

# 2020年度10月START Unity講座



13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

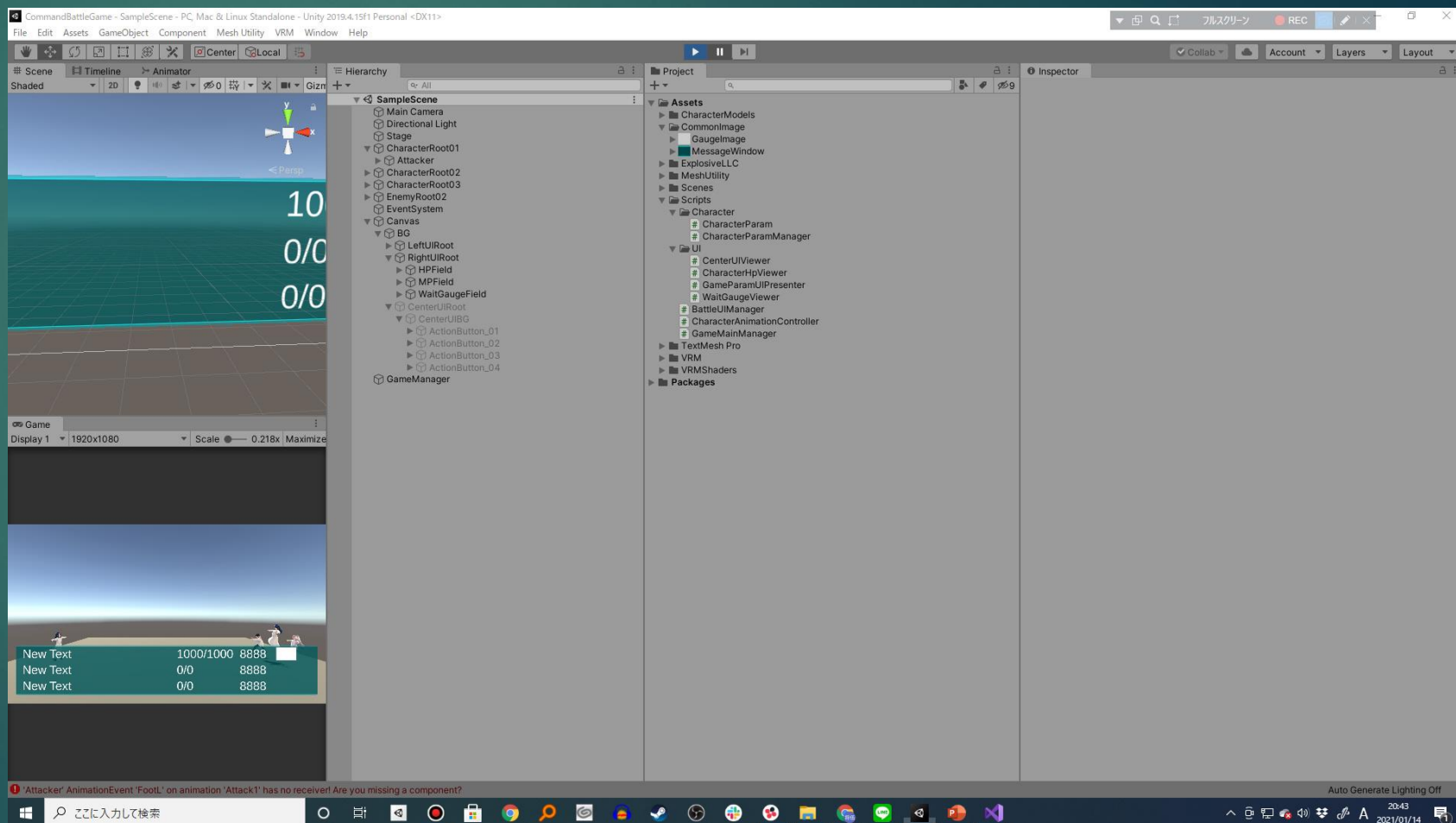
一緒にレベルアップして行きましょう！

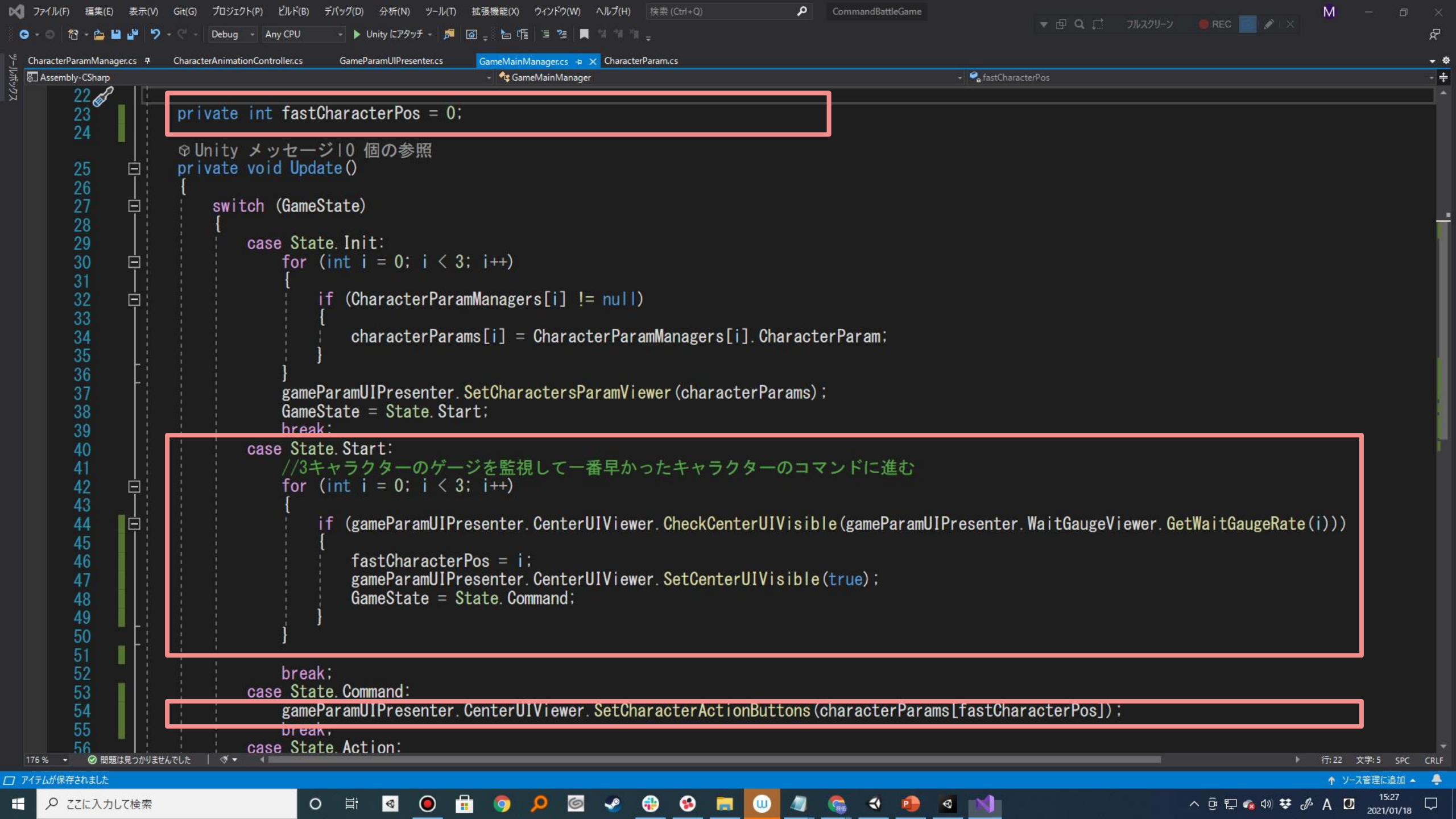
# Unity

3

## コマンドバトルゲーム

- 先週はゲージの実装とボタンの実装を行っていたと思います。
- さて、答え合わせをします。
- 先週出した宿題が下記となります。
- ①Stateの移動の一番早く埋まり切ったキャラクターに変更してください。
- ②また、一番早く埋まり切ったキャラクターの攻撃ボタンの一番上で再生する。
- ③ボタンを押したらCenterUIRootが消えるようにする。



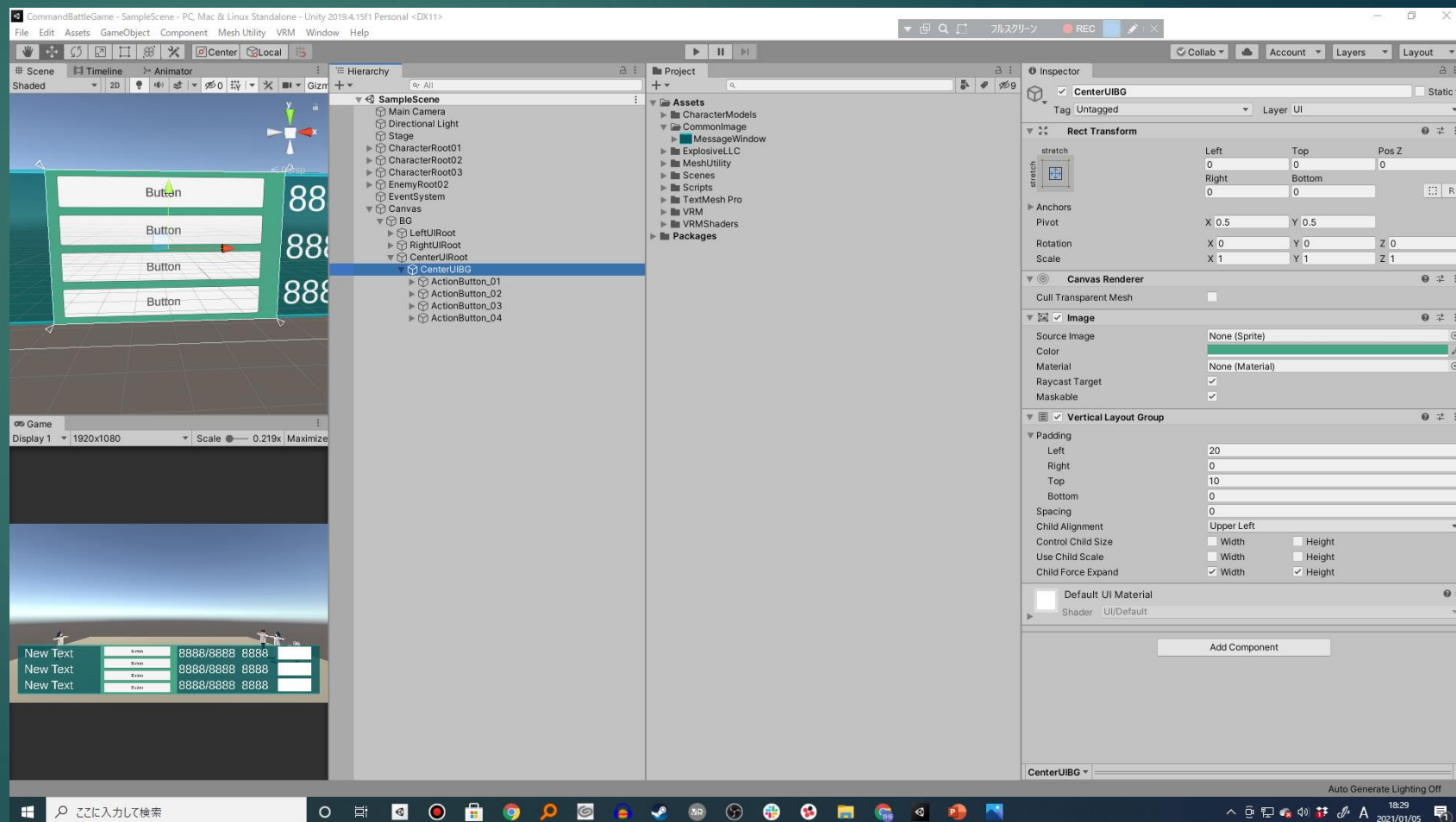


# Unity

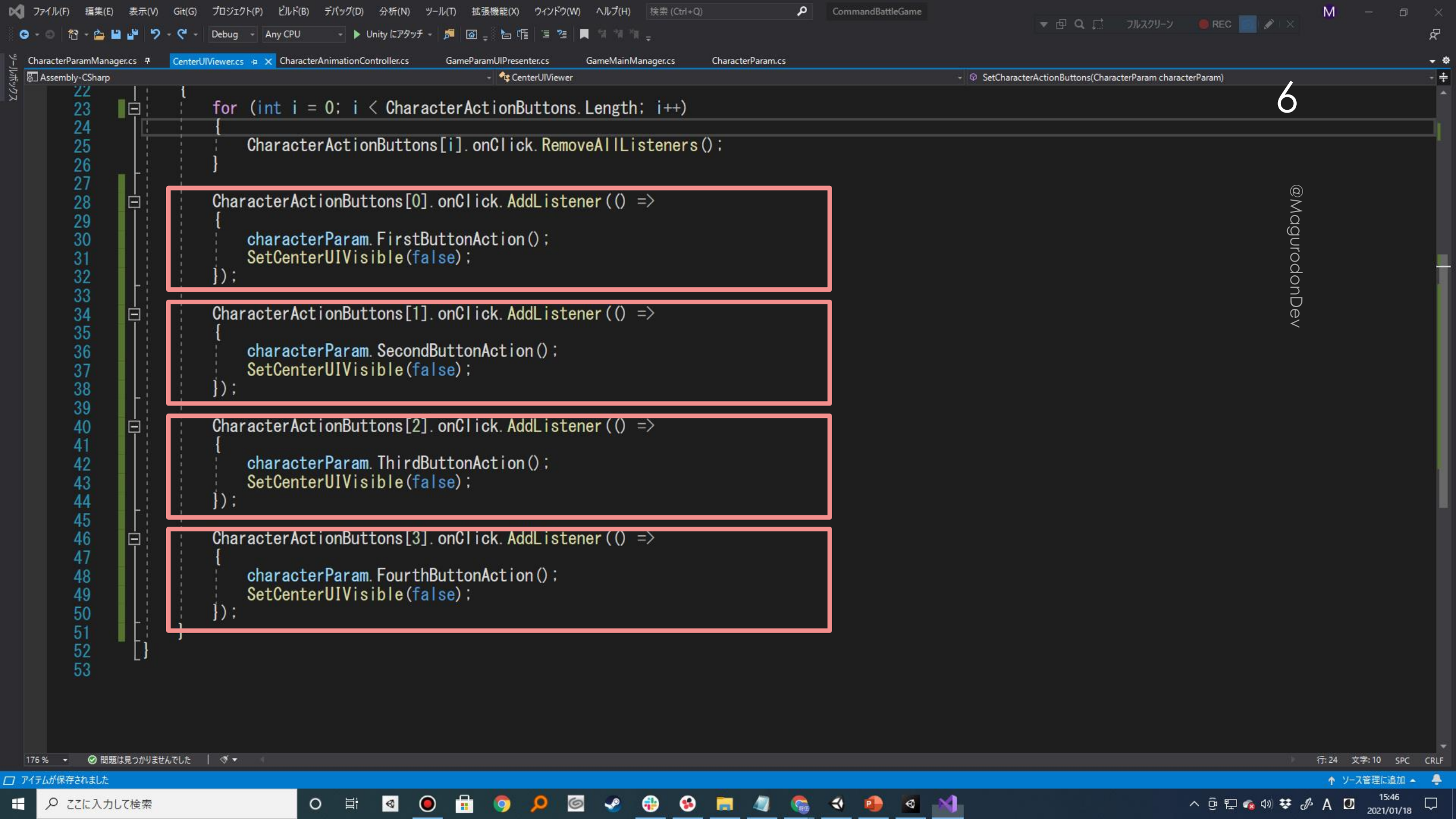
## コマンドバトルゲーム

5

- まずは一番早いスピードのキャラクターのボタンアクションを反映できるようにしました。
- 例えばSpellCasterのスピードをAttackerより上にしてみてください。
- SpellCasterの攻撃モーションが再生されたと思います。
- では次に、ボタンを押すと消えるようにしましょう。





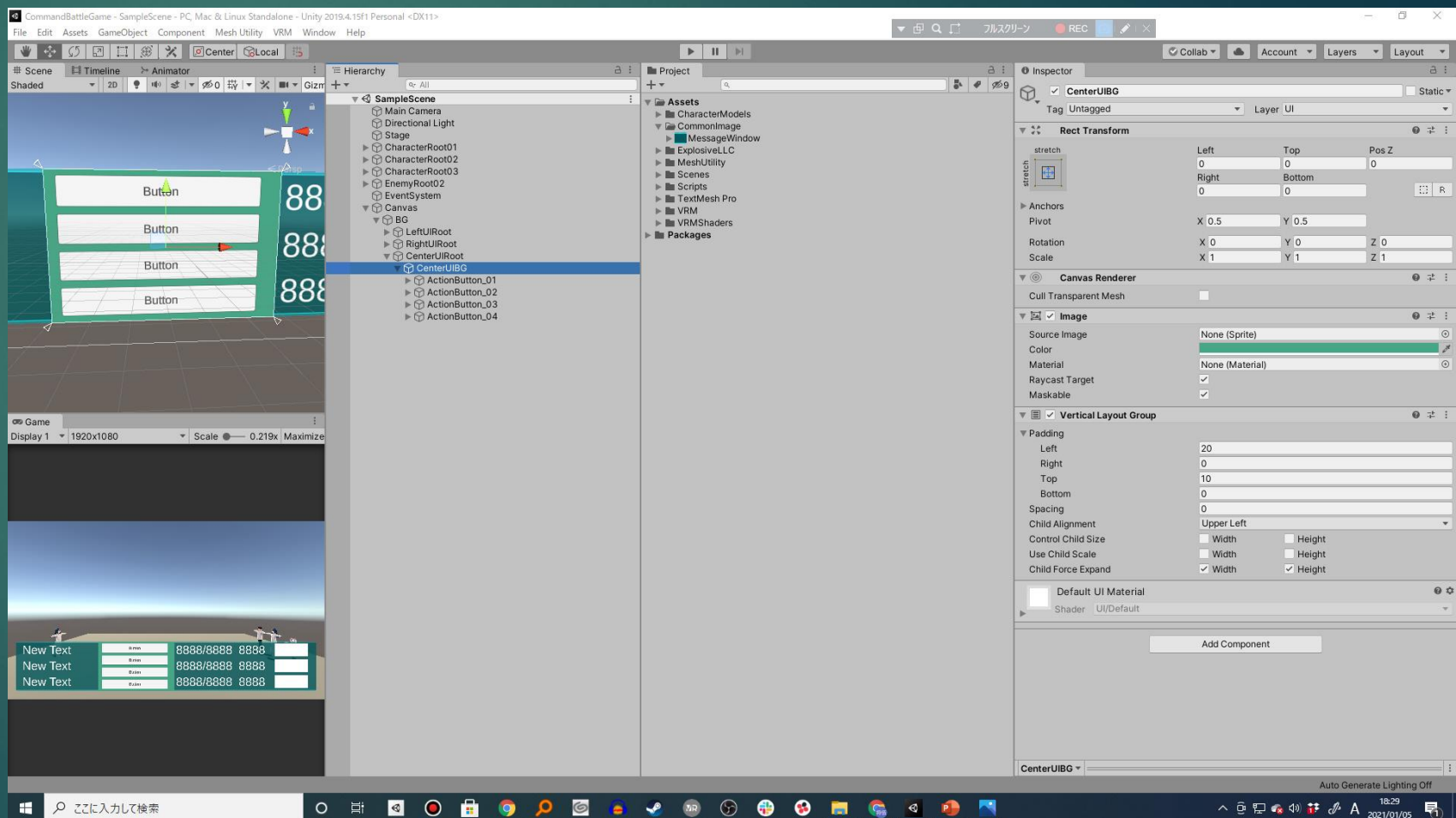


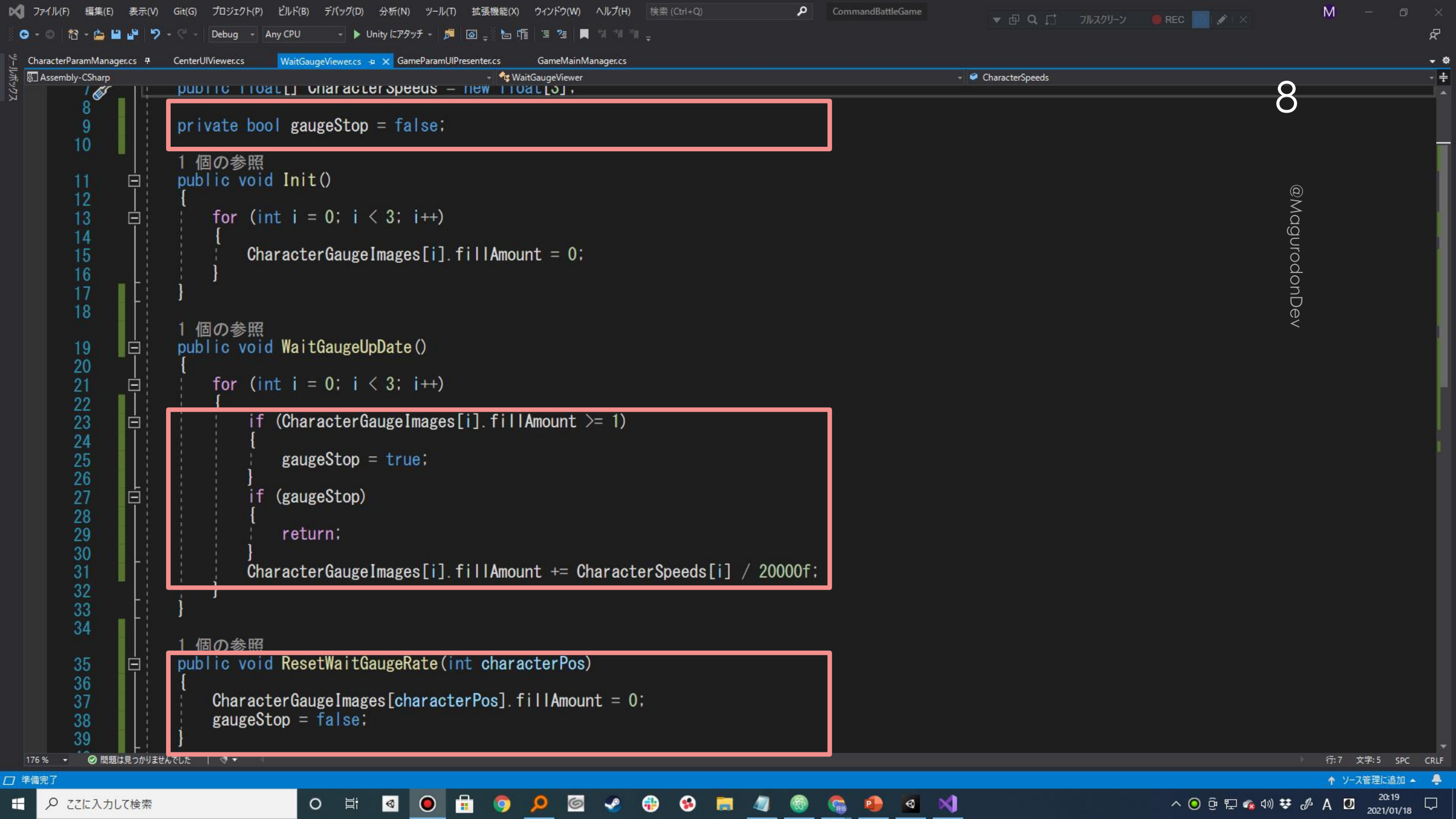
# Unity

## コマンドバトルゲーム

7

- これでボタンを押した時にウィンドウが消えるようになりました。
- それでは今から、すべてのキャラクターが順々にコマンドを実行できるようにしましょう。
- イメージとしては、
  - ①ゲージが埋まる
  - ②ほかのゲージは止まる
  - ③ボタンを押したらまた埋まり始める
- といった感じです。
- 編集するスクリプトは、WaitGaugeViewer.cs、CenterUIViewer.cs、GameMainManager.csです。





```
private bool gaugeStop = false;
```

```
1 個の参照  
public void Init()
```

```
{  
    for (int i = 0; i < 3; i++)  
    {  
        CharacterGaugeImages[i].fillAmount = 0;  
    }  
}
```

```
1 個の参照  
public void WaitGaugeUpdate()
```

```
{  
    for (int i = 0; i < 3; i++)  
    {  
        if (CharacterGaugeImages[i].fillAmount >= 1)  
        {  
            gaugeStop = true;  
            if (gaugeStop)  
            {  
                return;  
            }  
            CharacterGaugeImages[i].fillAmount += CharacterSpeeds[i] / 20000f;  
        }  
    }  
}
```

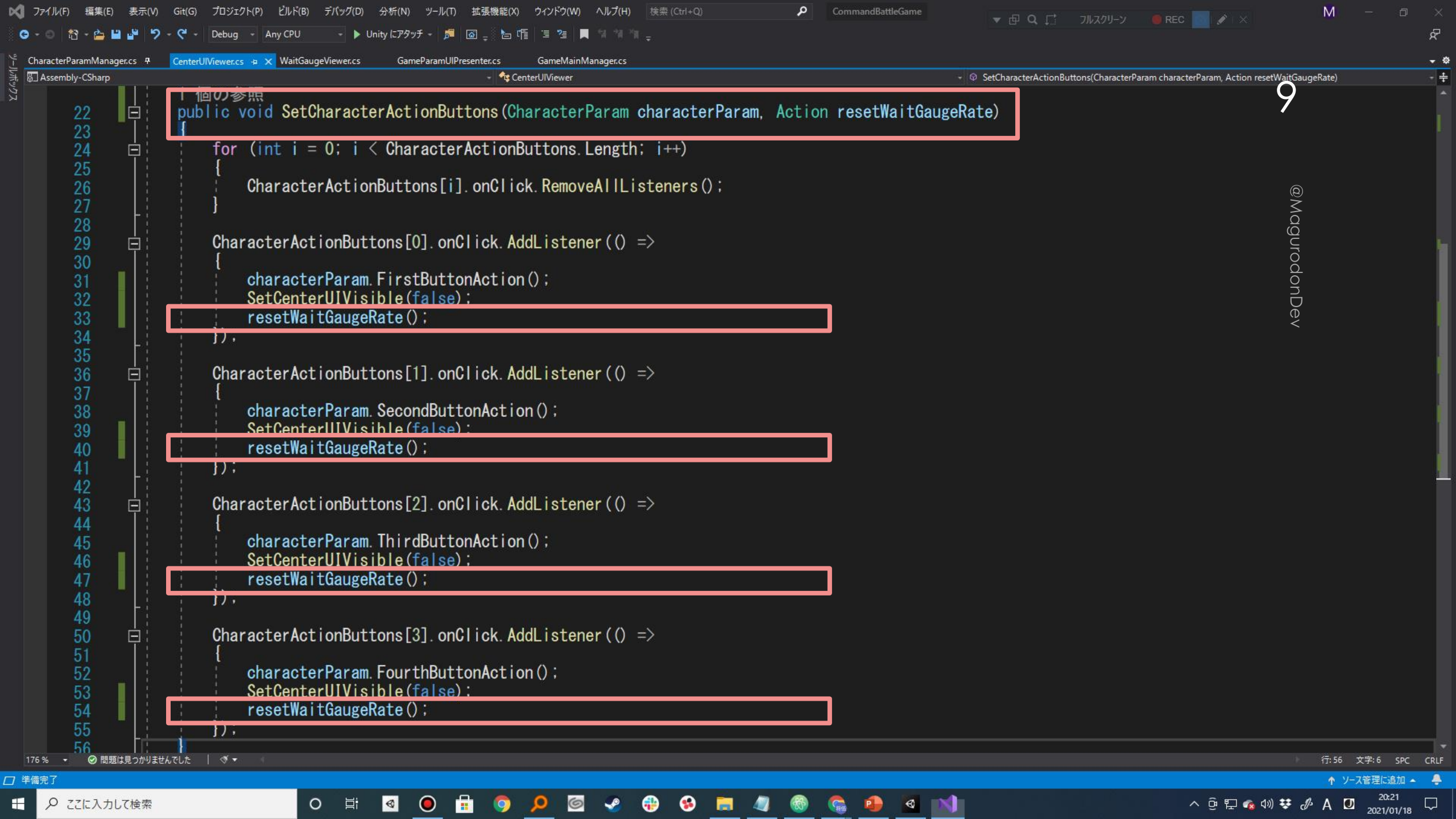
```
1 個の参照  
public void ResetWaitGaugeRate(int characterPos)
```

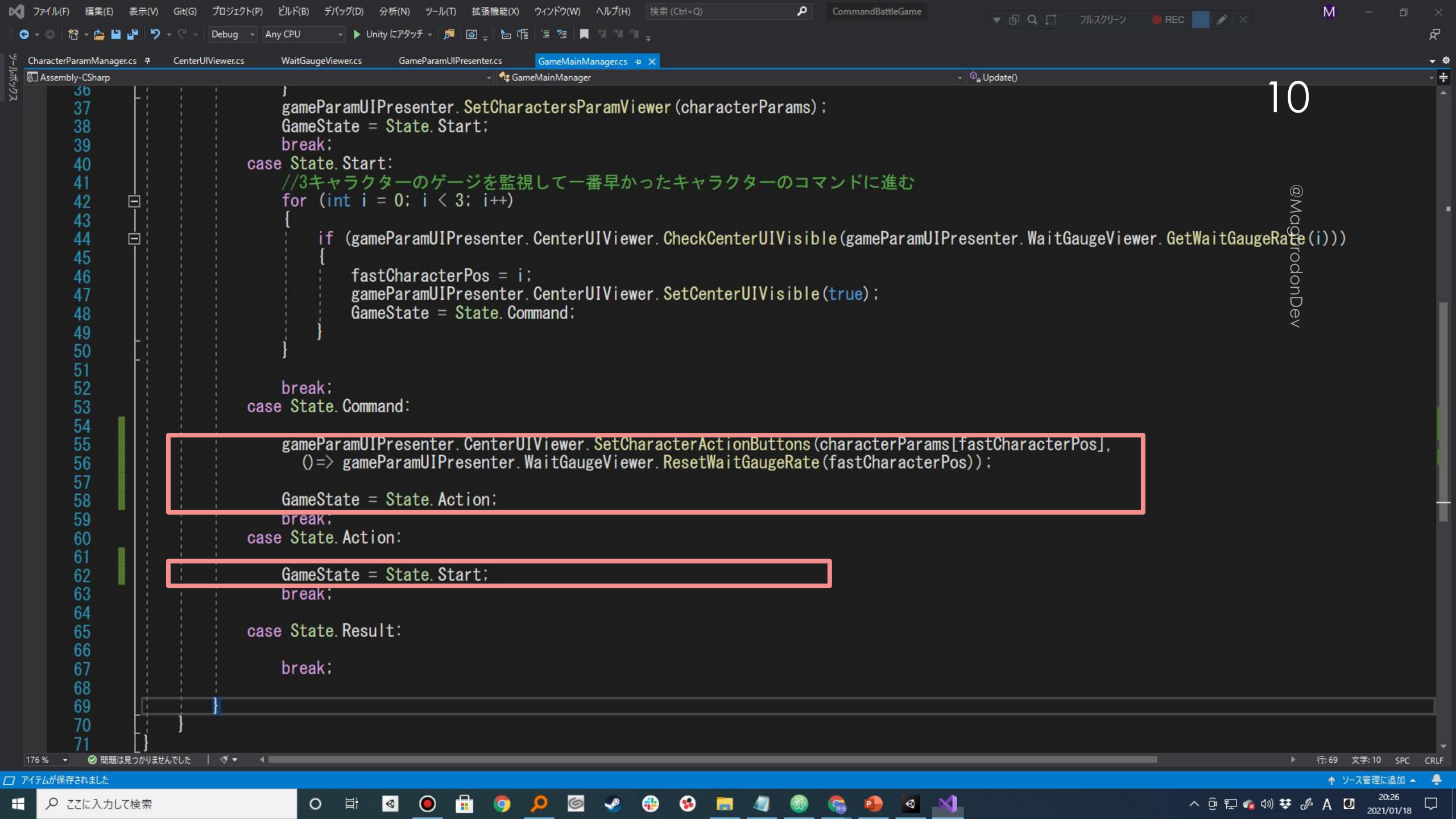
```
{  
    CharacterGaugeImages[characterPos].fillAmount = 0;  
    gaugeStop = false;  
}
```

8

@MagurodonDev





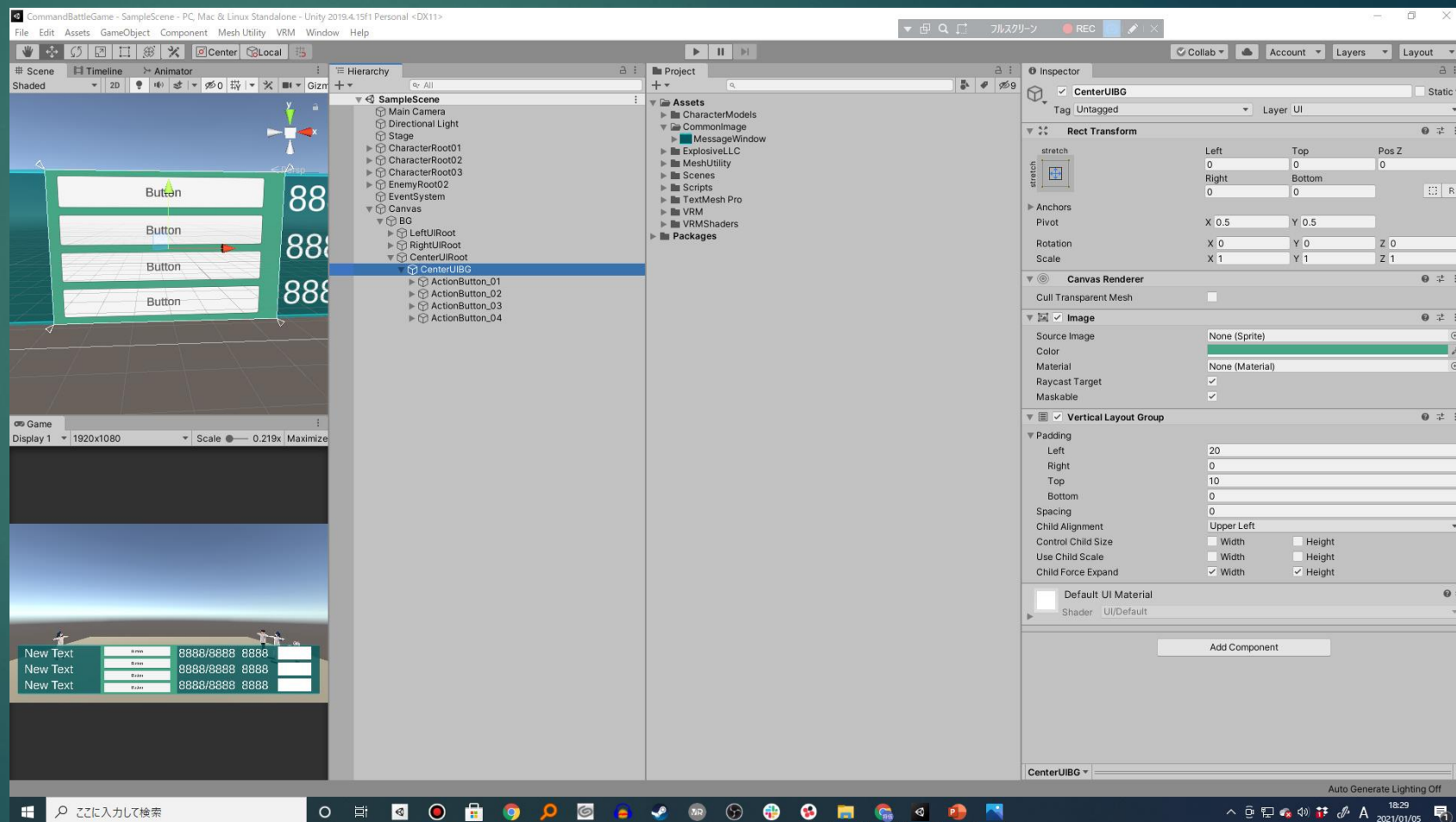


# Unity

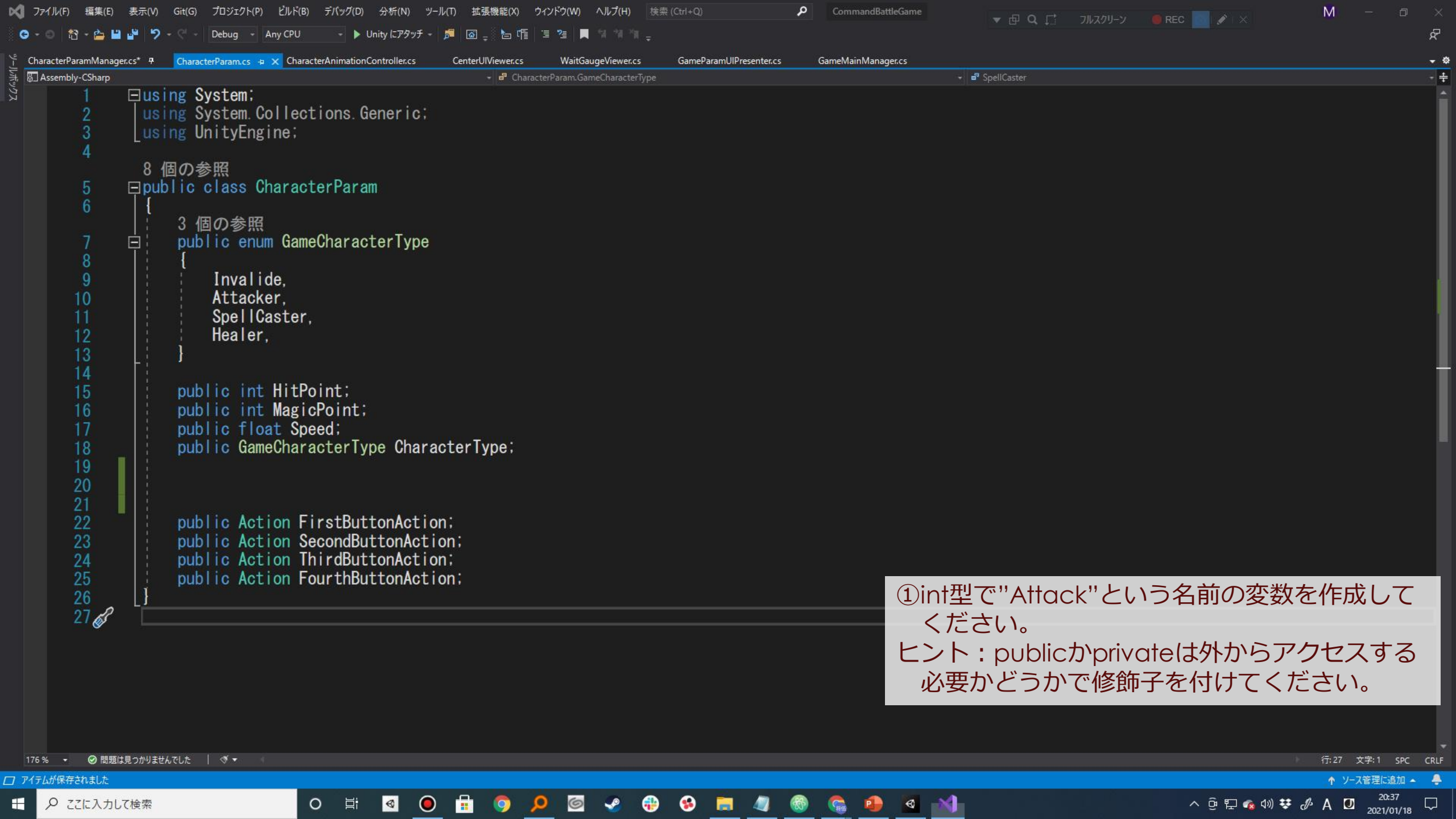
## コマンドバトルゲーム

11

- 再生してみると、ゲージが埋まったら他のキャラクターのゲージが止まりました。
- もちろん、ボタンを押してアクションさせるとまたゲージが埋まり始めます。
- 次はいよいよダメージの処理を書いていきましょう。
- キャラクターのパラメーターも追加していきましょうか。
- CharacterParamに攻撃力の変数を追加してください。
- 同じように、CharacterParamManagerで設定できるようにしましょう。



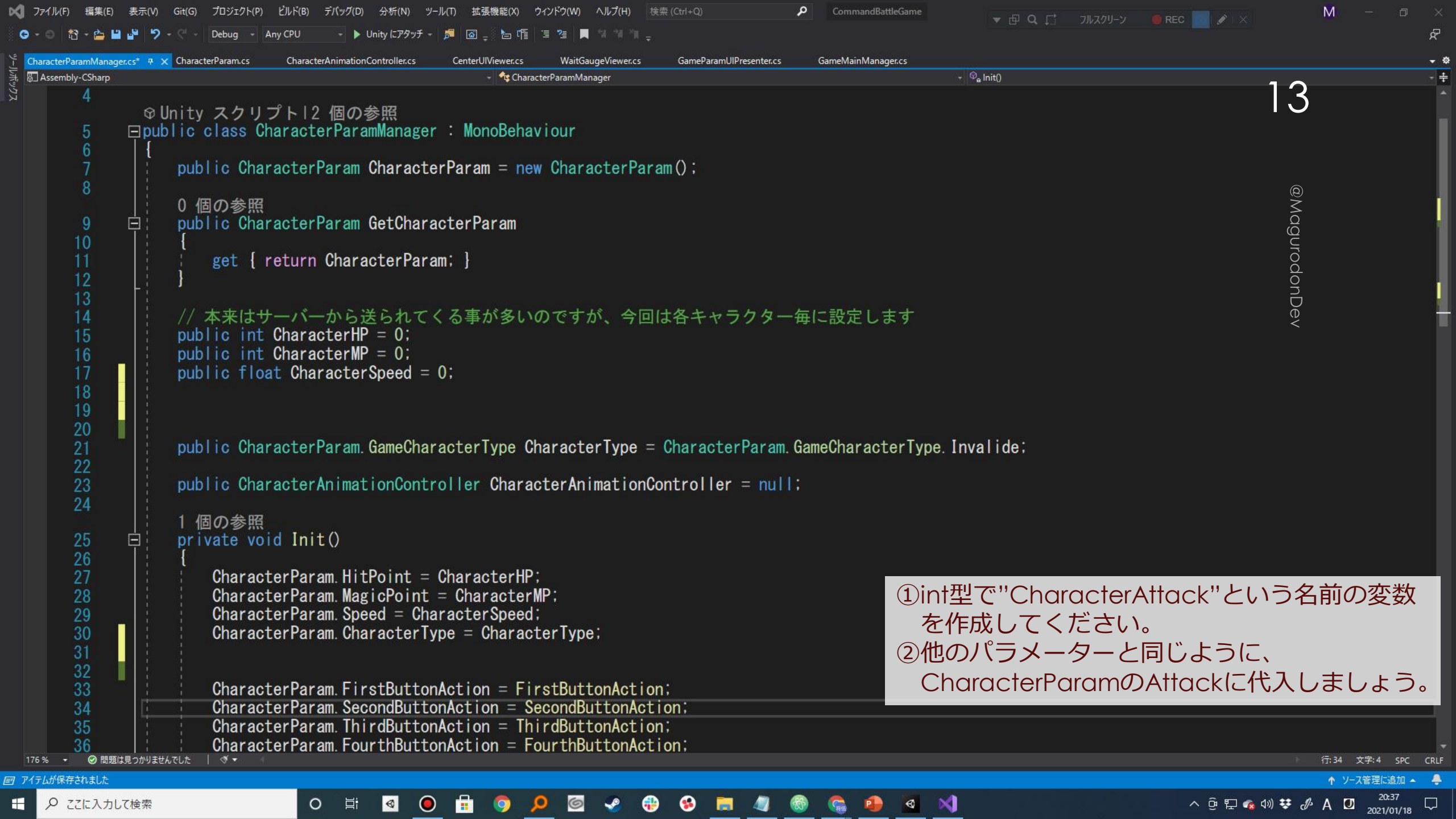




①int型で"Attack"という名前の変数を作成してください。

ヒント：publicかprivateは外からアクセスする必要かどうかで修飾子を付けてください。





13

@MagurodonDev

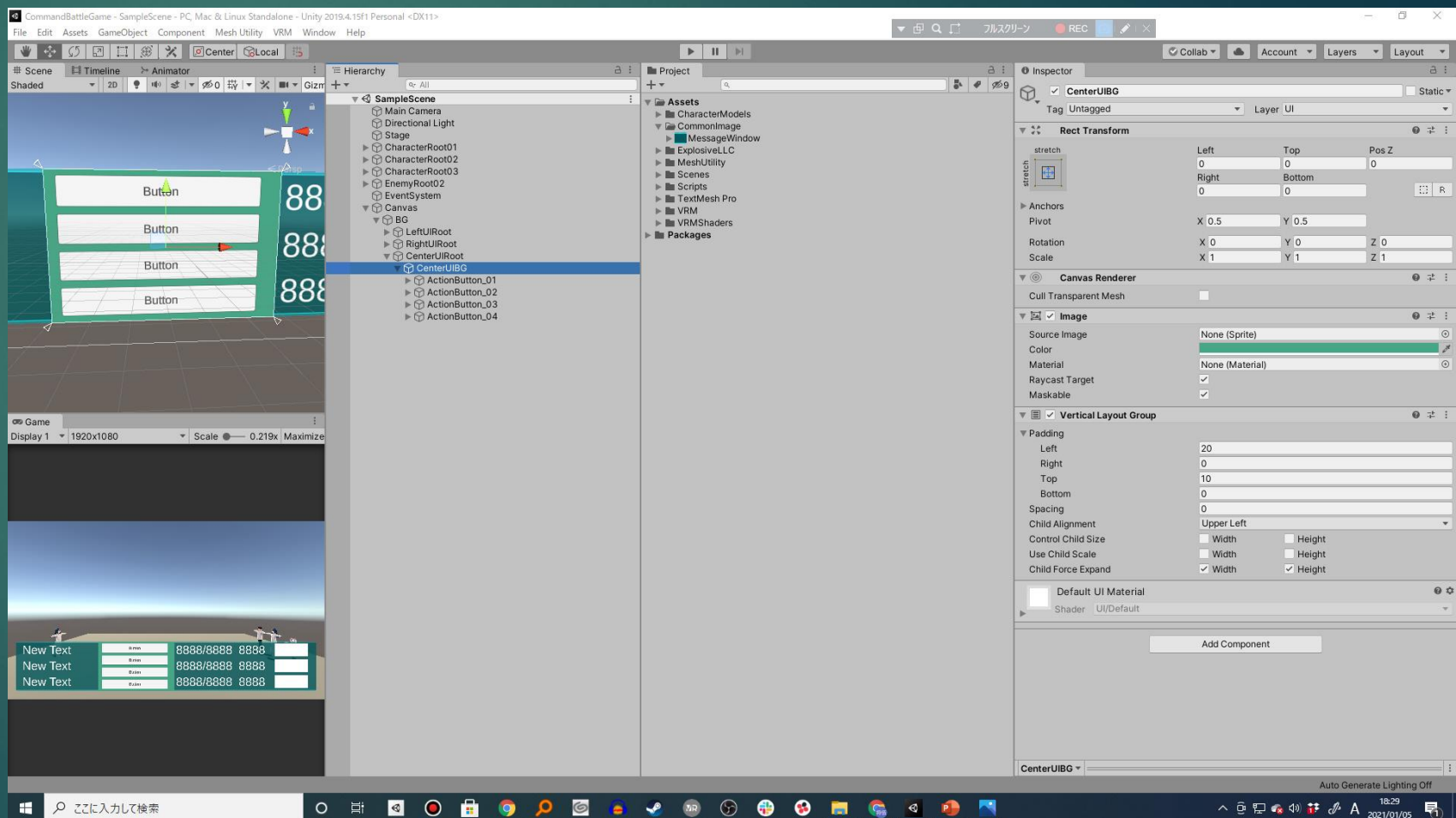
①int型で"CharacterAttack"という名前の変数  
を作成してください。  
②他のパラメーターと同じように、  
CharacterParamのAttackに代入しましょう。

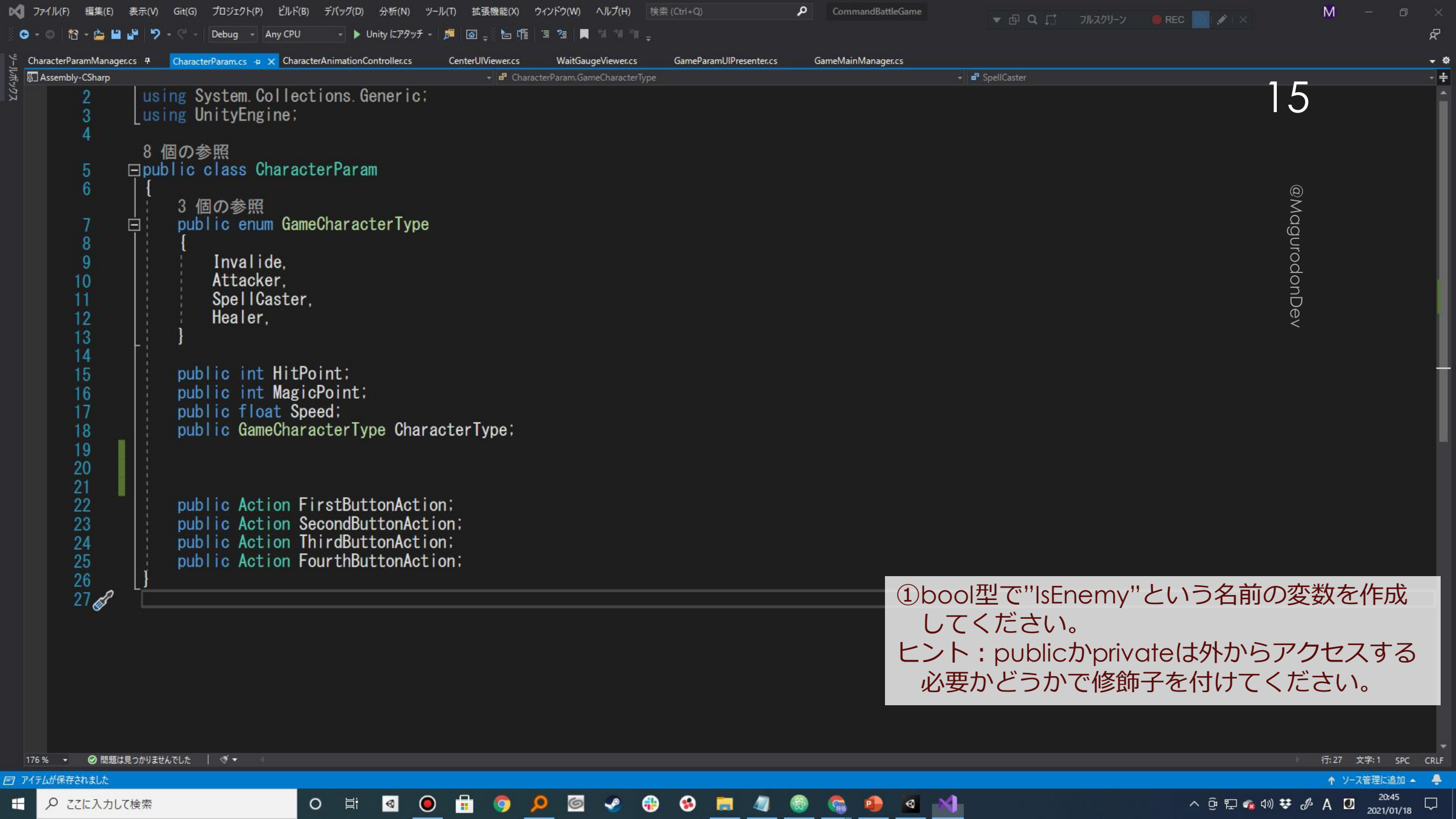
# Unity

## コマンドバトルゲーム

14

- Unity上のAttackerのCharacterParamManagerのAttackの値を100に設定します。
- ここまでできたら敵か味方かの判定のフラグをCharacterParamに設定しましょう。
- 先ほどと同じように、CharacterParamとCharacterParamManagerにフラグの変数を追加します。



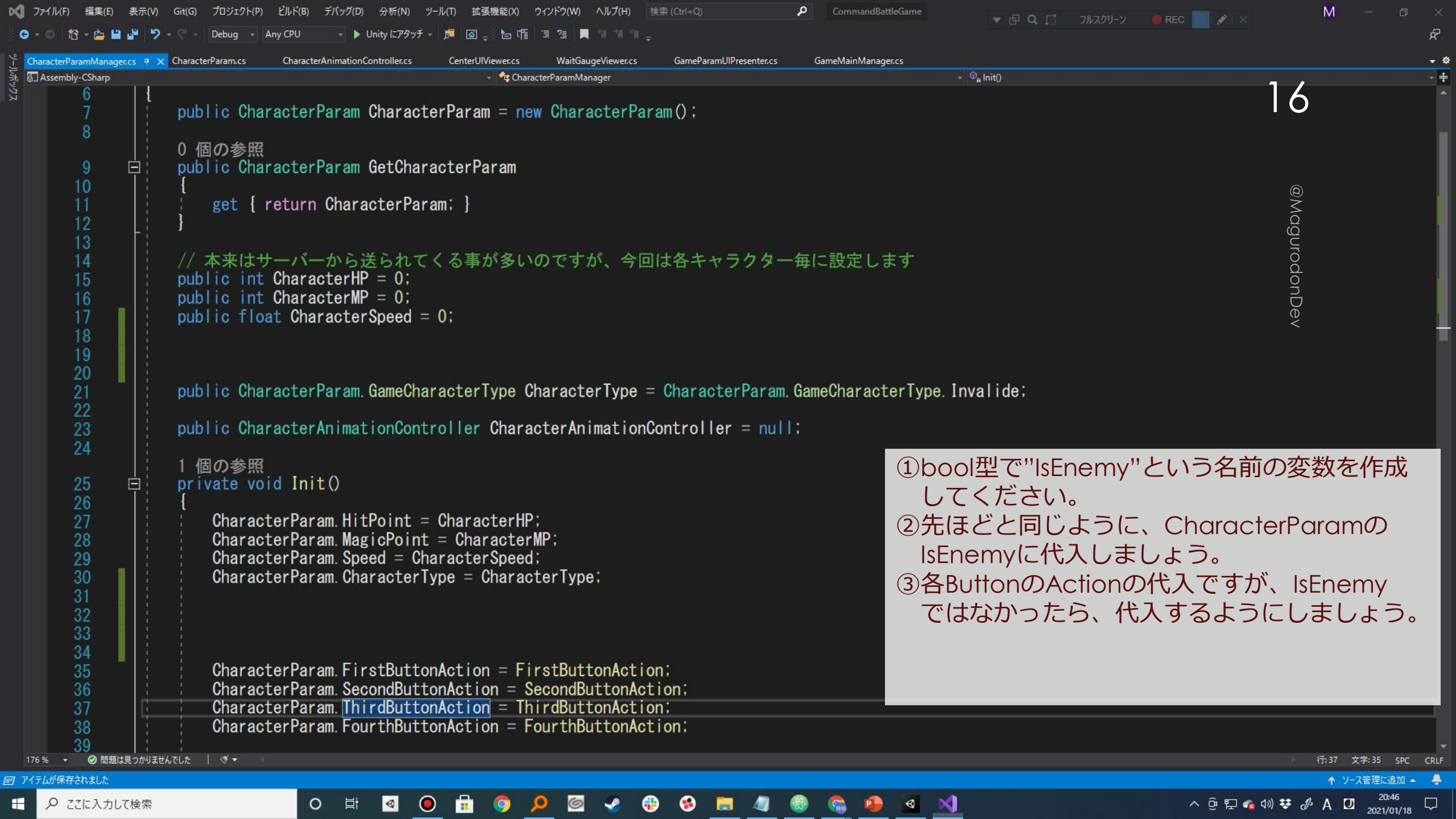


15

@MagurodonDev

①bool型で"IsEnemy"という名前の変数を作成してください。  
ヒント：publicかprivateは外からアクセスする必要かどうかで修飾子を付けてください。





```
6 {  
7     public CharacterParam CharacterParam = new CharacterParam();  
8 }
```

0 個の参照

```
9 public CharacterParam GetCharacterParam  
10 {  
11     get { return CharacterParam; }  
12 }
```

// 本来はサーバーから送られてくる事が多いのですが、今回は各キャラクター毎に設定します

```
14 public int CharacterHP = 0;  
15 public int CharacterMP = 0;  
16 public float CharacterSpeed = 0;
```

```
21 public CharacterParam.GameCharacterType CharacterType = CharacterParam.GameCharacterType.Invalid;  
22
```

```
23 public CharacterAnimationController CharacterAnimationController = null;  
24
```

1 個の参照

```
25 private void Init()  
26 {
```

```
27     CharacterParam.HitPoint = CharacterHP;  
28     CharacterParam.MagicPoint = CharacterMP;  
29     CharacterParam.Speed = CharacterSpeed;  
30     CharacterParam.CharacterType = CharacterType;  
31
```

```
35     CharacterParam.FirstButtonAction = FirstButtonAction;  
36     CharacterParam.SecondButtonAction = SecondButtonAction;  
37     CharacterParam.ThirdButtonAction = ThirdButtonAction;  
38     CharacterParam.FourthButtonAction = FourthButtonAction;  
39 }
```

- ①bool型で"IsEnemy"という名前の変数を作成してください。
- ②先ほどと同じように、CharacterParamのIsEnemyに代入しましょう。
- ③各ButtonのActionの代入ですが、IsEnemyではなかったら、代入するようにしましょう。

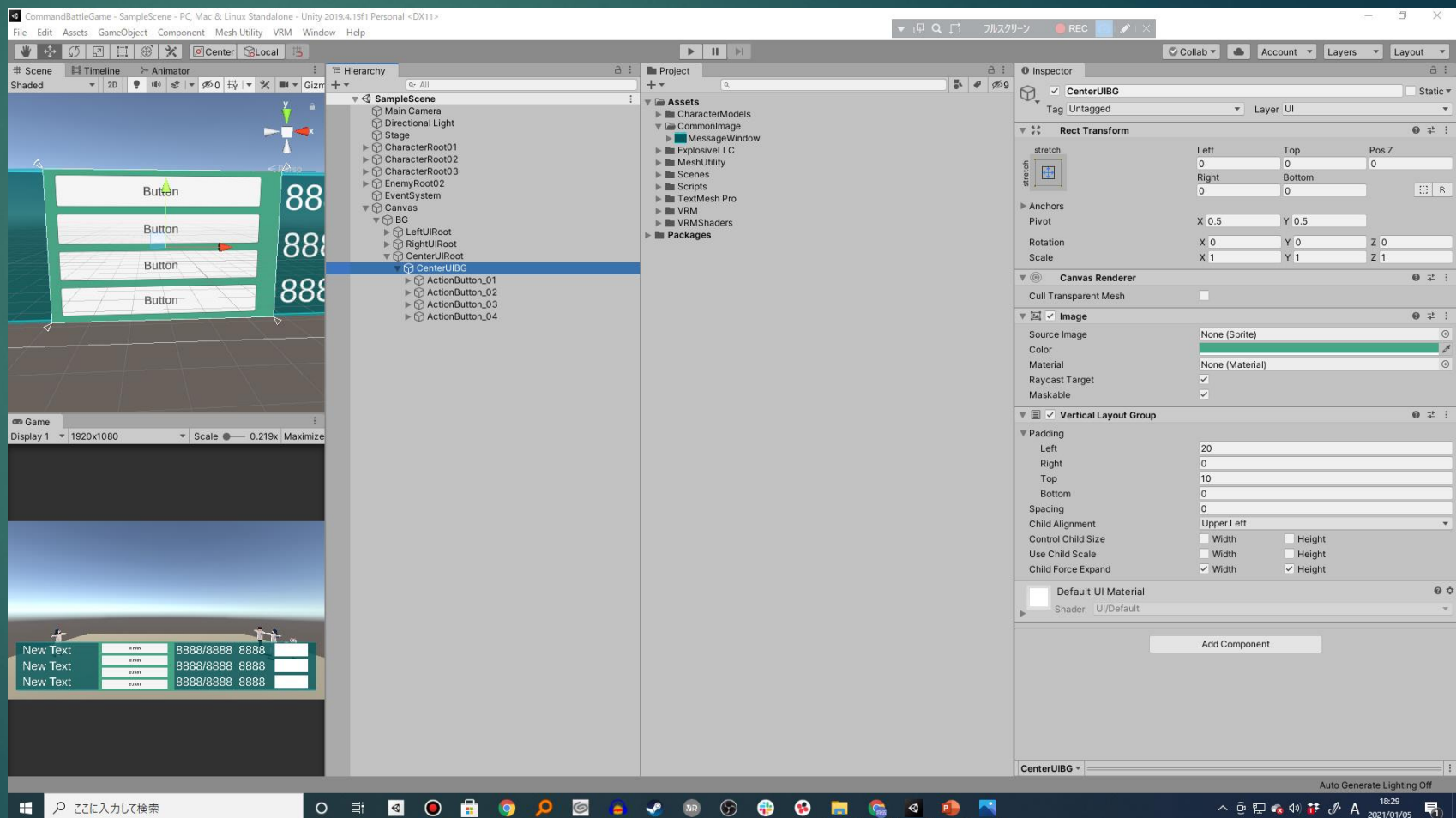


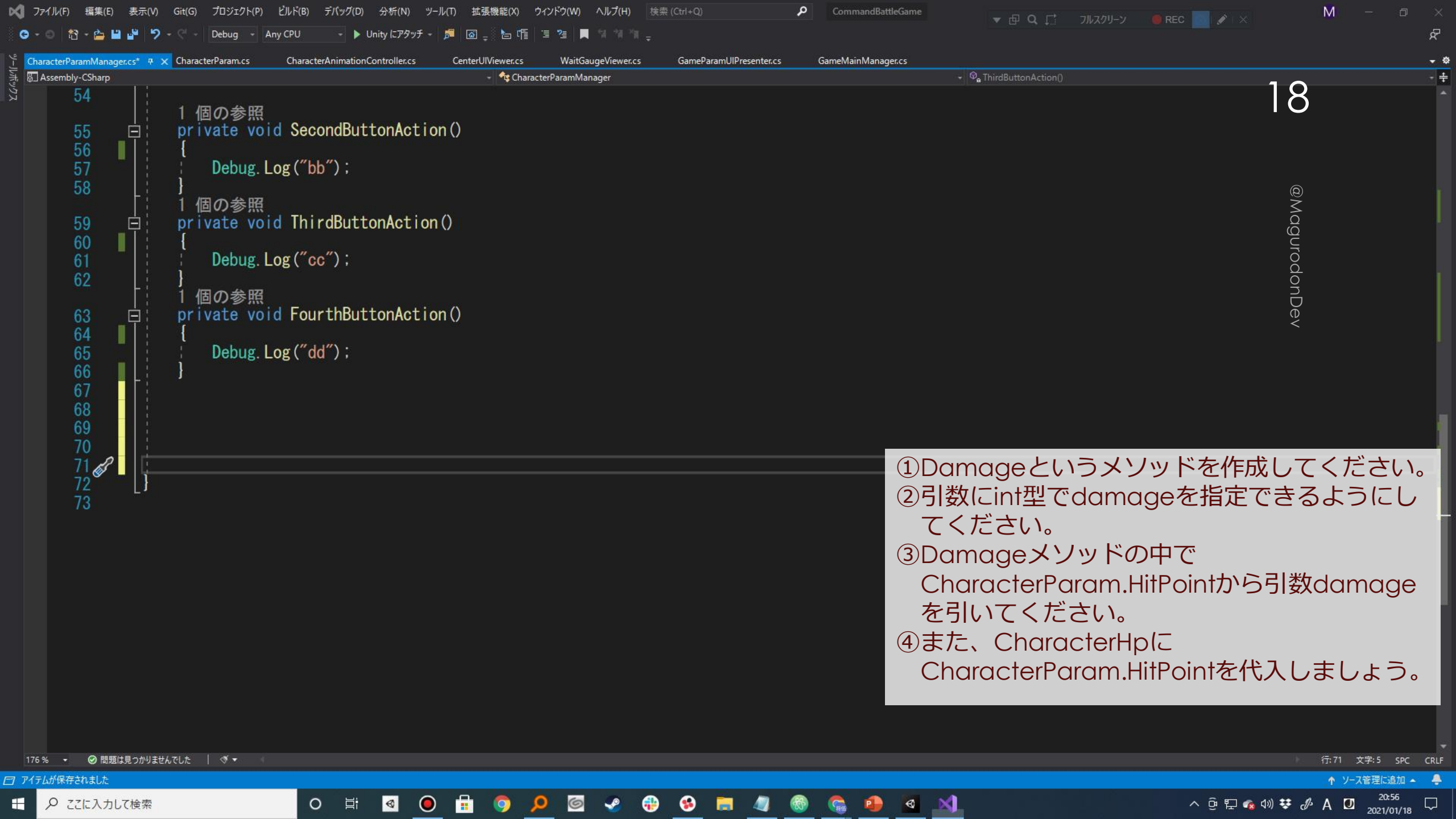
# Unity

## コマンドバトルゲーム

17

- Unity上で敵に  
CharacterParamManagerを  
Addし、HpやAttack等を設定し、  
IsEnemyフラグに☑をいれてお  
きます。
- さて、ダメージを与えるという  
処理はどう書けばよいでしょう  
か？
- ちょっと考えて書いてみましょ  
う。





```
54 1 個の参照
55 private void SecondButtonAction()
56 {
57     Debug.Log("bb");
58 }
59 1 個の参照
60 private void ThirdButtonAction()
61 {
62     Debug.Log("cc");
63 }
64 1 個の参照
65 private void FourthButtonAction()
66 {
67     Debug.Log("dd");
68 }
69
70
71
72
73
```

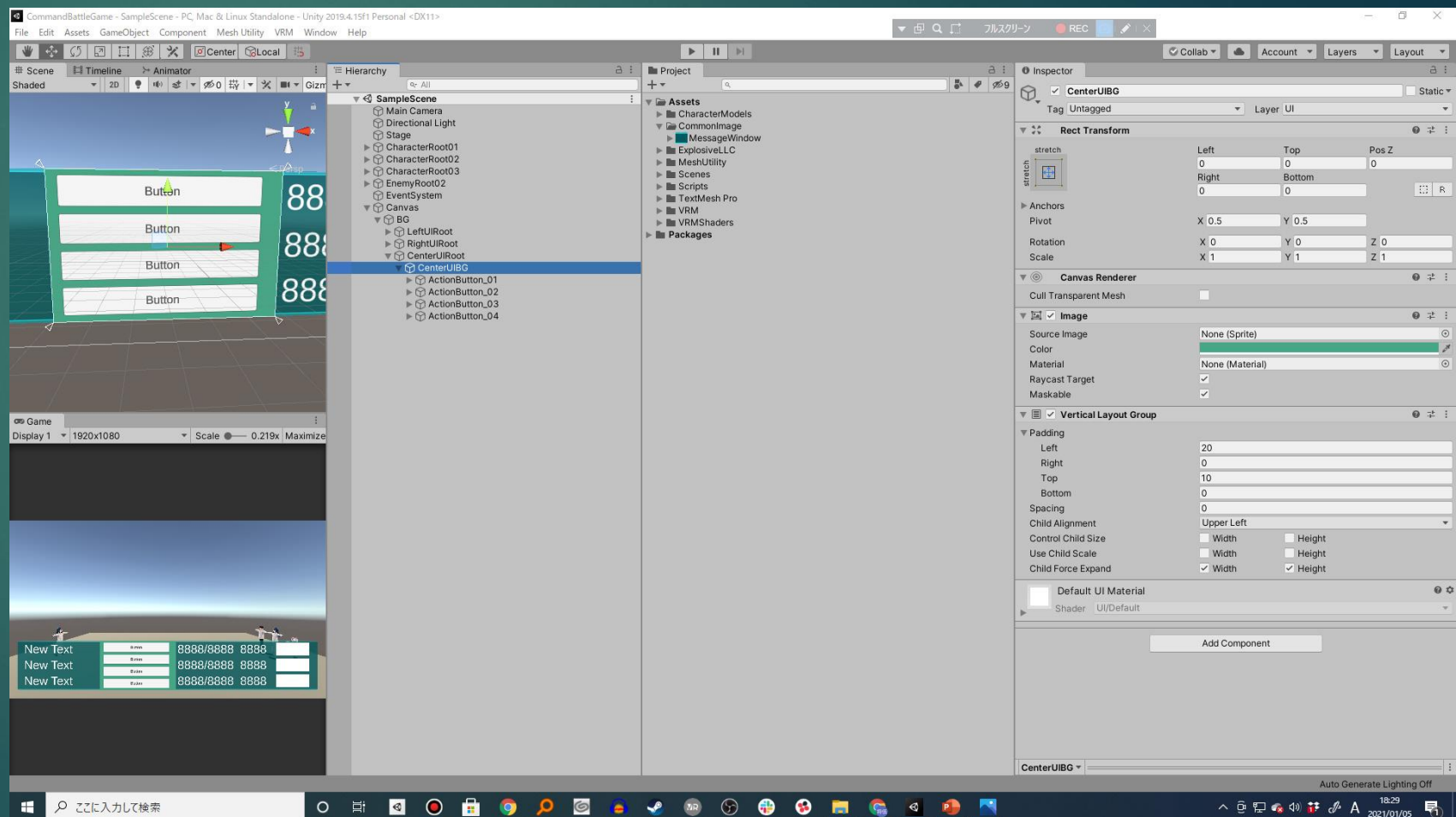
- ①Damageというメソッドを作成してください。
- ②引数にint型でdamageを指定できるようにしてください。
- ③Damageメソッドの中で  
CharacterParam.HitPointから引数damageを引いてください。
- ④また、CharacterHpに  
CharacterParam.HitPointを代入しましょう。

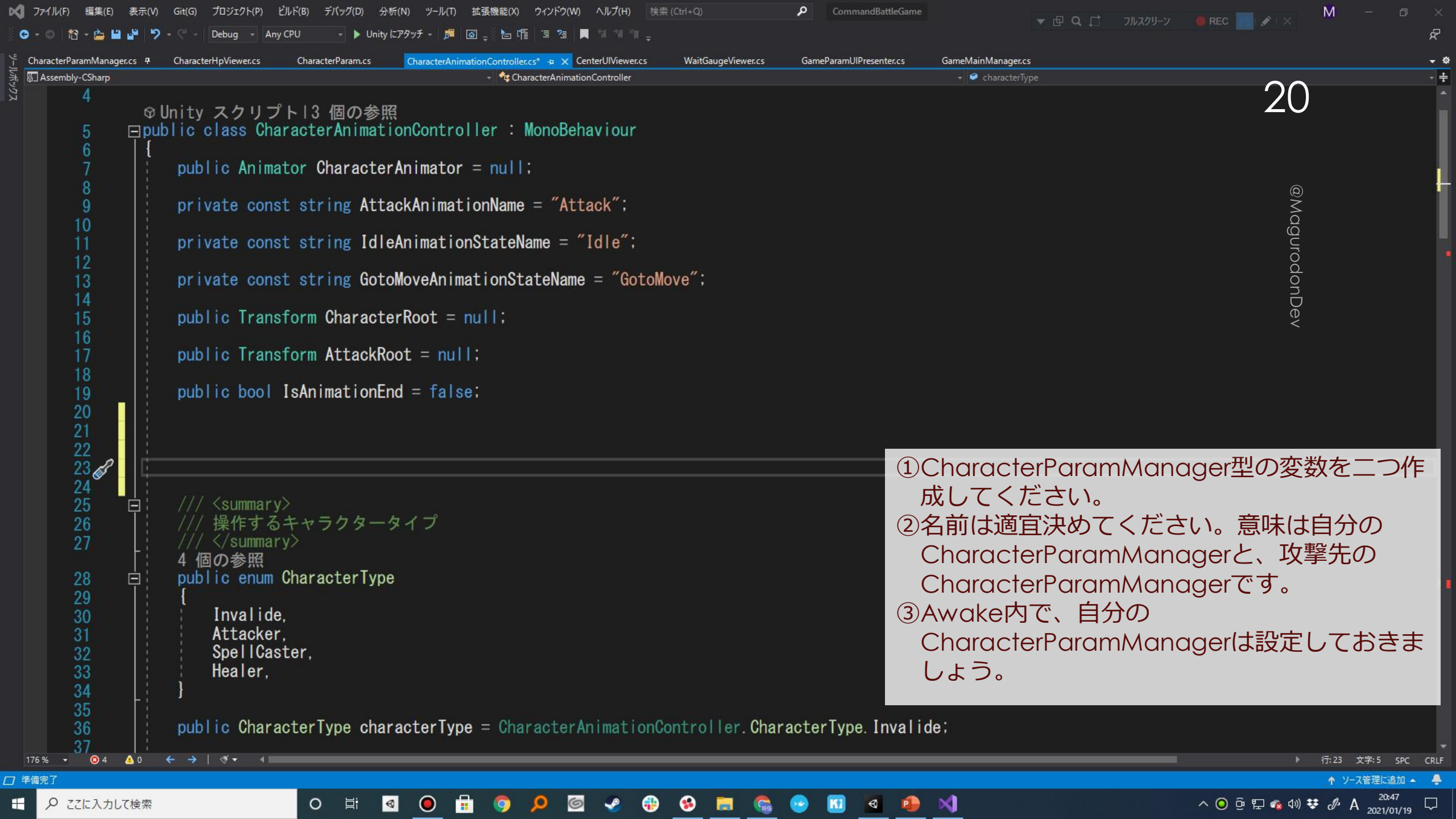
# Unity

19

## コマンドバトルゲーム

- さて、これでDamageメソッドが作れましたね。
- あとは自分のCharacterParamのAttackをのDamageメソッドの引数にして、AttackRootの同階層にある攻撃先のCharacterparamManagerに渡してあげればいいですね。
- ここで、プレイしてからエラーになり続けているHit()というAnimationEventのお話をします。
- まず、CharacterAnimationController.csを変更していきましょう。





20

@MagurodonDev

- ①CharacterParamManager型の変数を二つ作成してください。
- ②名前は適宜決めてください。意味は自分のCharacterParamManagerと、攻撃先のCharacterParamManagerです。
- ③Awake内で、自分のCharacterParamManagerは設定しておきましょう。

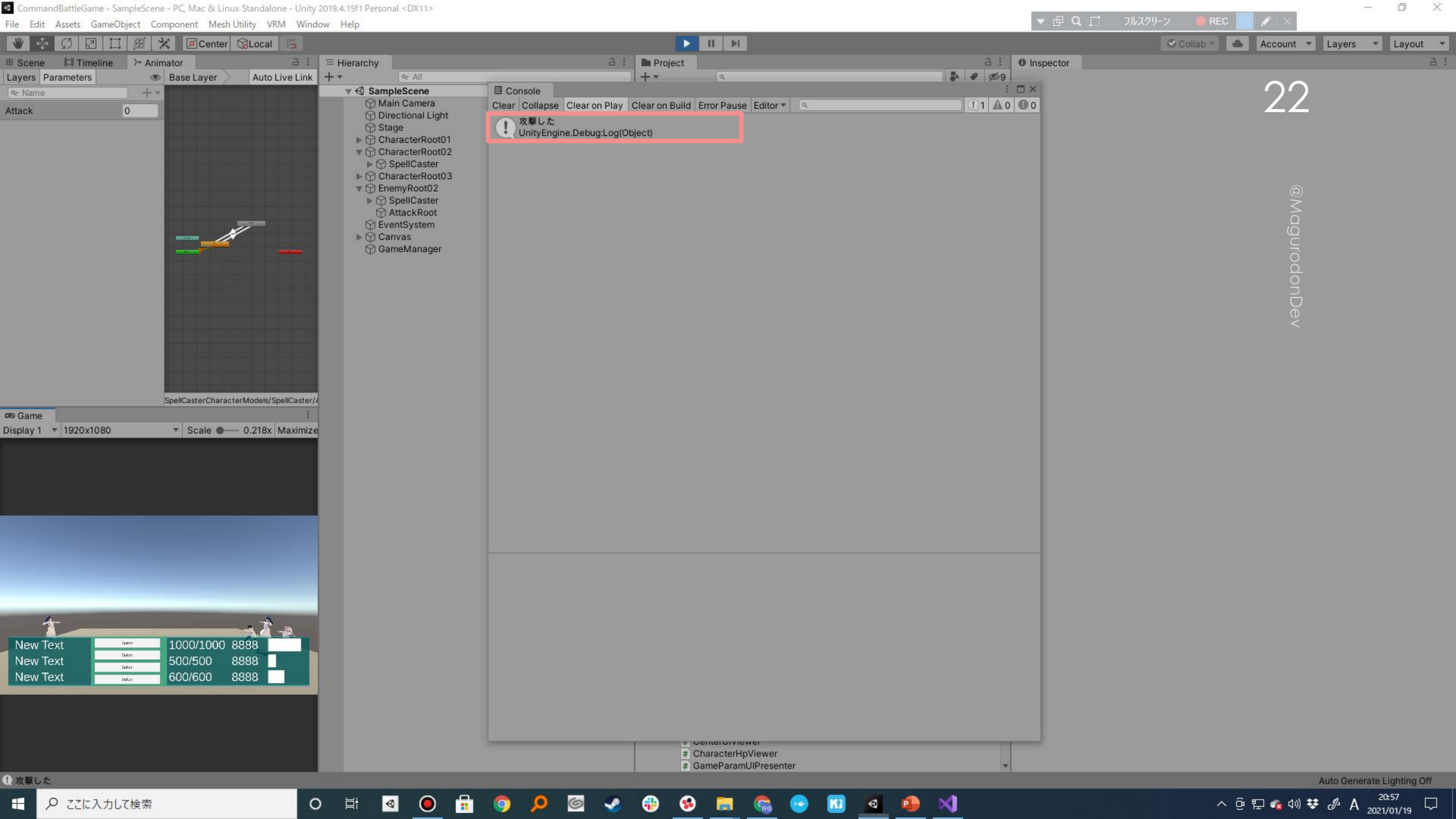


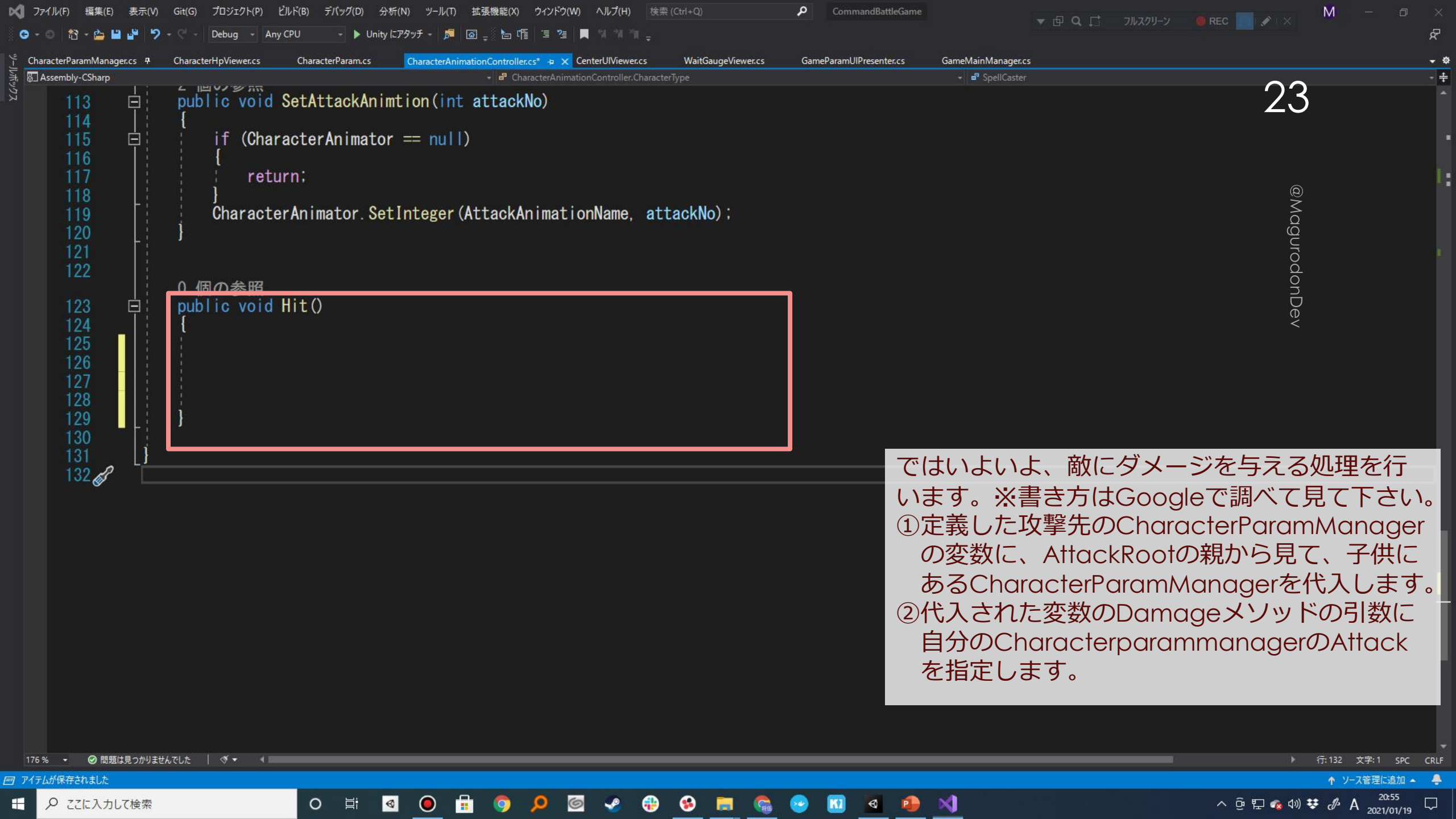
21

@MagurodonDev

```
113 public void SetAttackAnimtion(int attackNo)
114 {
115     if (CharacterAnimator == null)
116     {
117         return;
118     }
119     CharacterAnimator.SetInteger(AttackAnimationName, attackNo);
120 }
121
122
123 public void Hit()
124 {
125 }
126
127
128
129
130
131
132
```

- ①Hit()というメソッドを作成してください。
- ②その中にDebug.Log()で攻撃したという文字を表示させましょう。
- ③Unityに戻って、プレイしてみると、攻撃するキャラクターが攻撃したときにDebug.Log内のメッセージが表示されると思います。

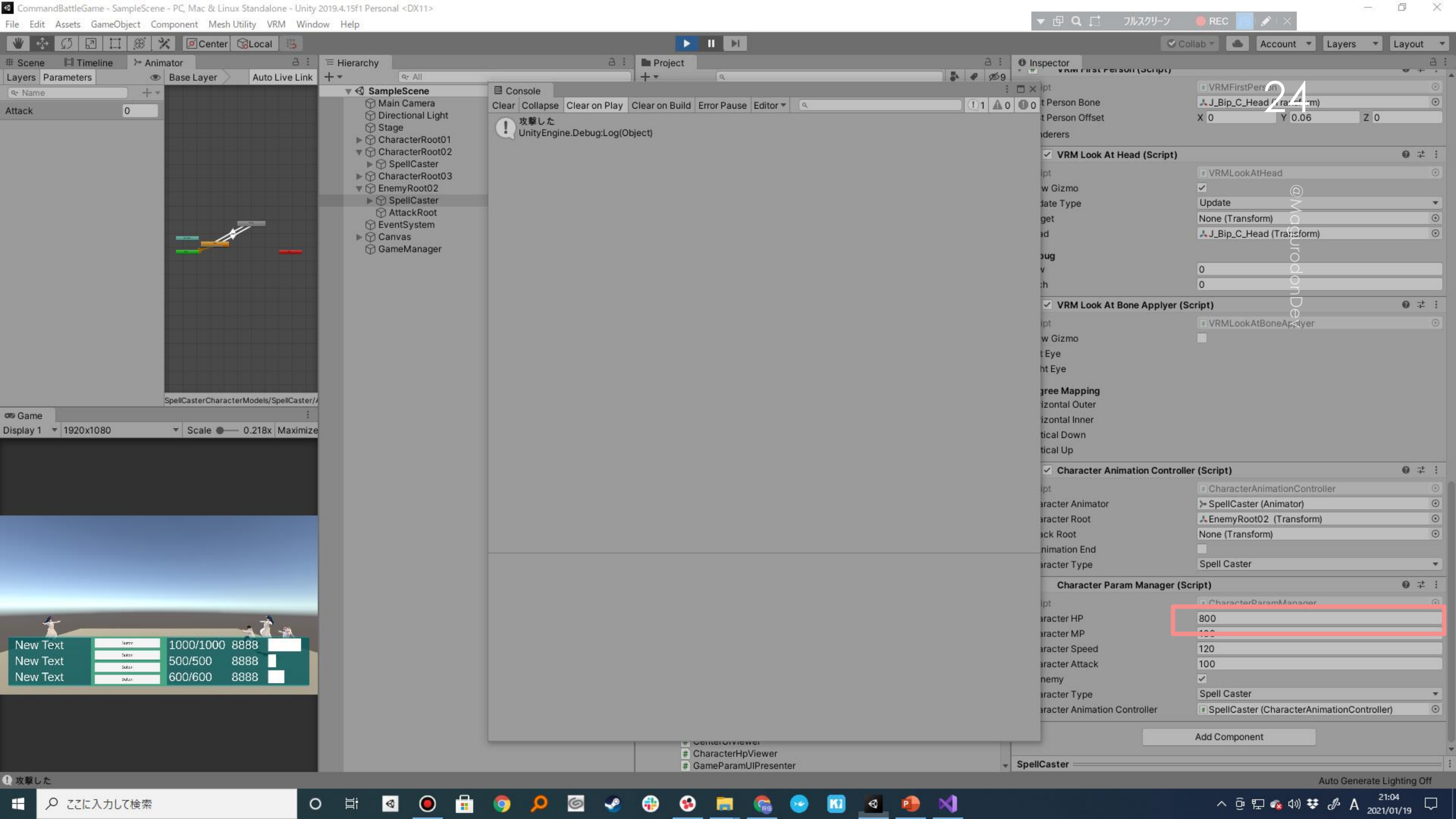




23

@MagurodonDev

ではいいよ、敵にダメージを与える処理を行います。※書き方はGoogleで調べて見て下さい。  
①定義した攻撃先のCharacterParamManagerの変数に、AttackRootの親から見て、子供にあるCharacterParamManagerを代入します。  
②代入された変数のDamageメソッドの引数に自分のCharacterparammanagerのAttackを指定します。



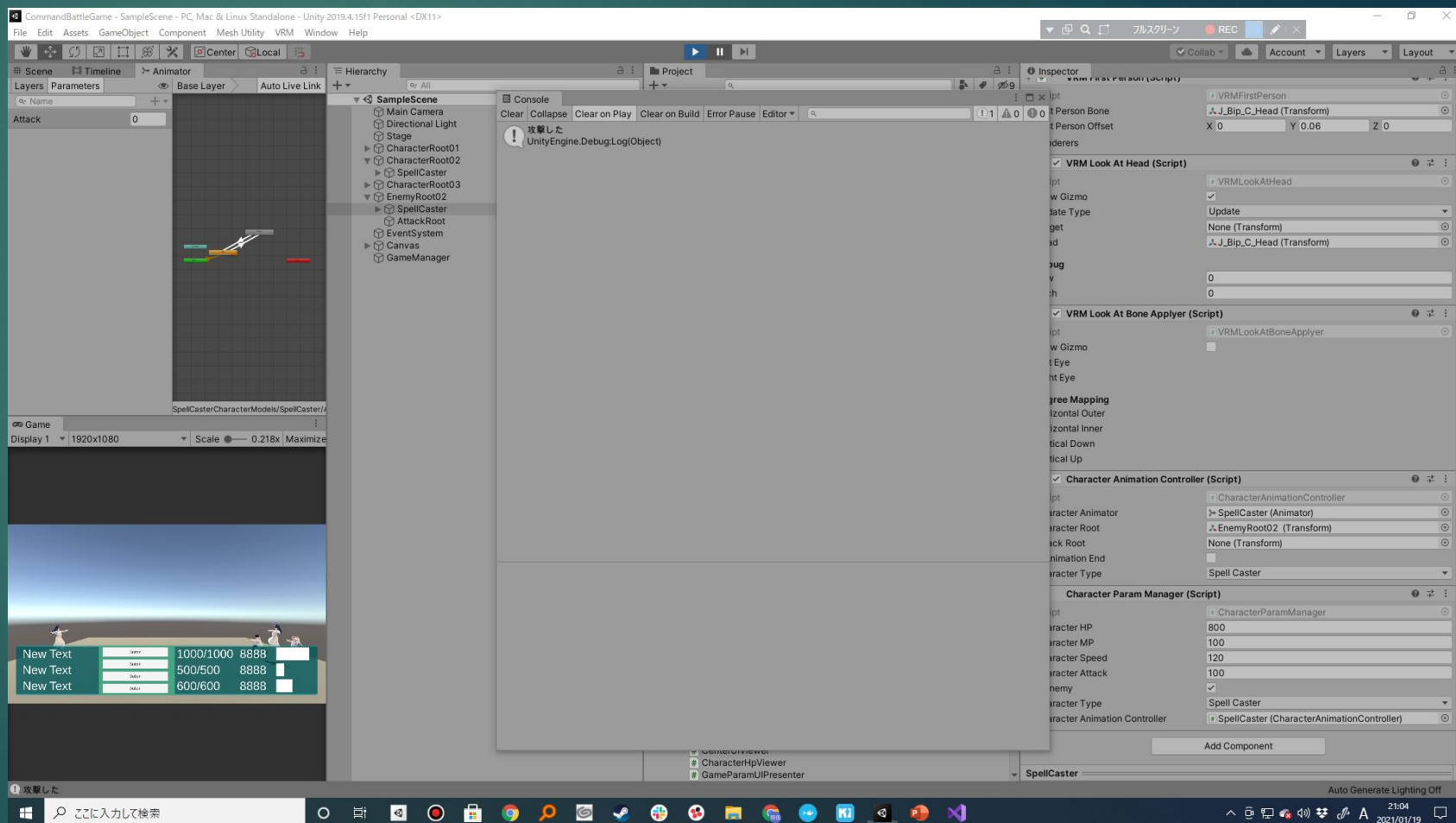


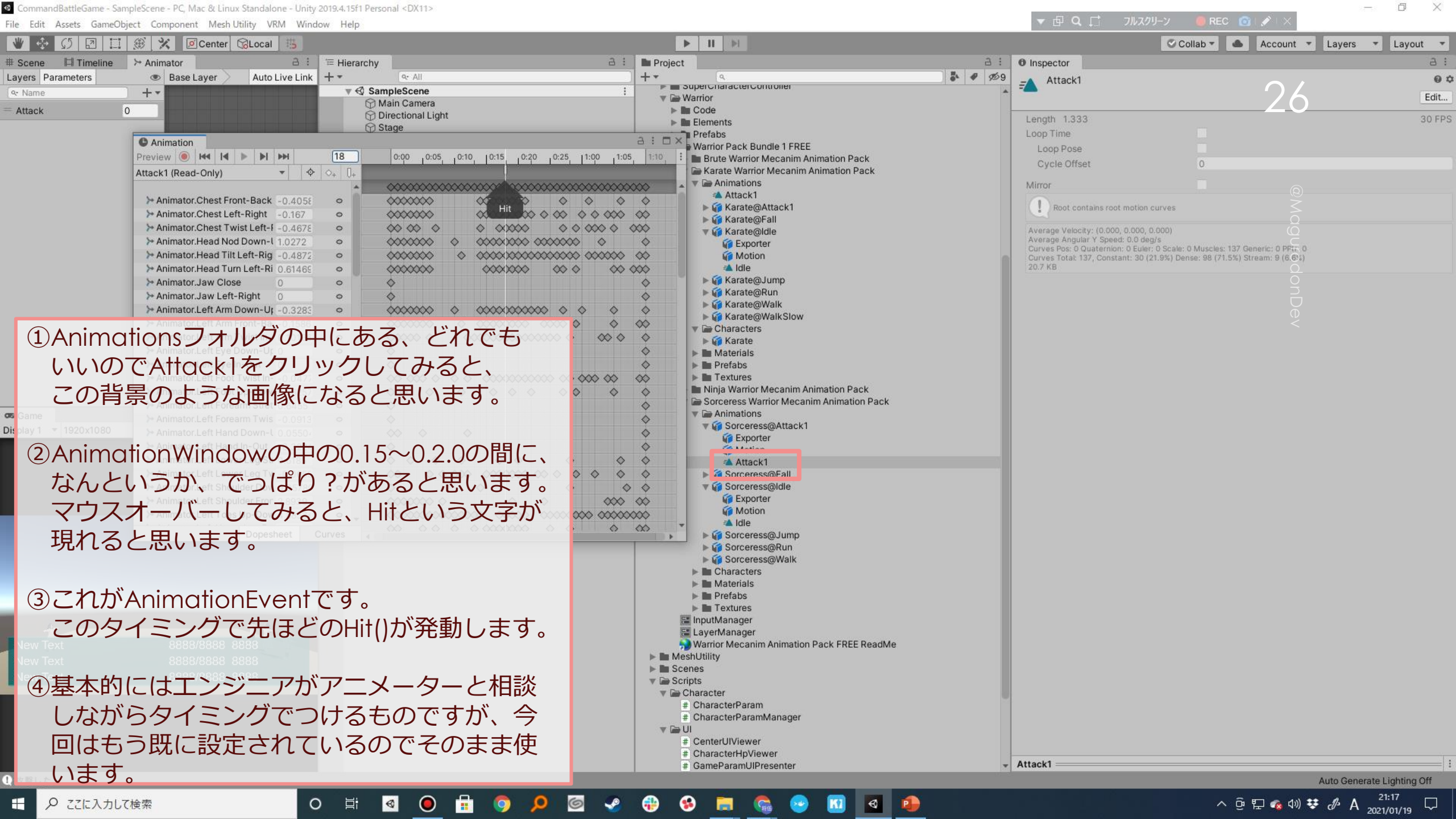
# Unity

25

## コマンドバトルゲーム

- これでDamageの処理を攻撃相手に渡せるようになりました。
- ここで、先ほどのHit()とは何かを説明しようと思います。
- Animationにはキーを打つことができます。
- UnityのWindowからAnimationをクリックし、でてきたタブからAnimationをクリックします。
- 次のページを見て下さい。





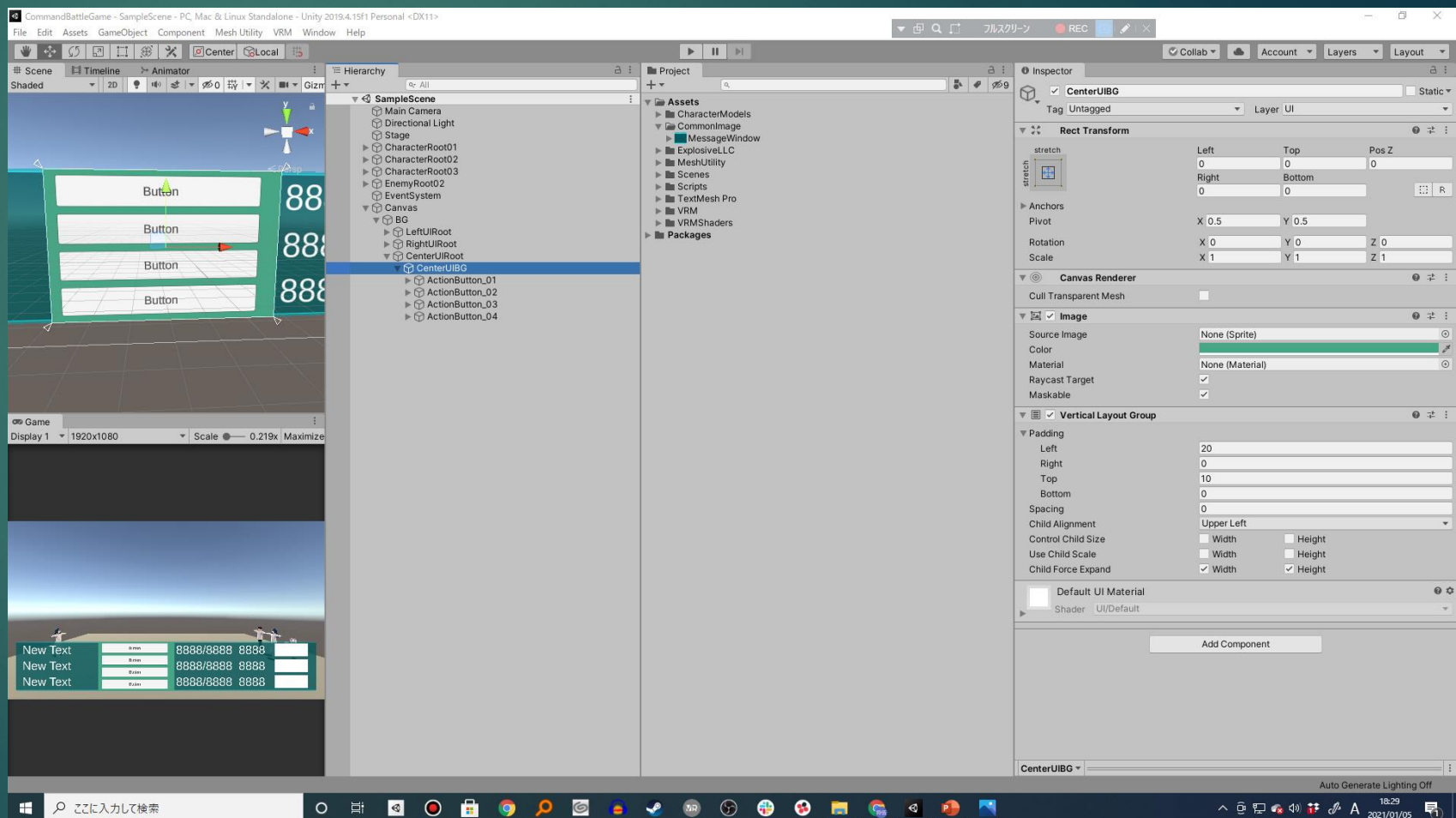
- ① Animationsフォルダの中にある、どれでもいいのでAttack1をクリックしてみると、この背景のような画像になると思います。
- ② AnimationWindowの中の0.15~0.2.0の間に、なんとなく でっぱり？があると思います。マウスオーバーしてみると、Hitという文字が現れると思います。
- ③ これがAnimationEventです。  
このタイミングで先ほどのHit()が発動します。
- ④ 基本的にはエンジニアがアニメーターと相談しながらタイミングでつけるものですが、今回はもう既に設定されているのでそのまま使います。

# Unity

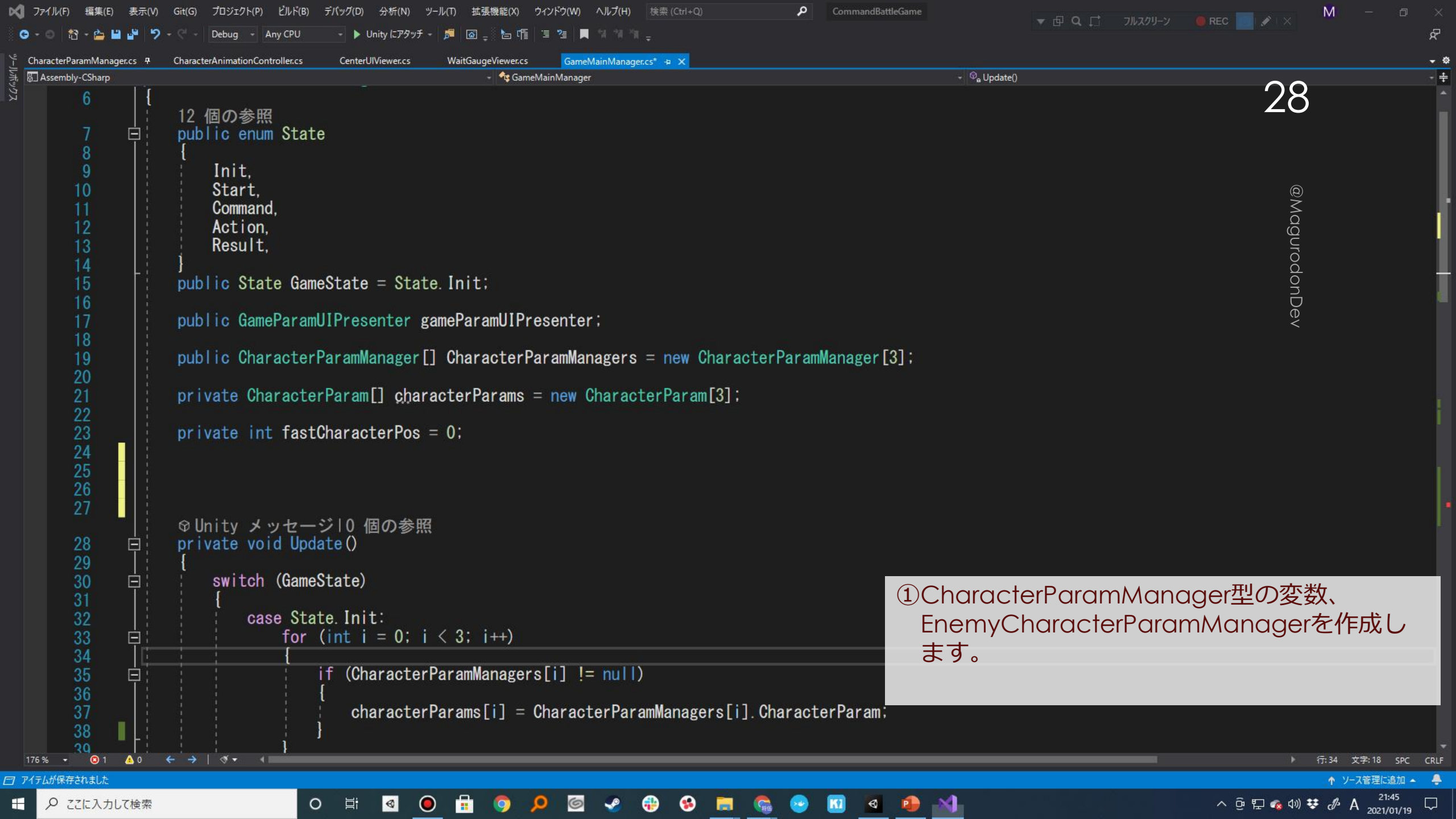
## コマンドバトルゲーム

27

- 今日は最後に敵を倒してみようと思います。
- GameMainManagerで敵のCharacterParamのHPが0を下回ったら、GameStateをResultに飛ばしてみましょう。
- まずは敵の一体のHpで設定してみましょう。



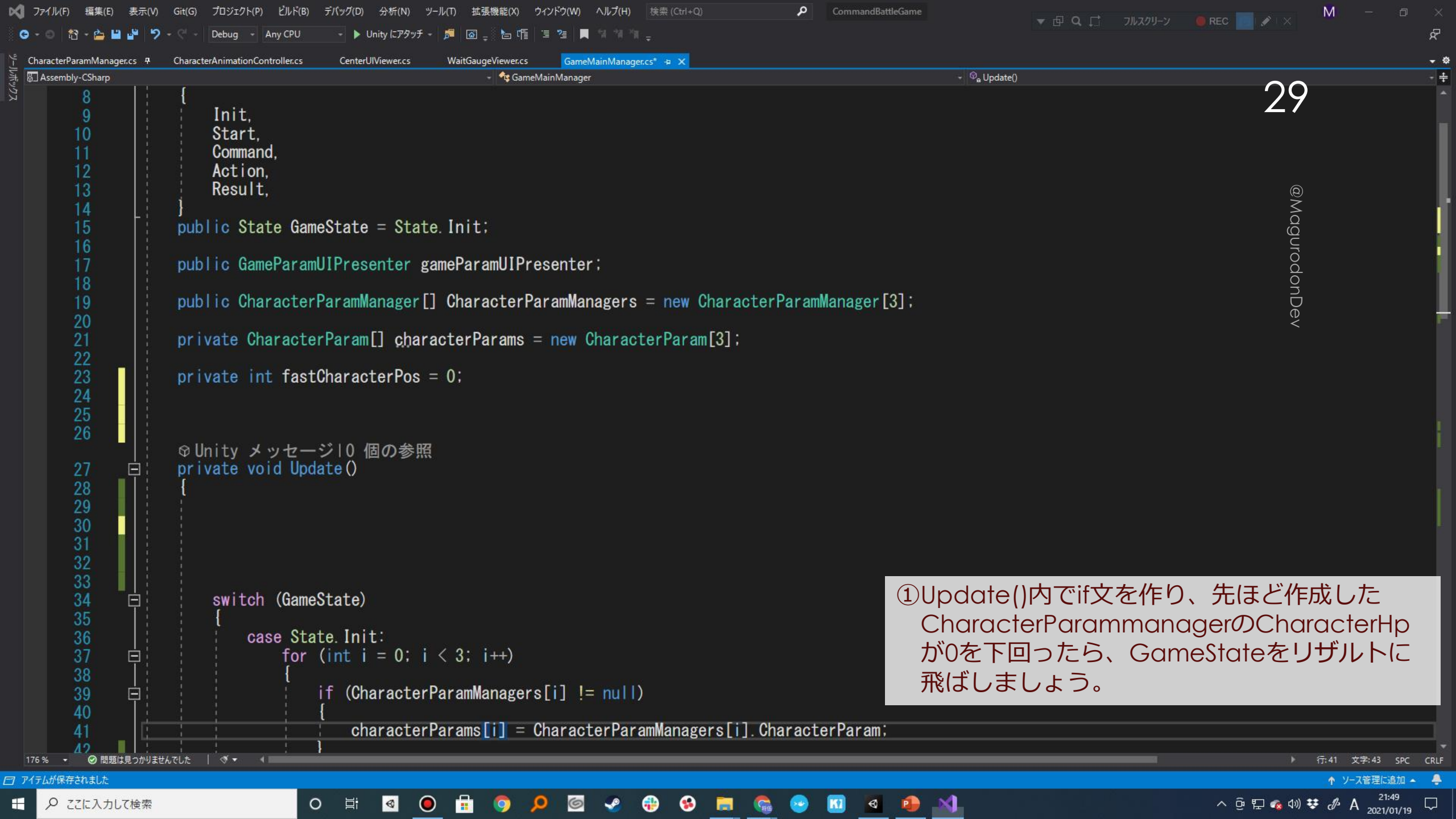




28

@MagurodonDev

①CharacterParamManager型の変数、EnemyCharacterParamManagerを作成します。



29

@MagurodonDev

①Update()内でif文を作り、先ほど作成したCharacterParammanagerのCharacterHpが0を下回ったら、GameStateをリザルトに飛ばしましょう。

# Unity

30

## コマンドバトルゲーム

- Unityに戻り、GameMainManagerに敵のEnemyCharacterParamManagerを設定してください。
- ゲームをプレイしてみましょう。
- 敵のHpを削り切ったら、ゲージの埋まりの進みが止まるはずです。
- 来週はMagicPointの設定や、ヒーラーの設定、及び、敵からの攻撃を行っていかうとおもいます。

