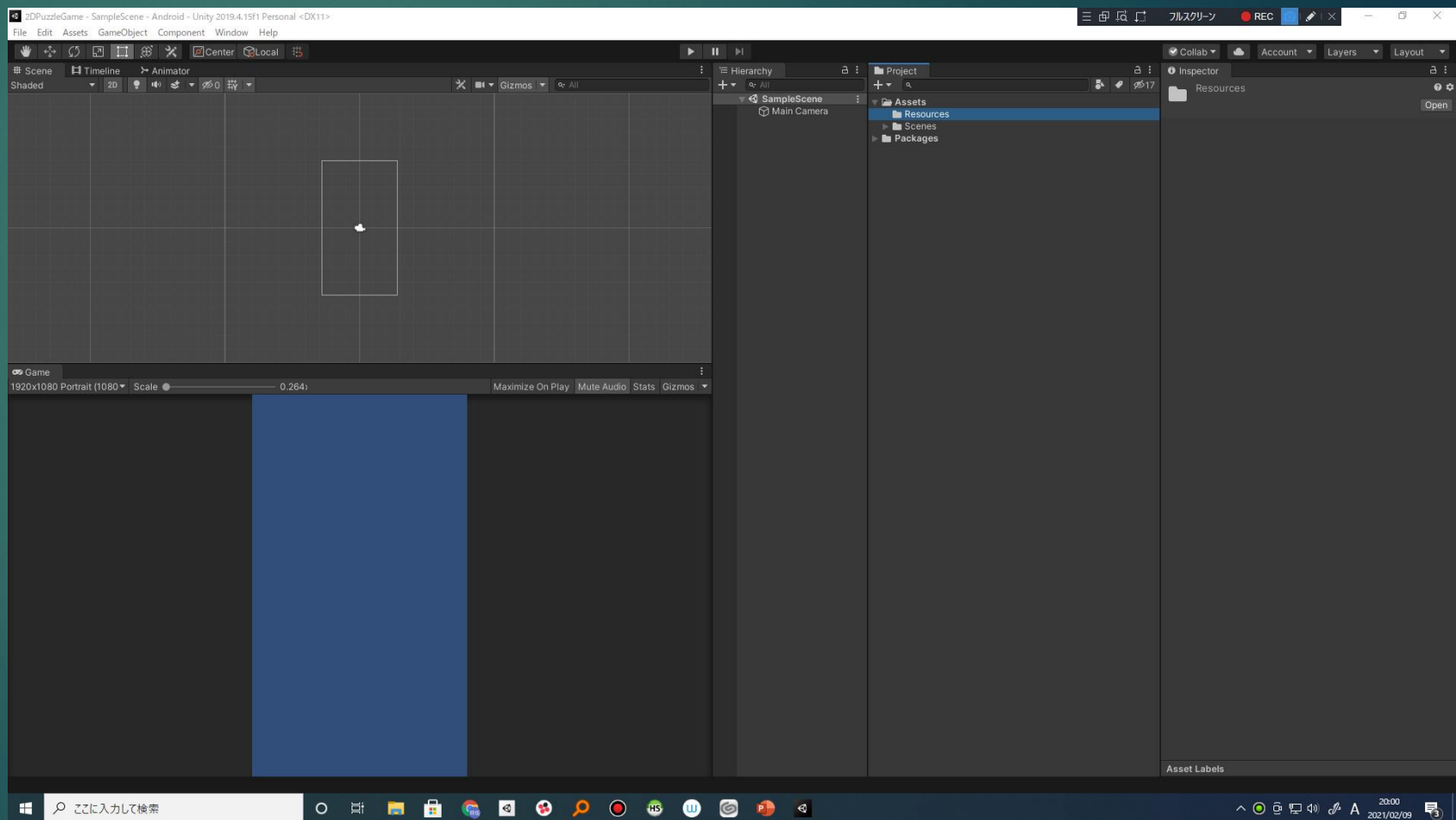
- さて、今回からは心機一転、2Dのパズルゲームを作成していきます。
- 要するにパズドラですね。
- あと4回ですべての実装は難しいので、ロジックの部分を実装していきます。
- もし余裕があれば、属性によって敵にダメージを与えたりしてこうと考えてます。
- では早速ですが、Unityの新規プロジェクトを2Dで作成してください。



# Unity

## 2Dパズルゲーム

- では早速ですが、ロジックを考えていきましょう。
- パズドラのゲーム性はどう言語化できますでしょうか。
- 仕様を考えていきます。
- 1 : 6種類の球が6列5個ランダムで落ちてくる。
- 2 : 操作をして3つ以上同じ球が続くと消える。
- 3 : 消えた分だけ、また上から球が落ちてくる。
- 4 : コンボ数があがればあがるだけダメージ上がる。
- 5 : 6種類の球に対応した属性のダメージを与える。

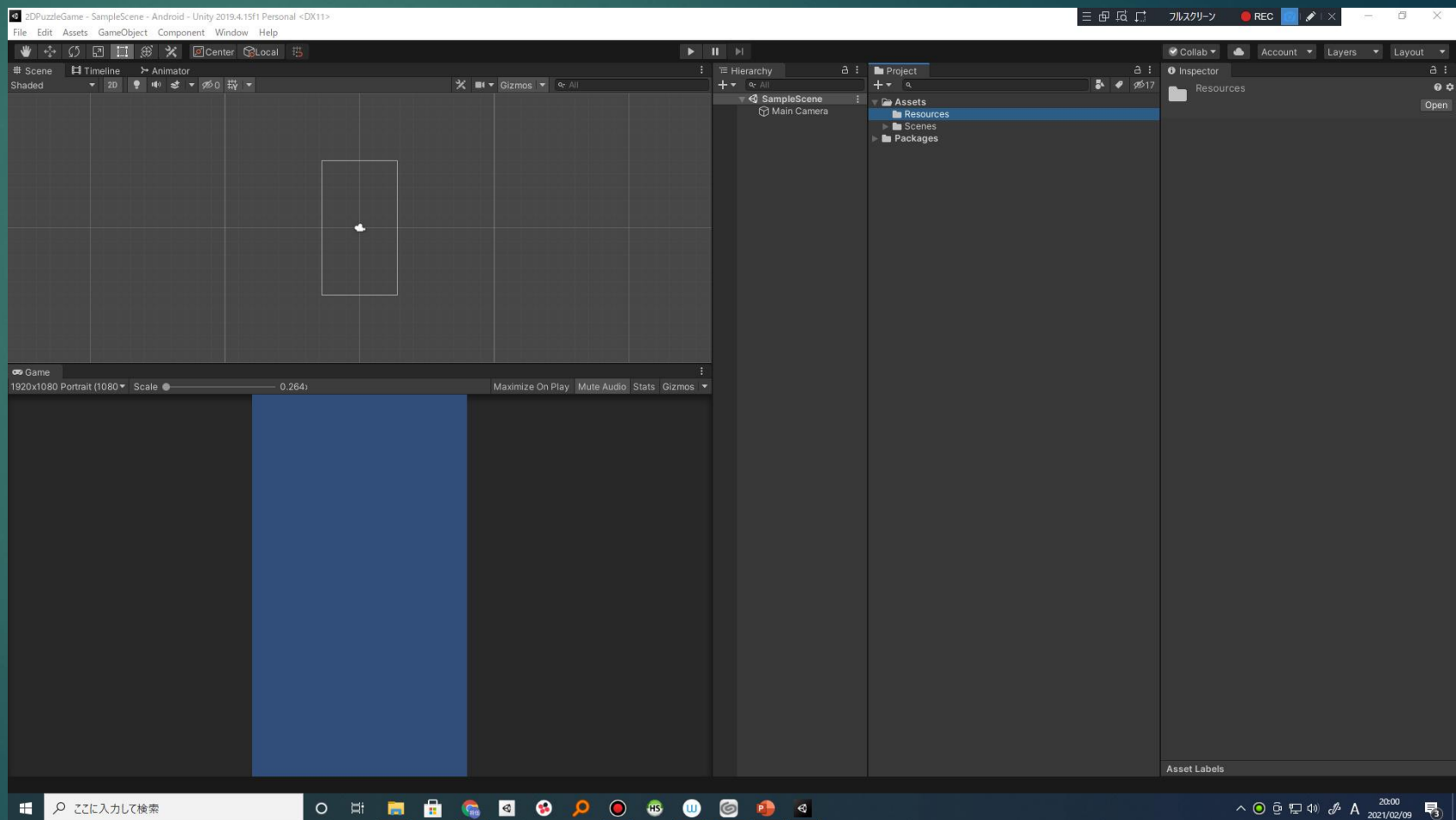


# Unity

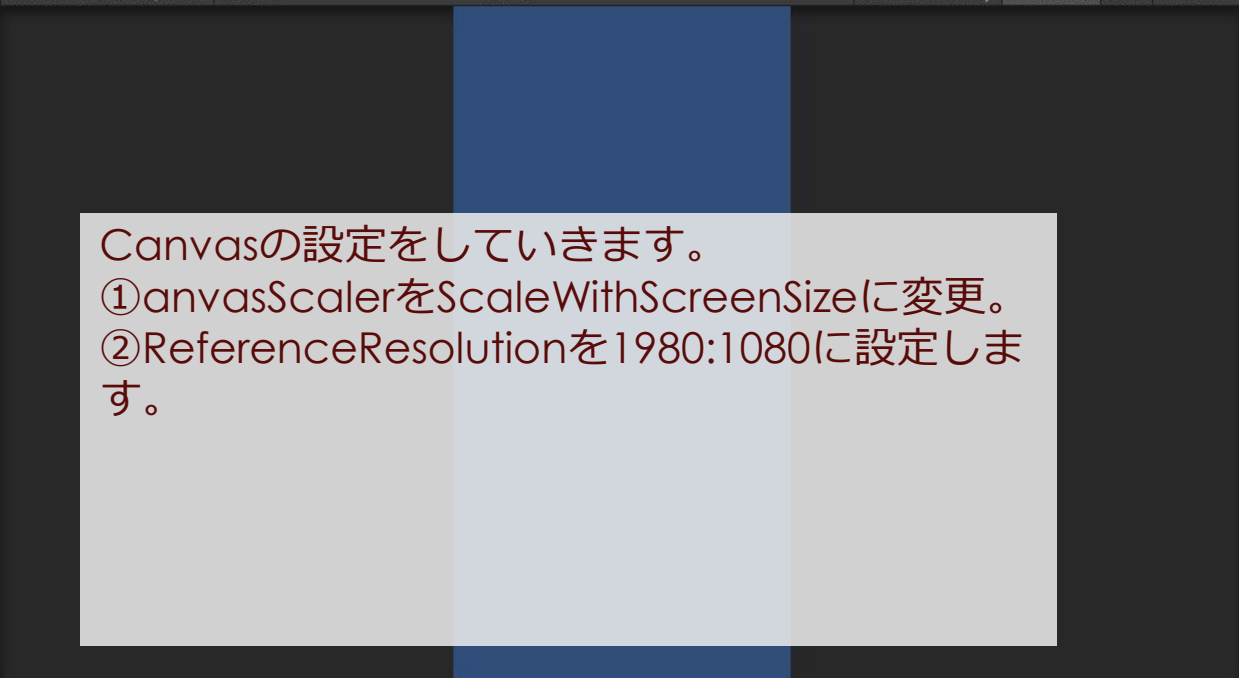
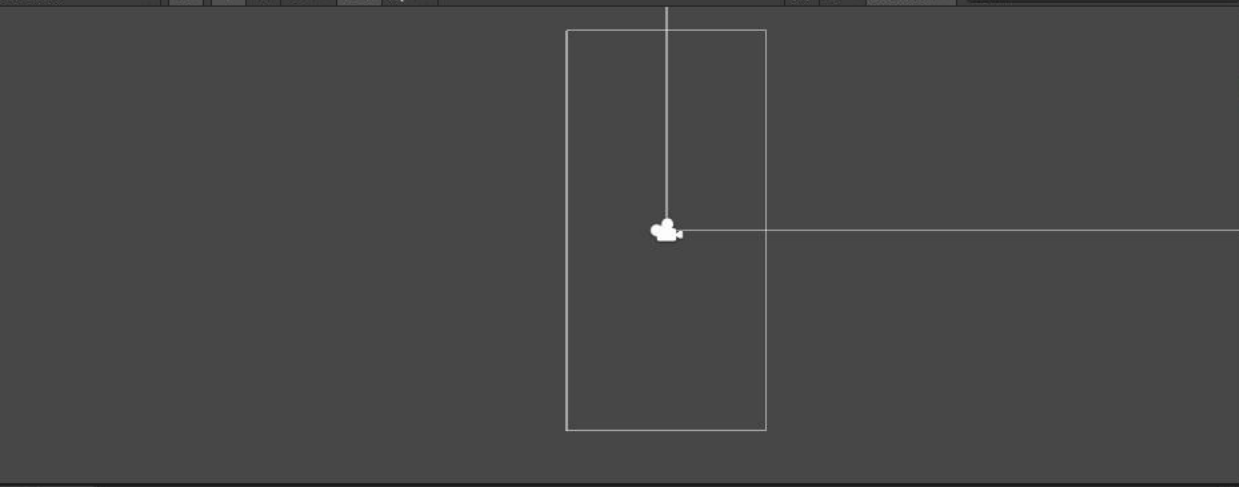
## 2Dパズルゲーム

5

- 完成までの手順を定めていきましょう
- 1 : 5\*6のボールを生成する。
- 2 : ボールを操作する
- 3 : ボールを入れ替える
- 一旦今日はここまで実装してみましょう。
- まず5\*6のボールを作成しましょう。
- ただ闇雲に産んでもいけないので、場所を固定します。
- いろんな方法があるのですが、UnityのUIの機能を使います。

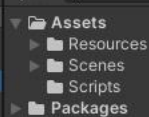
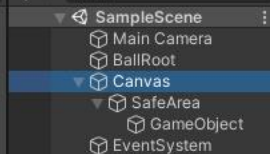


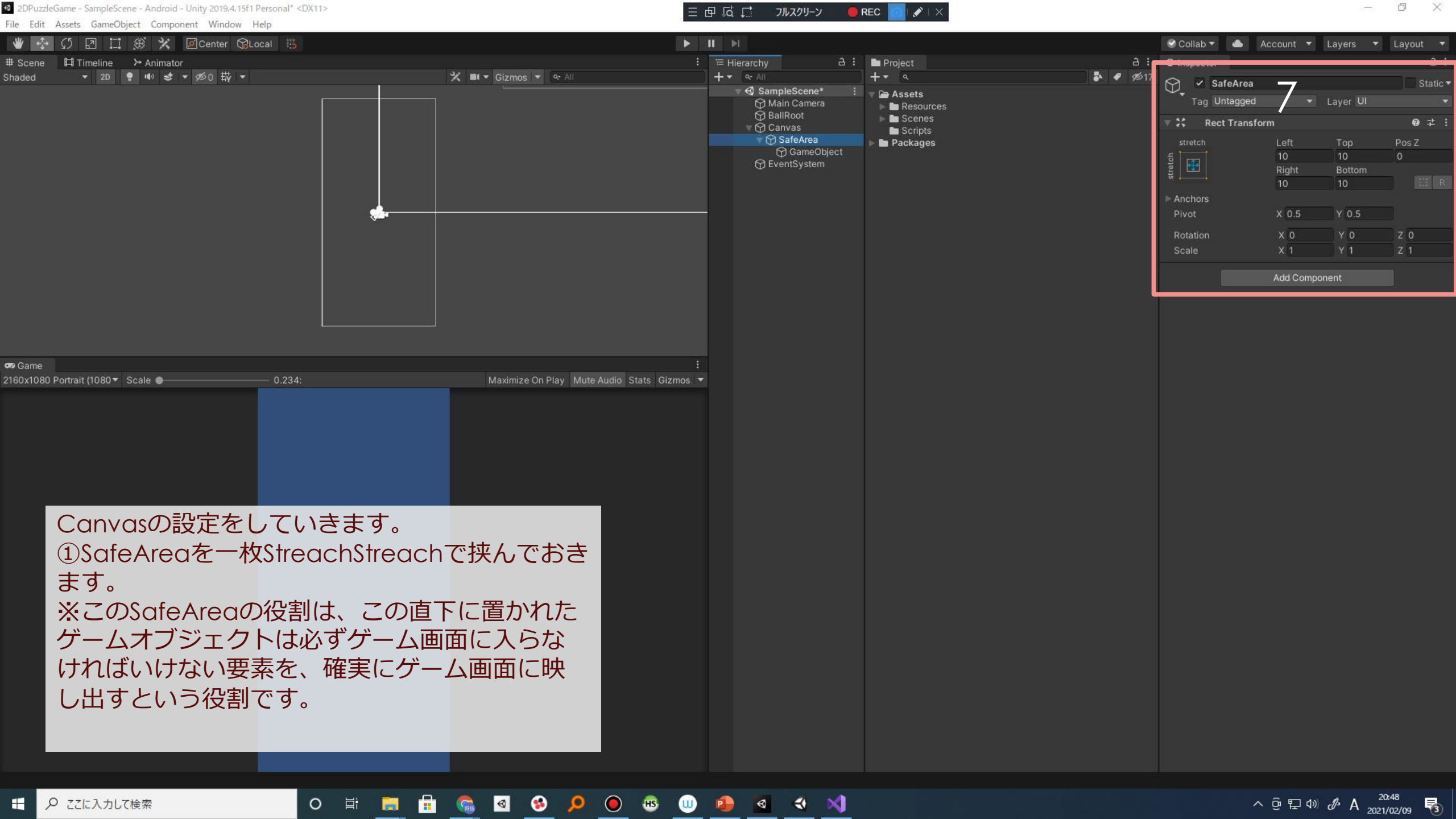


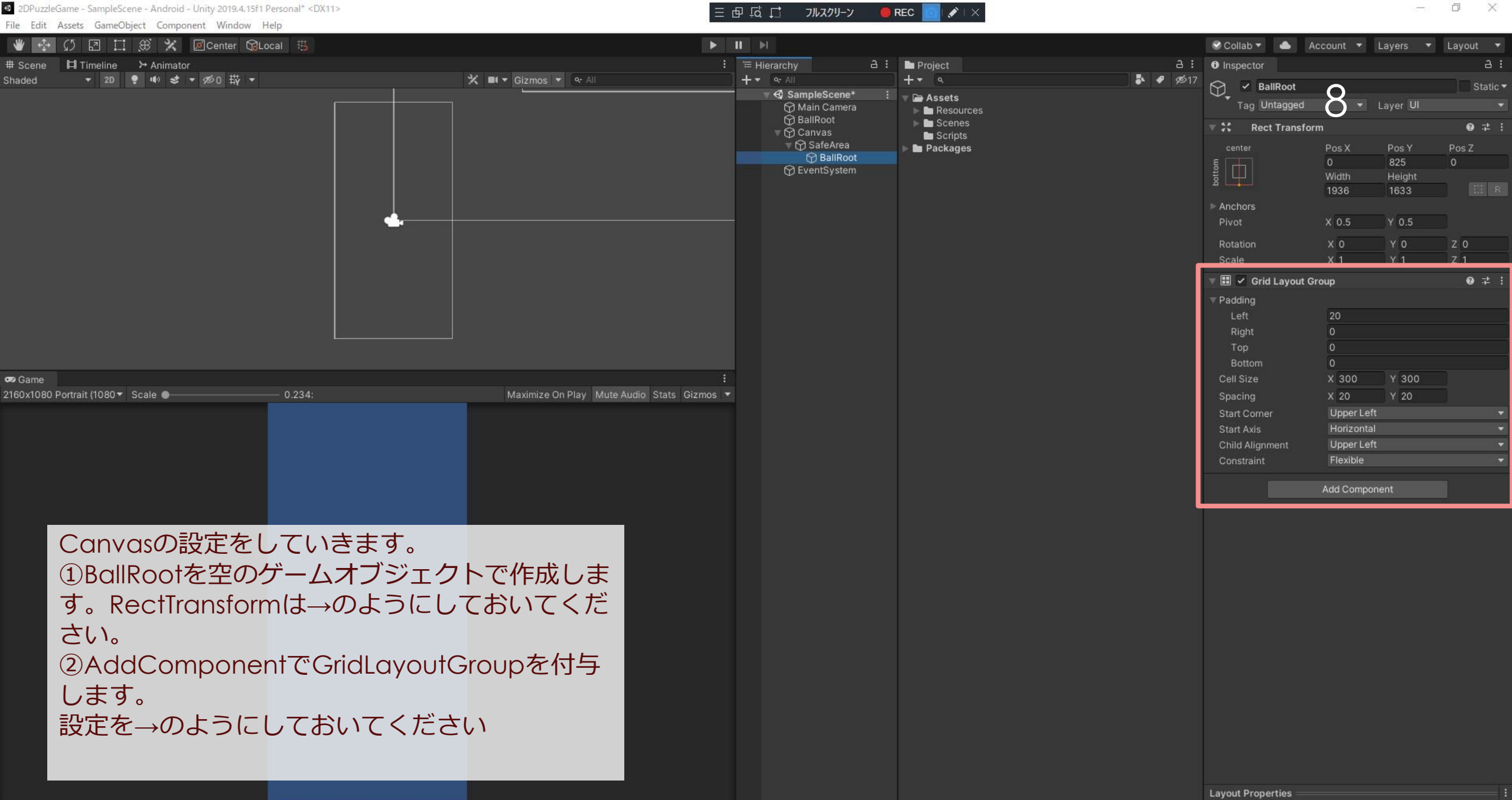


Canvasの設定をしていきます。

- ① canvasScalerをScaleWithScreenSizeに変更。
- ② ReferenceResolutionを1980:1080に設定します。







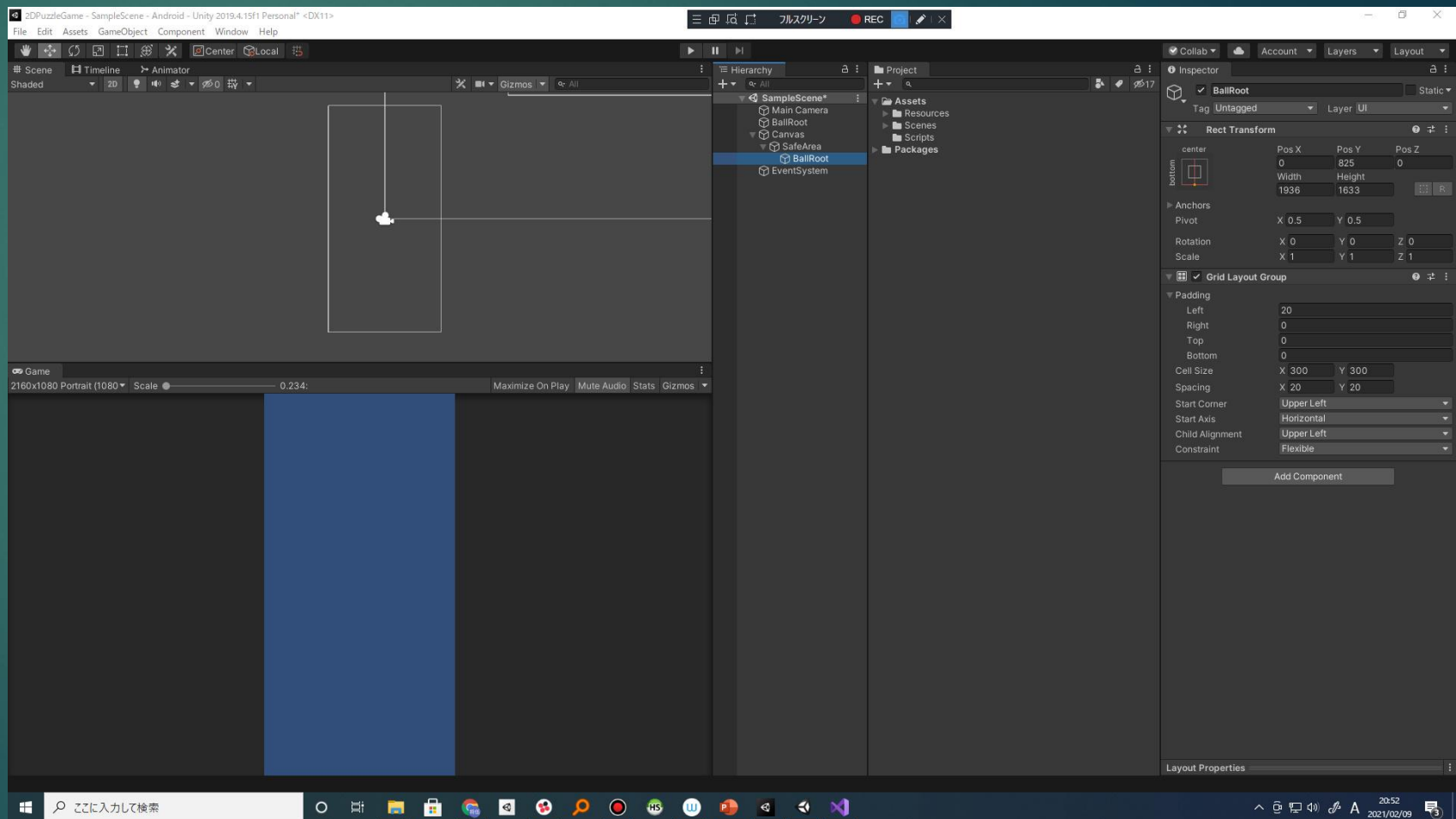


# Unity

## 2Dパズルゲーム

9

- ※GridLayoutGroupとは
- UIの要素をグリッド上に自動配置してくれる機能です。
- ほかにはVerticalLayoutやhorizontalLayoutなど、工夫次第では要素を水平に均等に並べたり、垂直に均等に並べたりすることが可能です。
- 試しに、このBallRayoutの直下でUIのImageを作成して、Ctrl+DでDuplicateしてみてください。
- 均等に6\*5並ぶと思います。

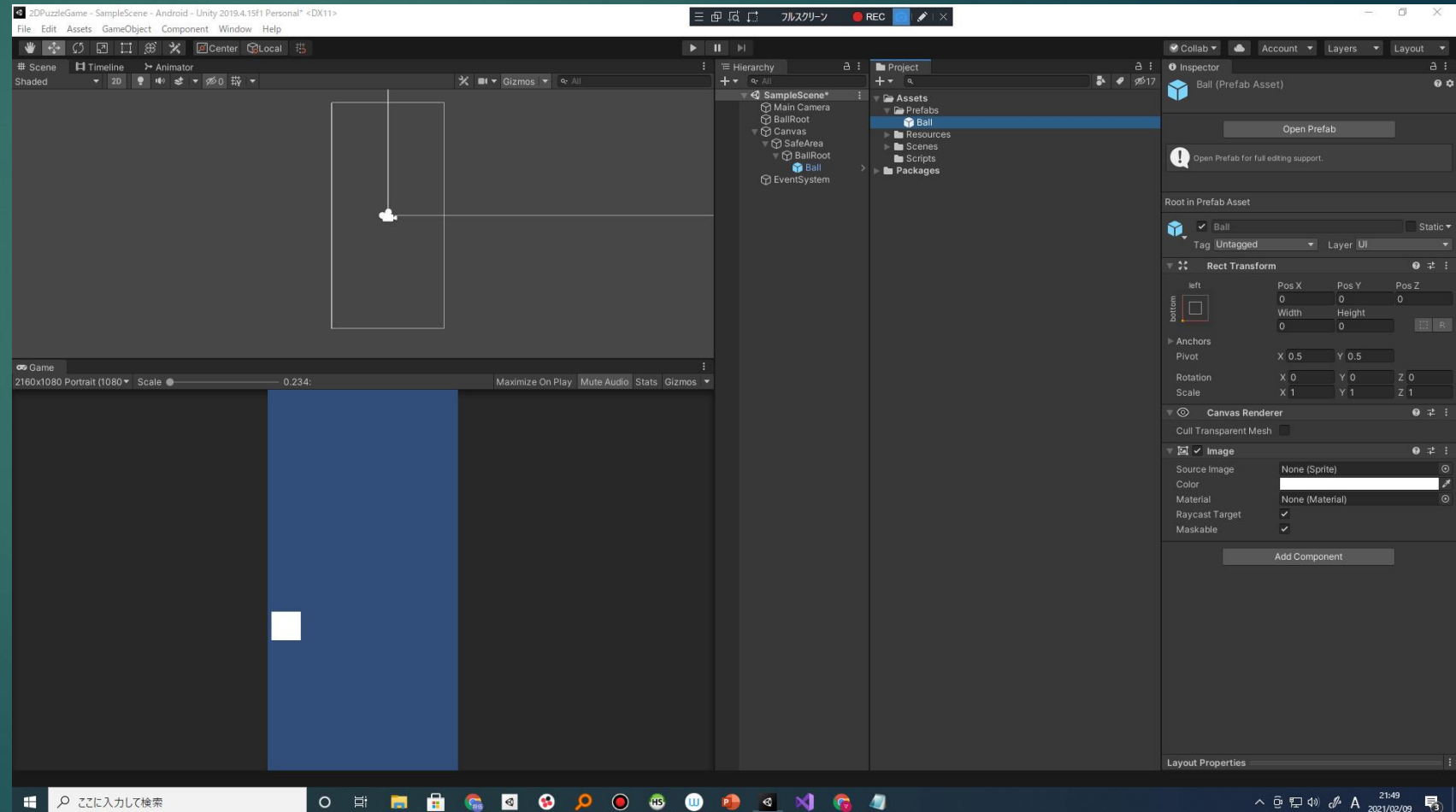


# Unity

## 2Dパズルゲーム

10

- では、自動でボールを産むためのスクリプトを作成します。
- 先ほどDuplicateしたImageを一つだけ残して、Ballという名前をつけます。
- Project欄でPrefabsというフォルダを作成し、その中にD&DしてPrefabにしてください。
- Scriptsフォルダを作成し、BallGenerator.csというスクリプトを作成します。
- このスクリプトの役割はPrefabを30個生むという役割です



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5 public class BallGenerator : MonoBehaviour
6 {
7
8
9
10
11 private void Start()
12 {
13
14
15
16
17 }
```

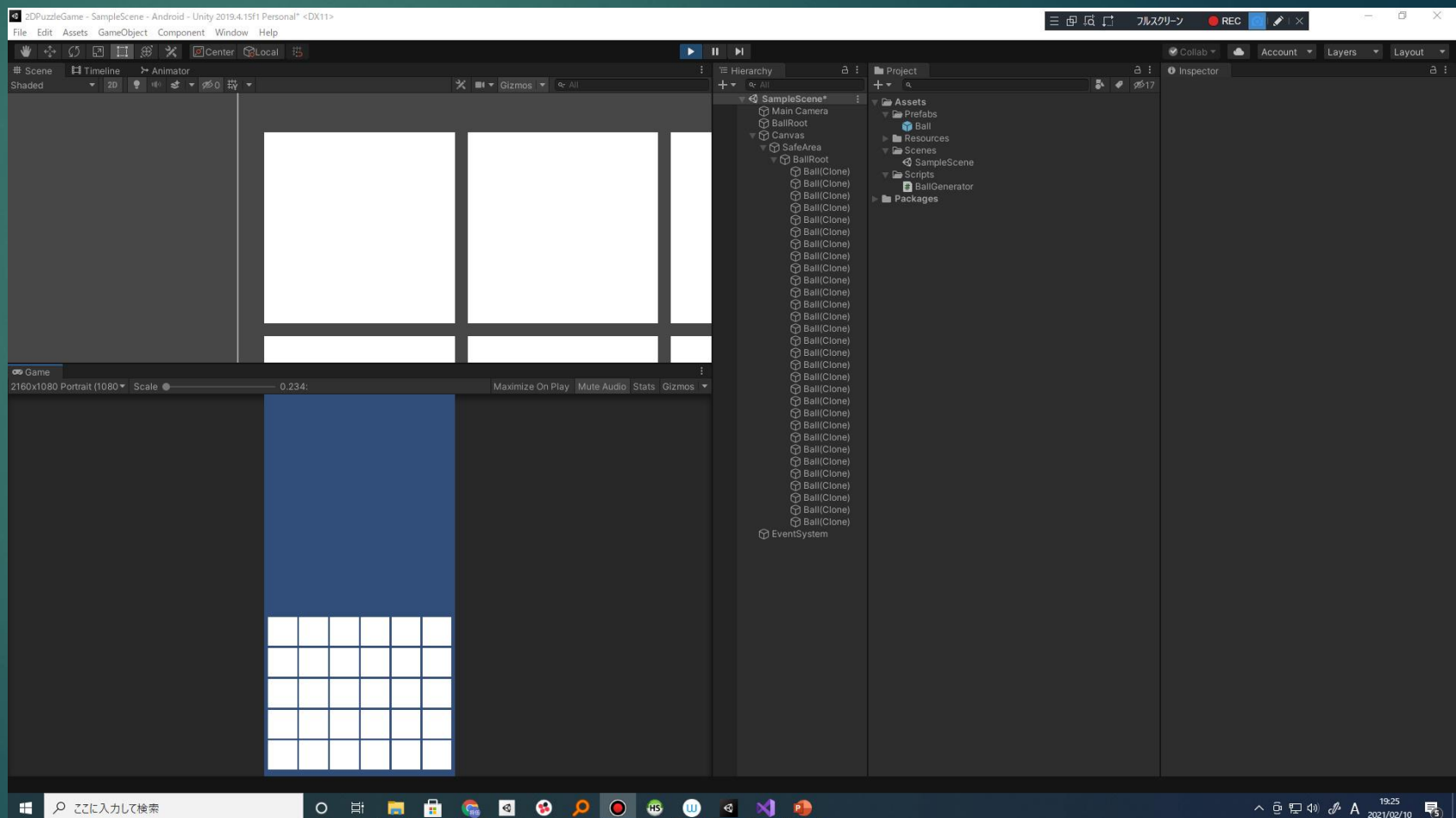
- ①publicでgameObjectの変数を作成してください。 名前はBallとします。
  - ②Startメソッド内に、for文で30回、BallをInstantiateするスクリプトを書いてください
  - ③Instantiateするときに自分の親のオブジェクトを階層構想で見た時の親のオブジェクトに指定してください。
- ※ここはGoogleで調べてみましょう。

# Unity

## 2Dパズルゲーム

12

- BallRootに先ほどのスクリプトをAddし、Hierarchy欄に残していたBallPrefabを削除してプレイしてみてください。
- 右の画像のようになればOKです。
- 次に移動する処理を書いていきましょう。
- Scriptsフォルダの中にBallController.csを作成してください。
- どうやって動かせばよいか各自、30分ほど考えてみてください。

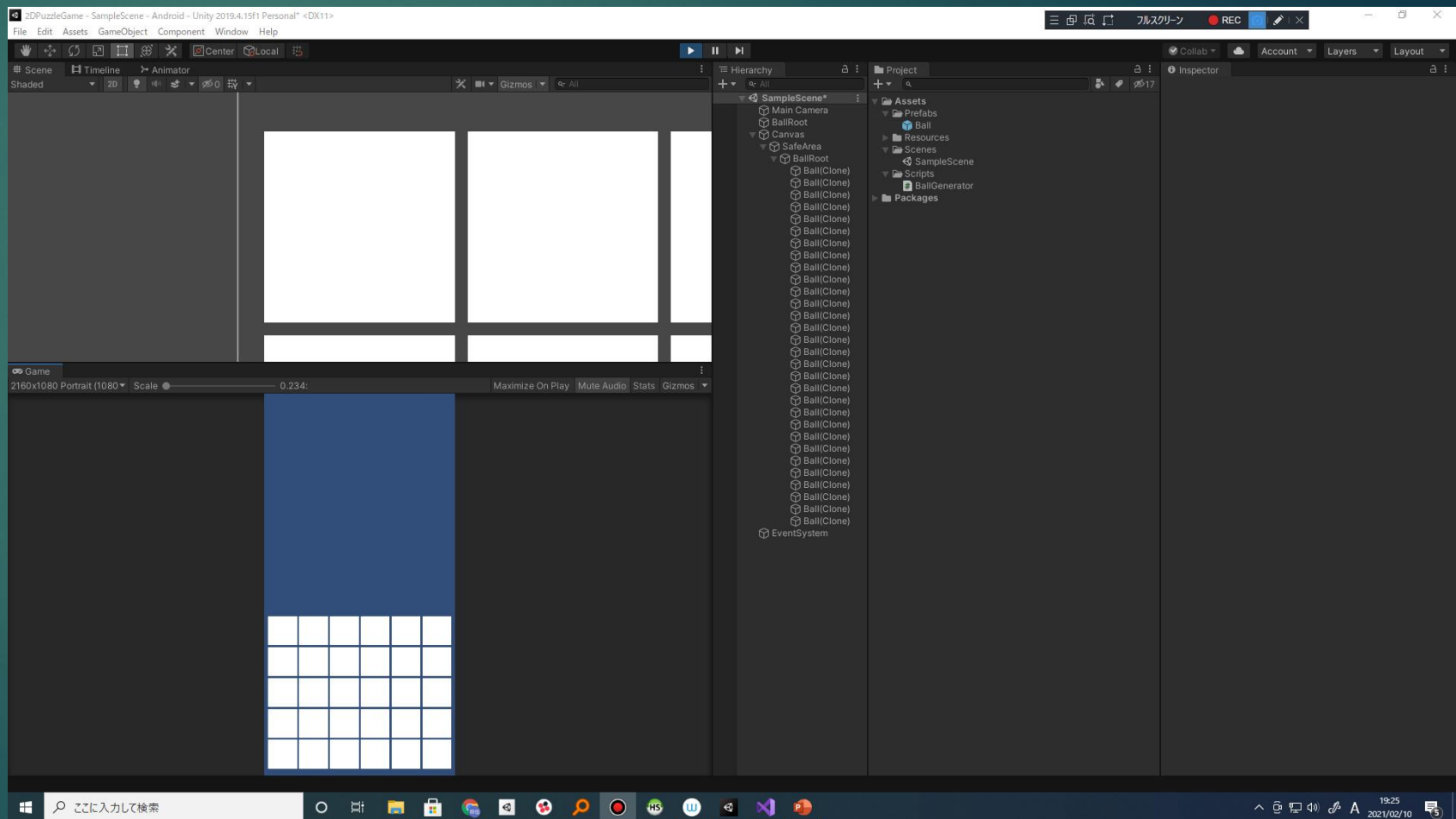


# Unity

## 2Dパズルゲーム

13

- ヒント
- 必要な情報は下記の二つ
- 自分のRectTransform
- マウスの位置
- 自分のRectTransform.Position  
をMouseのPositionに代入し続  
ければよいですね。
- やり方は一つではないです。
- Update内でやっていただいても  
良いし、ほかの方法でも良いで  
す。
- Googleで調べていただいてもか  
まいません。



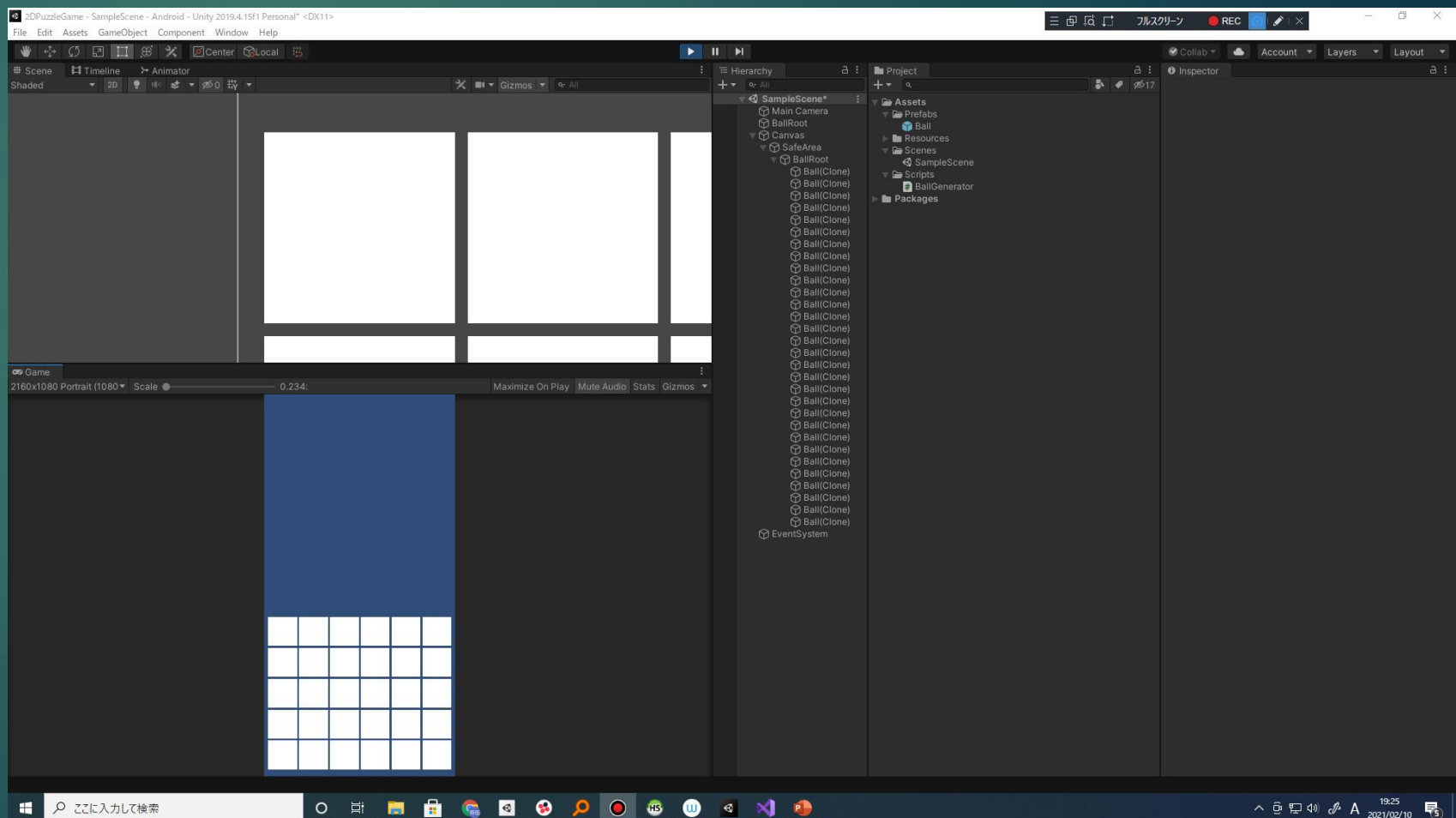


# Unity

## 2Dパズルゲーム

14

- 一応、こんな方法もあるという解説をしておきます
- インターフェースを使う方法
- Unityには様々なインターフェースが最初から用意されています。
- ※インターフェースとは
- インターフェースが定義されたメソッド先では、インターフェース内の機能を絶対に実装しなければいけません。
- その代わりに、その機能を使うことができるようになります。
- 要するに、機能を使うための実装だと思えばいいです。



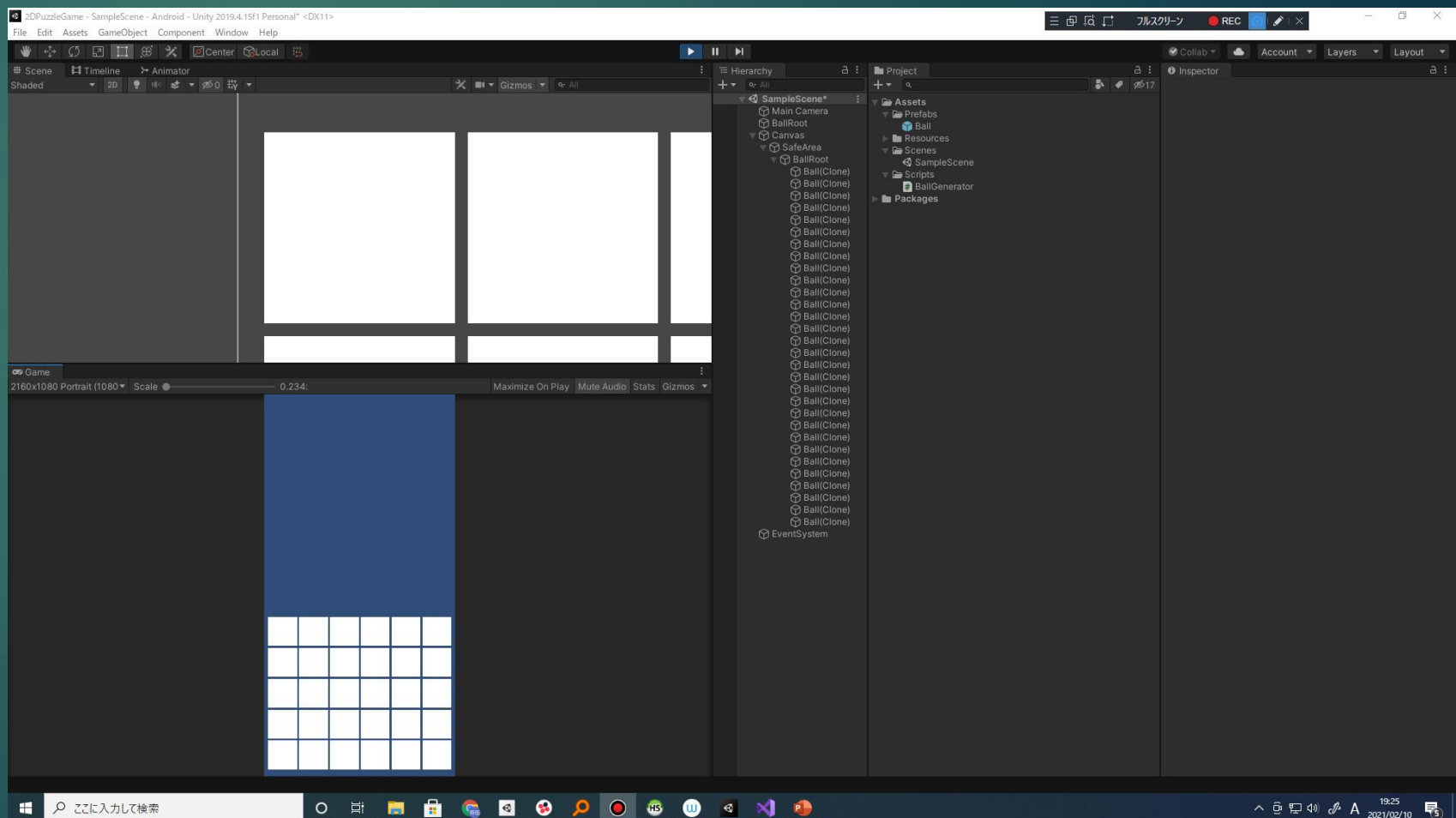


# Unity

## 2Dパズルゲーム

15

- 今回、先生のスクリプトでは、下記インターフェースを実装して機能を作っております。
- ※IPointerDownHandler
- ポインターダウンなので、押された瞬間の判定に使っています
- ※IDragHandler
- ドラッグ中の判定に使っています
- ※IPointerUpHandler
- ポインターアップなので、離れた瞬間の判定に使っています

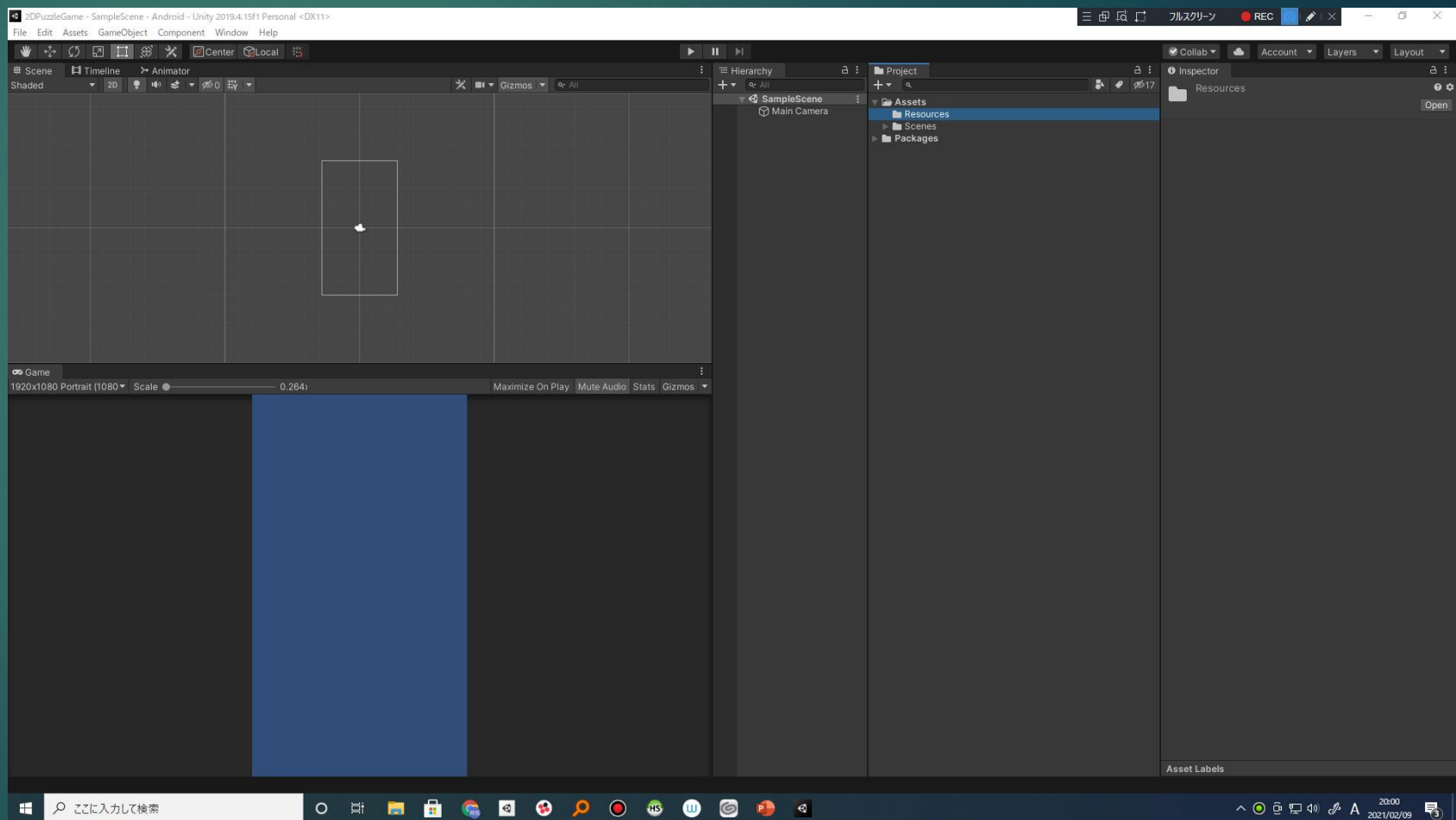


# Unity

## 2Dパズルゲーム

16

- さて、動きましたでしょうか？
  - 次はボールを入れ替えていきます。
  - 必要な情報を洗い出してみます。
  - 1：整列後の自分の位置
  - 2：整列後の相手の位置
  - 3：位置を伝えるための接触判定
- 
- まずここを入れ替えてみましょう。
  - 動かした先のボールに自分整列後の位置を伝えて、自分の位置を相手のボールの位置に変更する。
  - といった感じですね。
  - 入れ替えができれば次の段階ですね。

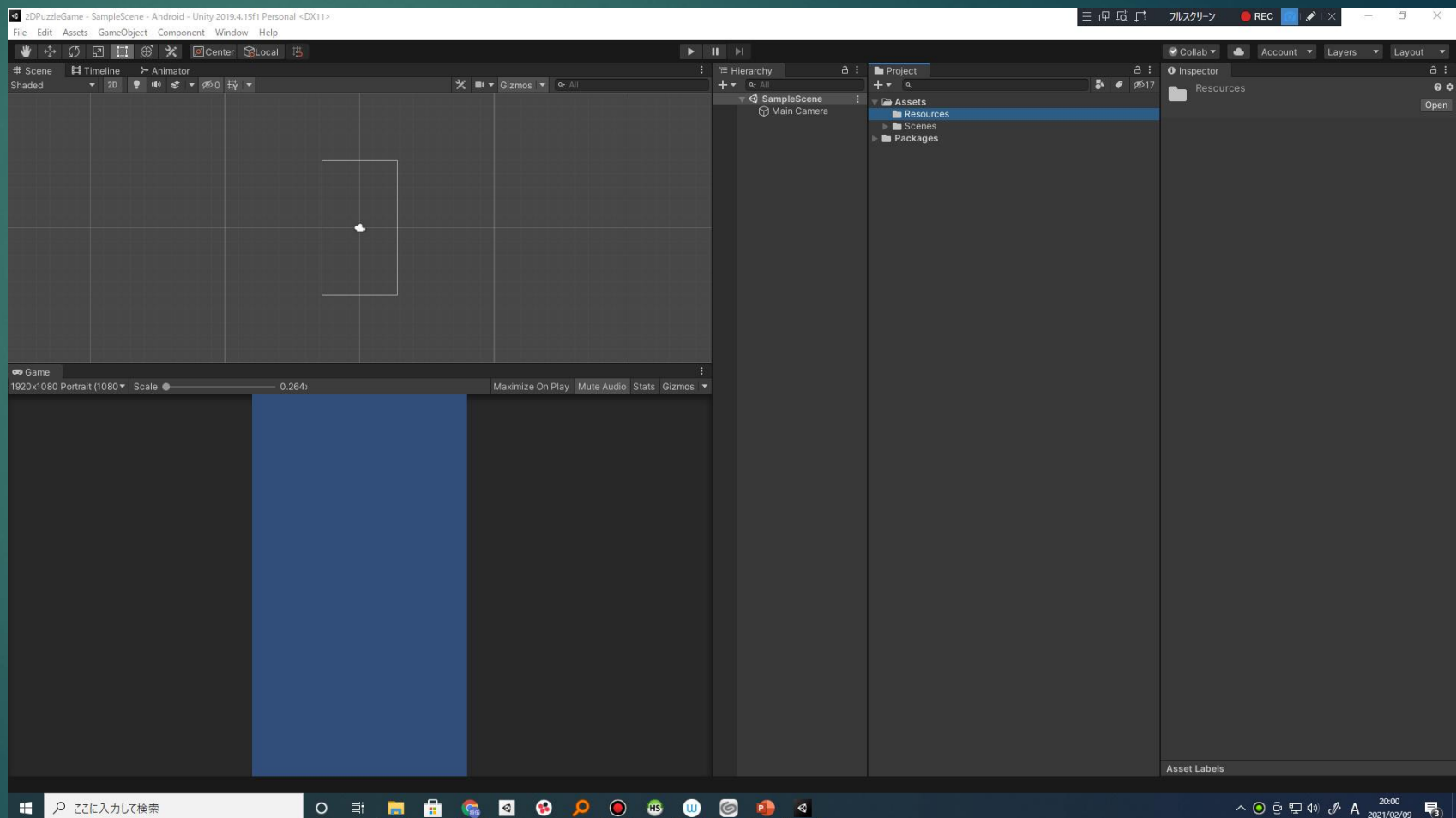


# Unity

## 2Dパズルゲーム

17

- ※整列後の位置の取り方。
- StartやAwake内で自分のTransformから取得する方法が一般的ですが、今回使ったGridLayoutのコンポーネント直下の場合はこの方法が使えません。
- じゃあどうするのかというとUnityのループで1フレーム待つてあげれば取得することができます。
- 待つということはコルーチンを使ってみてください。
- 1フレーム待つには `yield return new WaitForEndOfFrame();` が使えます。

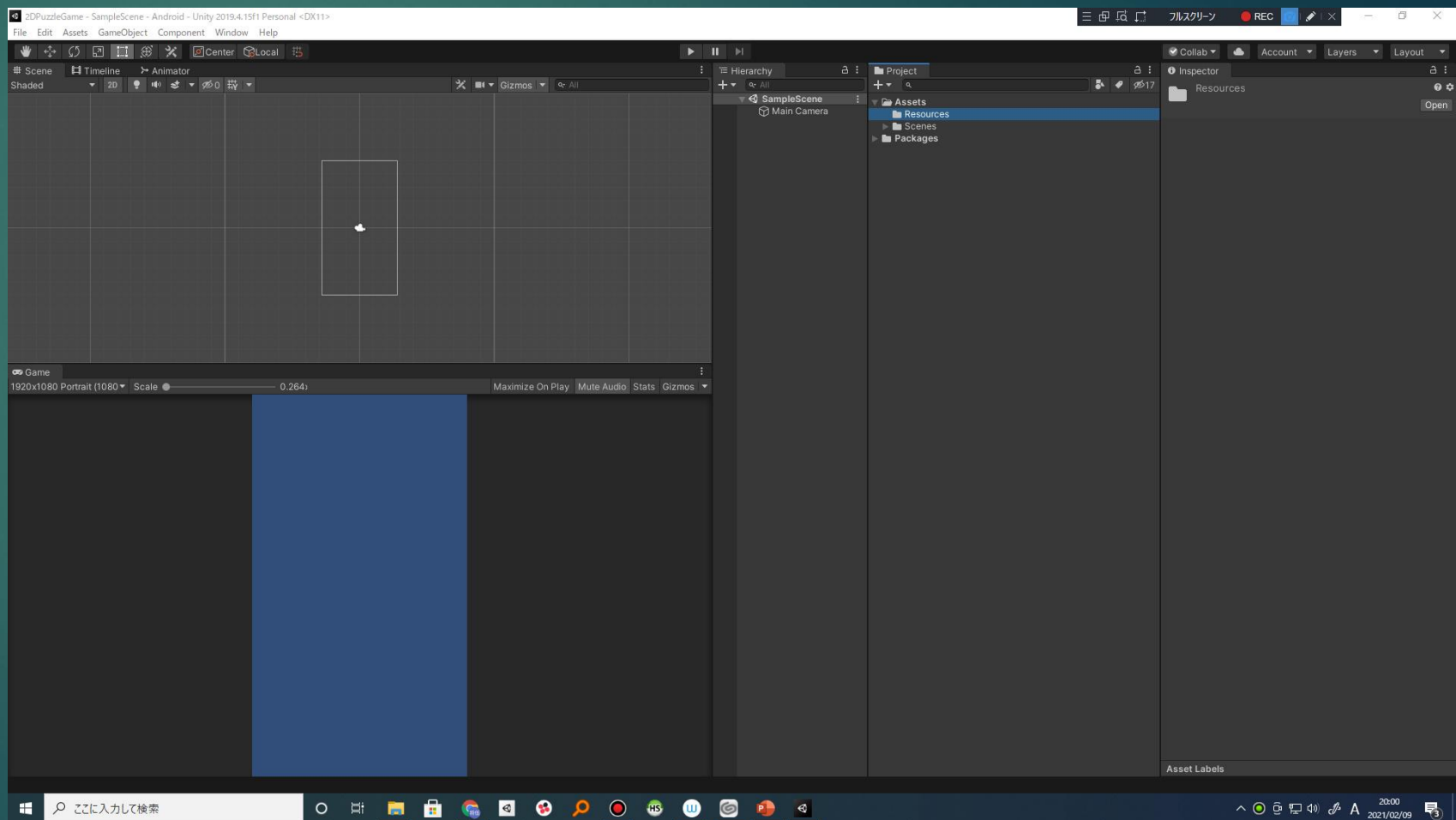


# Unity

## 2Dパズルゲーム

18

- 次の段階として、そのままでは入れ替わったかが分からないので、本日配布した6つのTextureを、開始時にランダムで自分のImageのSpriteにいれるようにしてみましょう。
- 必要な情報
- Spriteの配列
- ランダムの数値の算出方法
- ランダムの数値の算出方法はRandom.Rangeでできますので、これを使ってみてください。



# Unity

## 2Dパズルゲーム

19

- 最終的に右図のようになり、入れ替えが可能になれば本日は終了です。
- 次回はタップから離したら、3つ並んでいるかを確認して消す、新しく消えた球を補充する、コンボを数える。
- までいきましょう。
- まだできていない方は宿題をお願いします。

