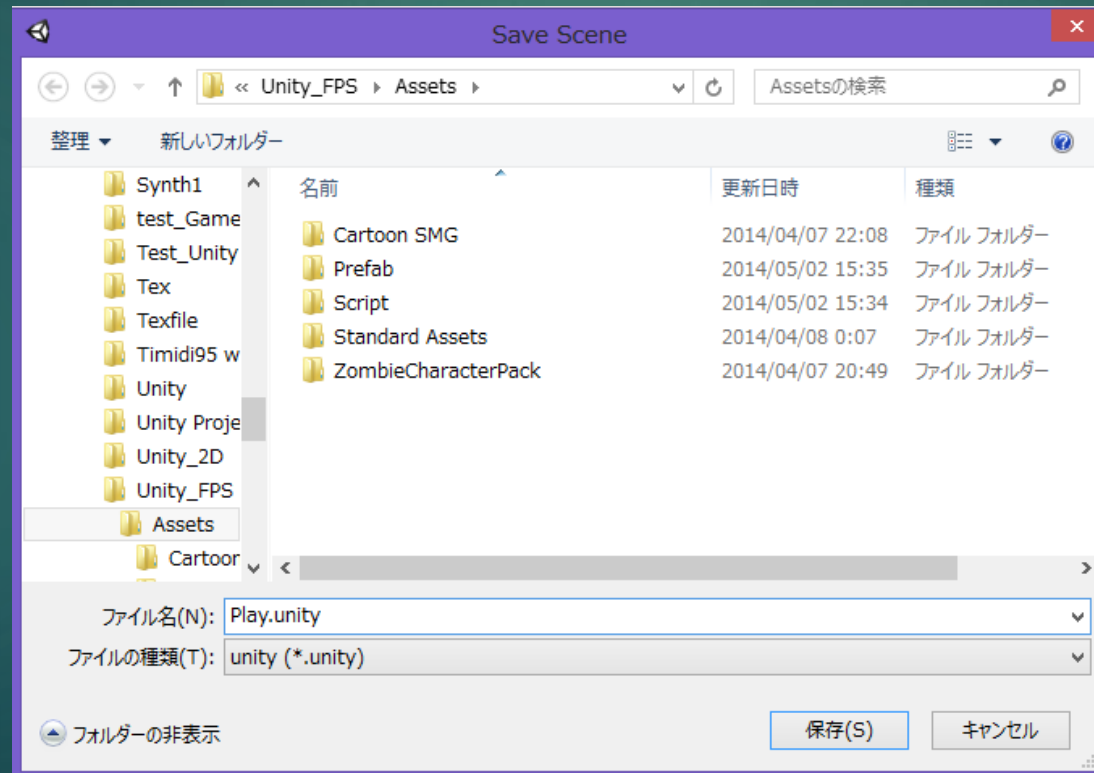


FPS (続き)

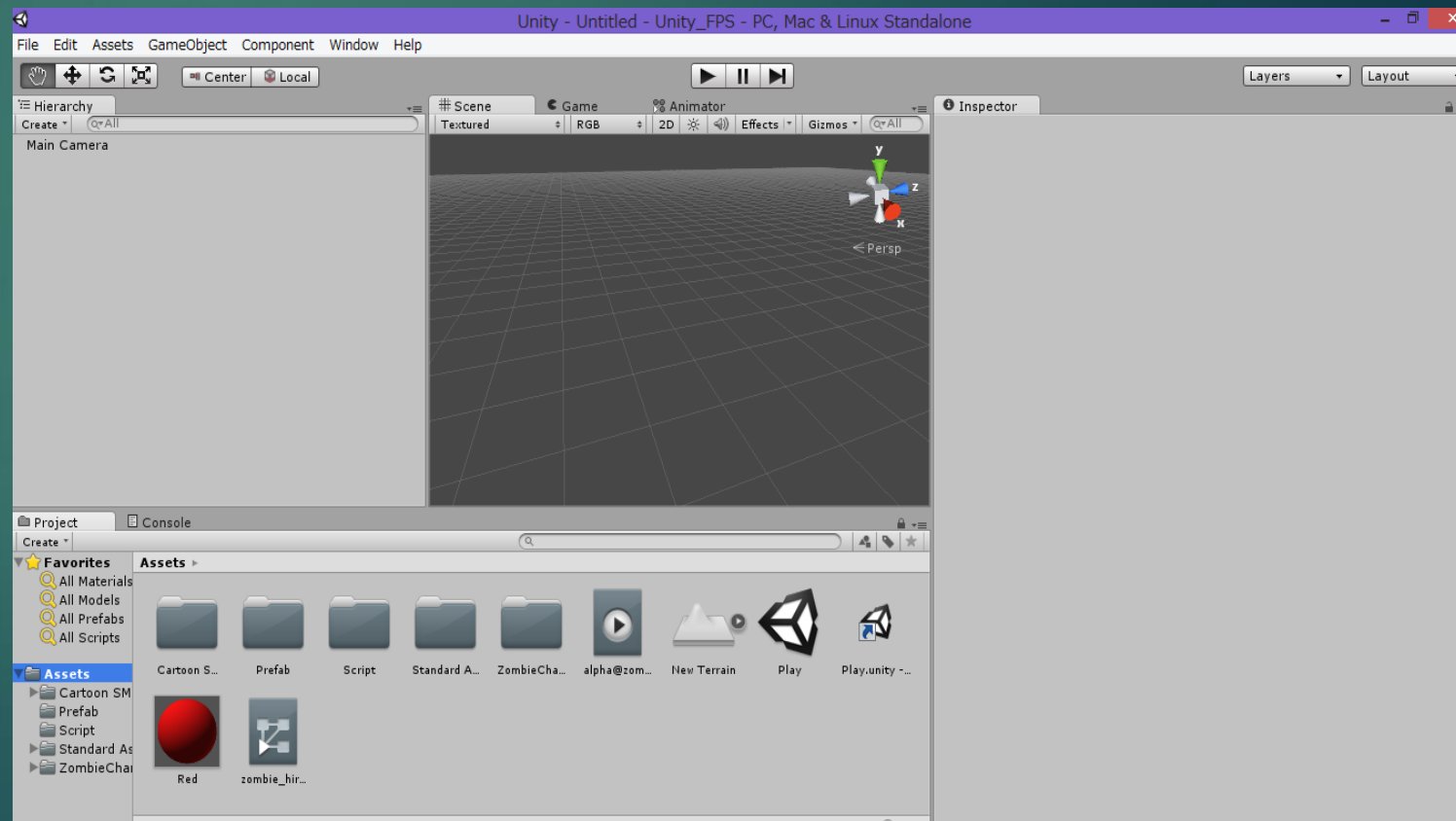
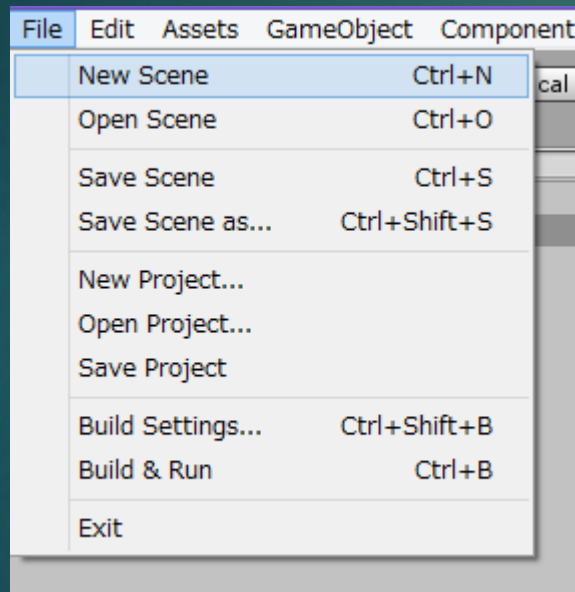
1. タイトル画面の作成

いままで作ったものをセーブする。「CTRL」 + 「S」
名前は「Play」とする。



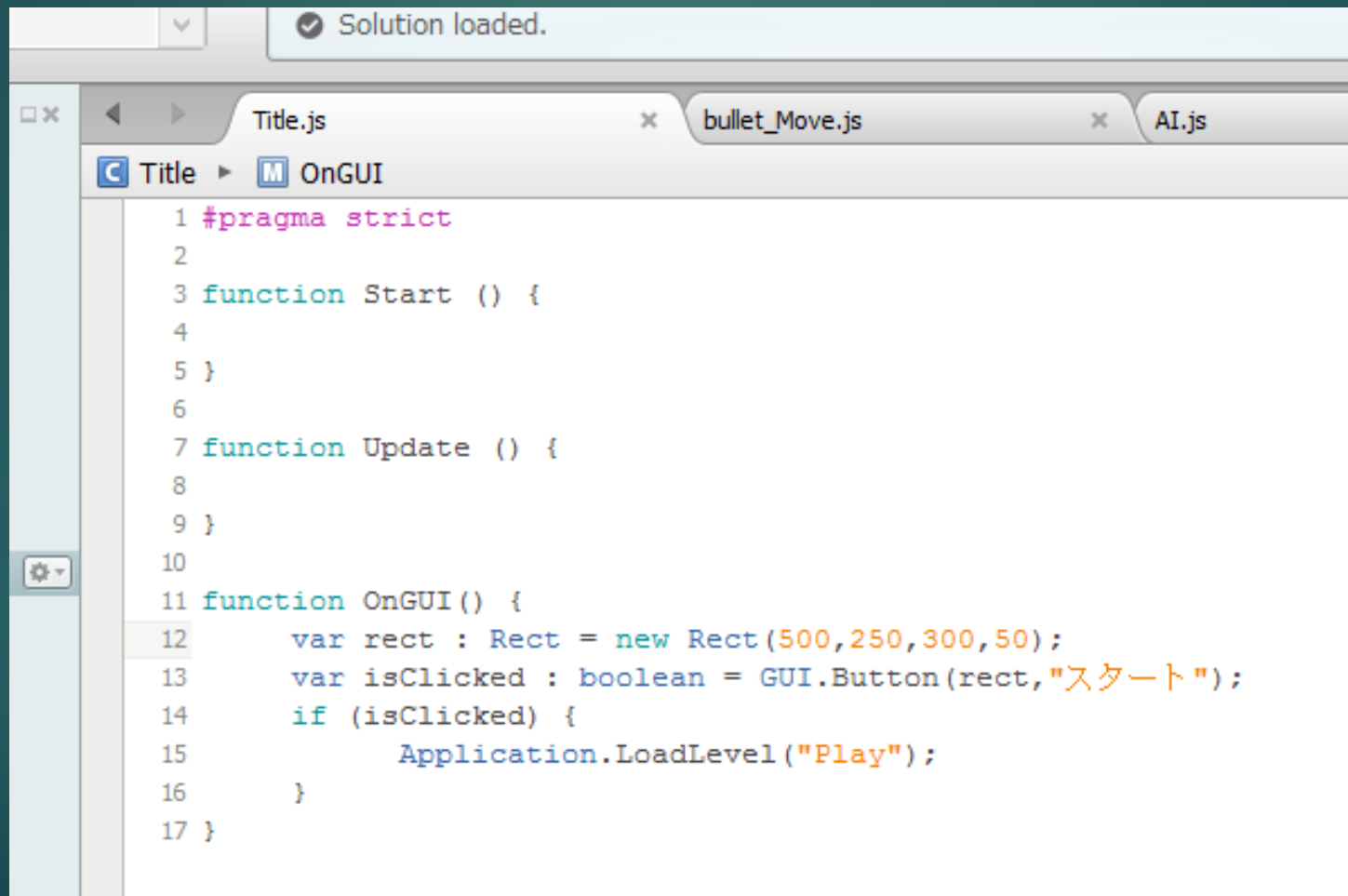
「File」 → 「New Scene」 を選択。

これで新しいゲームシーンが作れる。



スクリプトを書きます。JavaScriptを1つ作成して、名前を「Title」として
ください。

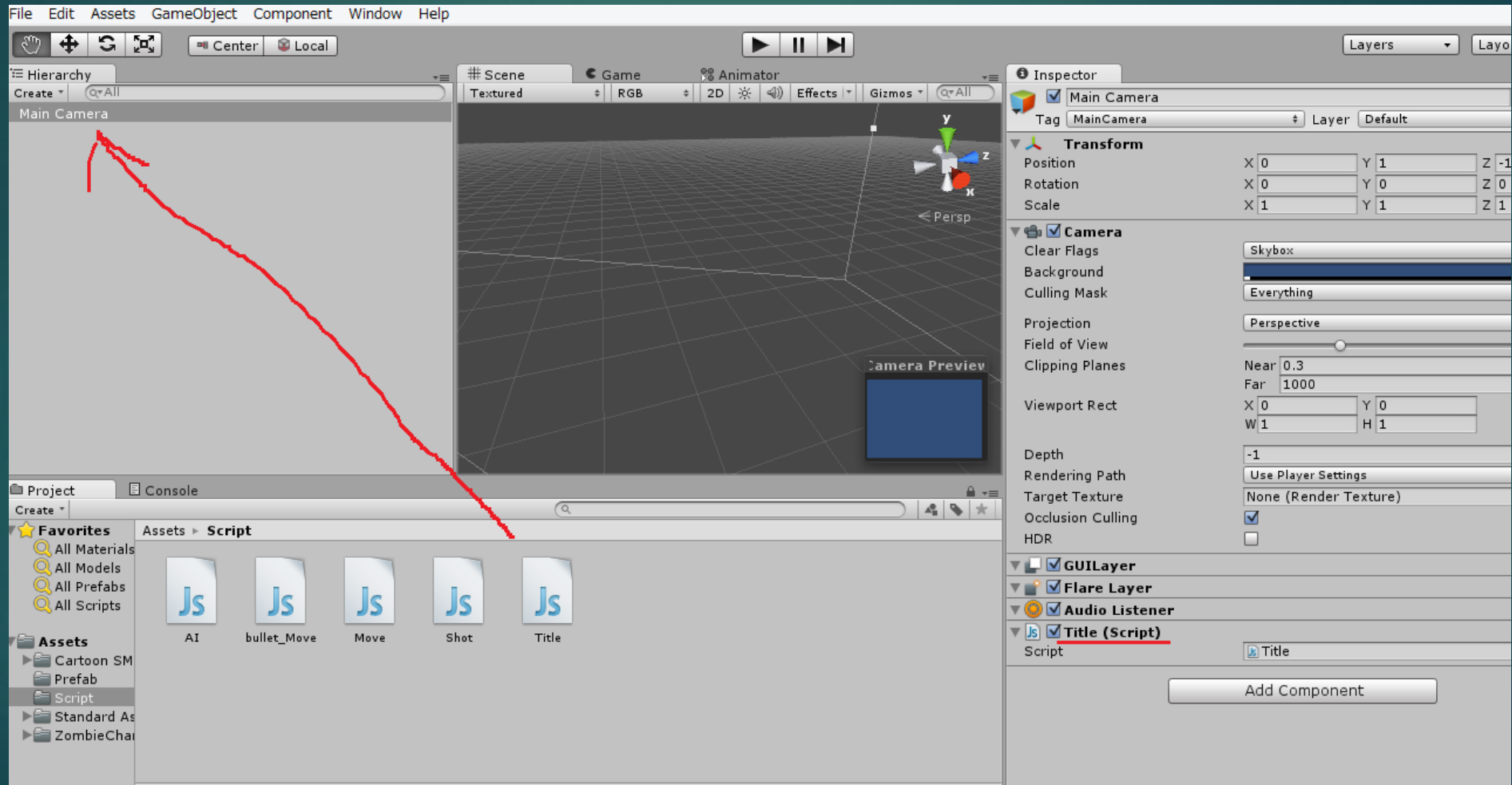
「Title」を開き、下のコードを書いてください。



The screenshot shows a code editor window with the following content:

```
Solution loaded.
Title.js bullet_Move.js AI.js
Title ▶ OnGUI
1 #pragma strict
2
3 function Start () {
4
5 }
6
7 function Update () {
8
9 }
10
11 function OnGUI() {
12     var rect : Rect = new Rect(500,250,300,50);
13     var isClicked : boolean = GUI.Button(rect,"スタート");
14     if (isClicked) {
15         Application.LoadLevel("Play");
16     }
17 }
```

書けたら、それを「Main Camera」にドラッグしてください。
Main CameraのInspectorに図のように出ればOKです。



ここまできたら1回保存。名前を「Title」とする。

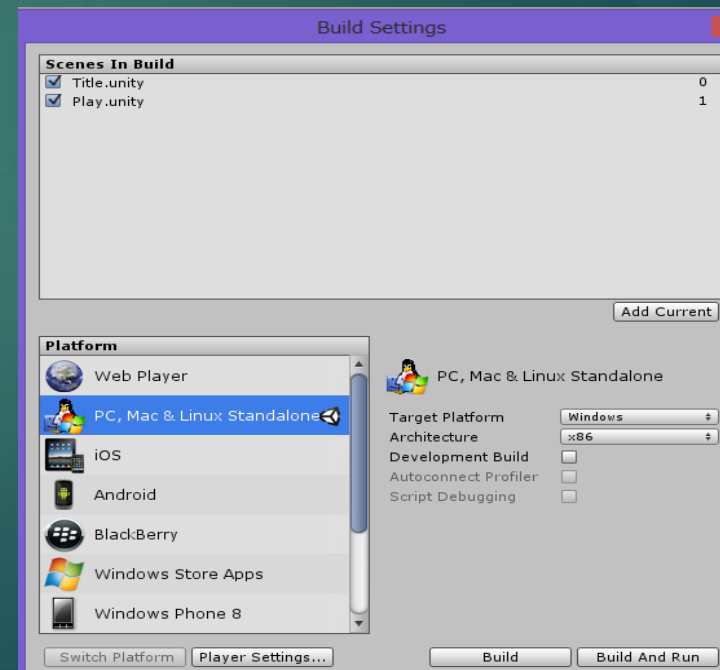
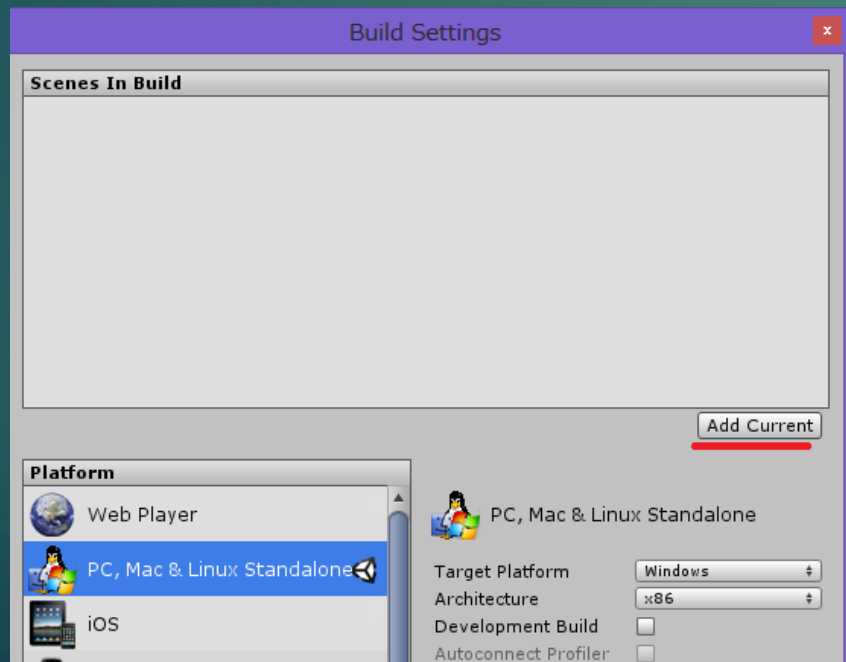
次にシーンとシーン（「Play」と「Title」）を結ぶ作業をします。

メニューの「File」→「Build Settings」を選択。

下のようなウィンドウが出るので「Add Current」を押してください。

そしたらもう一回「Play」を起動させて、同じ作業をしてください。

1番上のシーンから始まるので、「Title」が1番上にくるようにしてください。

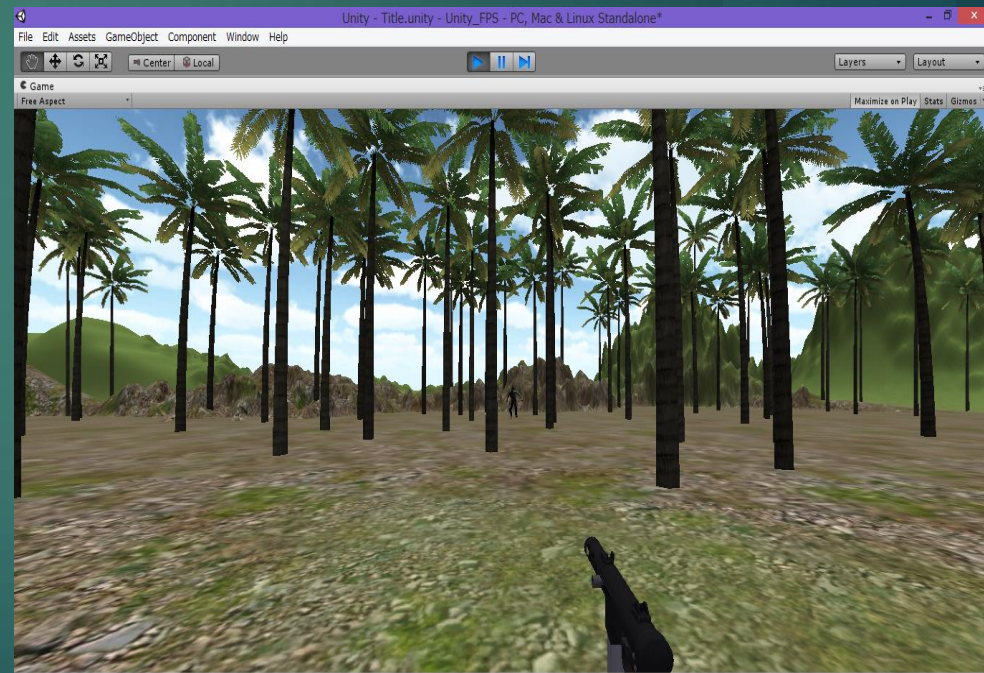
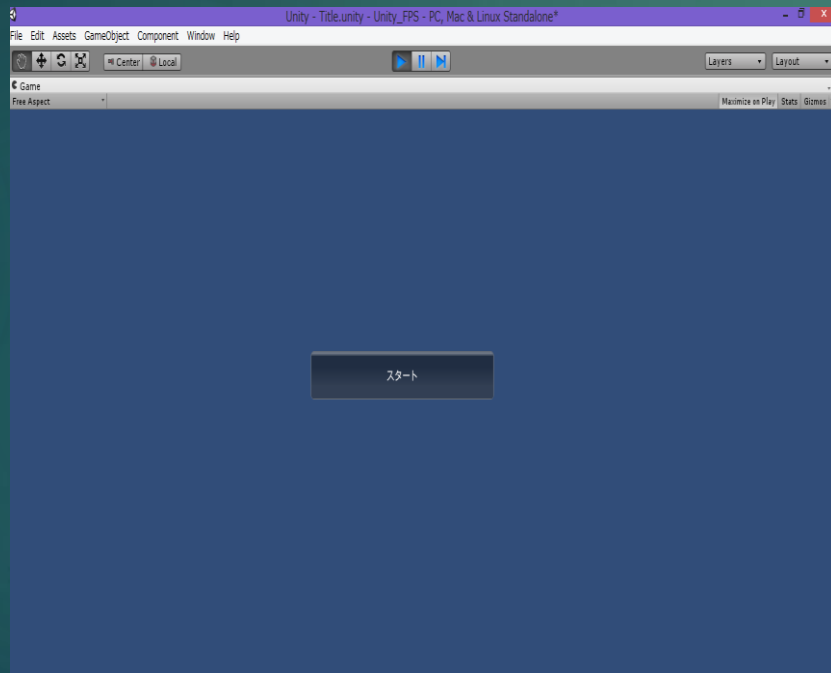


できたら、「Title」に戻ってください。

実行すると、真ん中に「スタート」というボタンがあると思います。

それをクリックすると、いままで作ったゲーム画面に跳ぶと思います。

これでシーンとシーンを繋ぐことに成功しました。



シーンの切り替えをする場合は、今の作業をやるとシーンが繋がるようになります。

JavaScriptの場合は、

```
if (ここにシーン遷移の条件を書く) {  
    Application.LoadLevel(“遷移先のシーン名”);  
}
```

で書けば大丈夫です。

C#の方はまだやったことないのでわかりません(´Δ`;))

2. ゲームのルールを決める

少しでもゲームに近づけるためにルールを決める。

今回は、

- ・ゾンビを倒したら得点が入る。
- ・制限時間が過ぎたらゲームクリア→結果画面
- ・ゾンビに触れたらプレイヤーのHPが減る→0になったらゲームオーバー画面へ

ひとまずこんな感じにする。

3. ゾンビを倒したら得点が入るようにする。

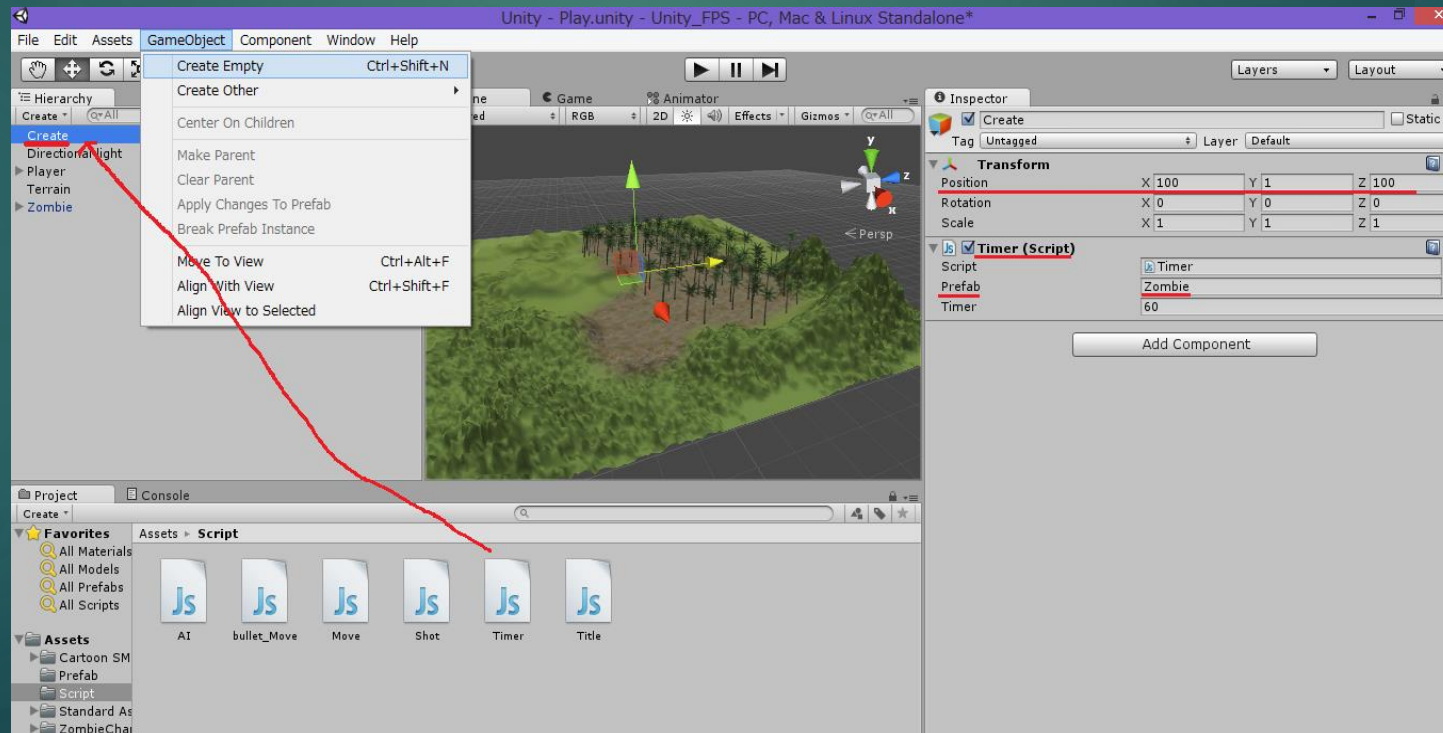
まずタイマーとゾンビをランダムに発生させるスクリプトを書く。

JavaScriptを1つ作成する。名前は「Timer」にする。

Timerの中身

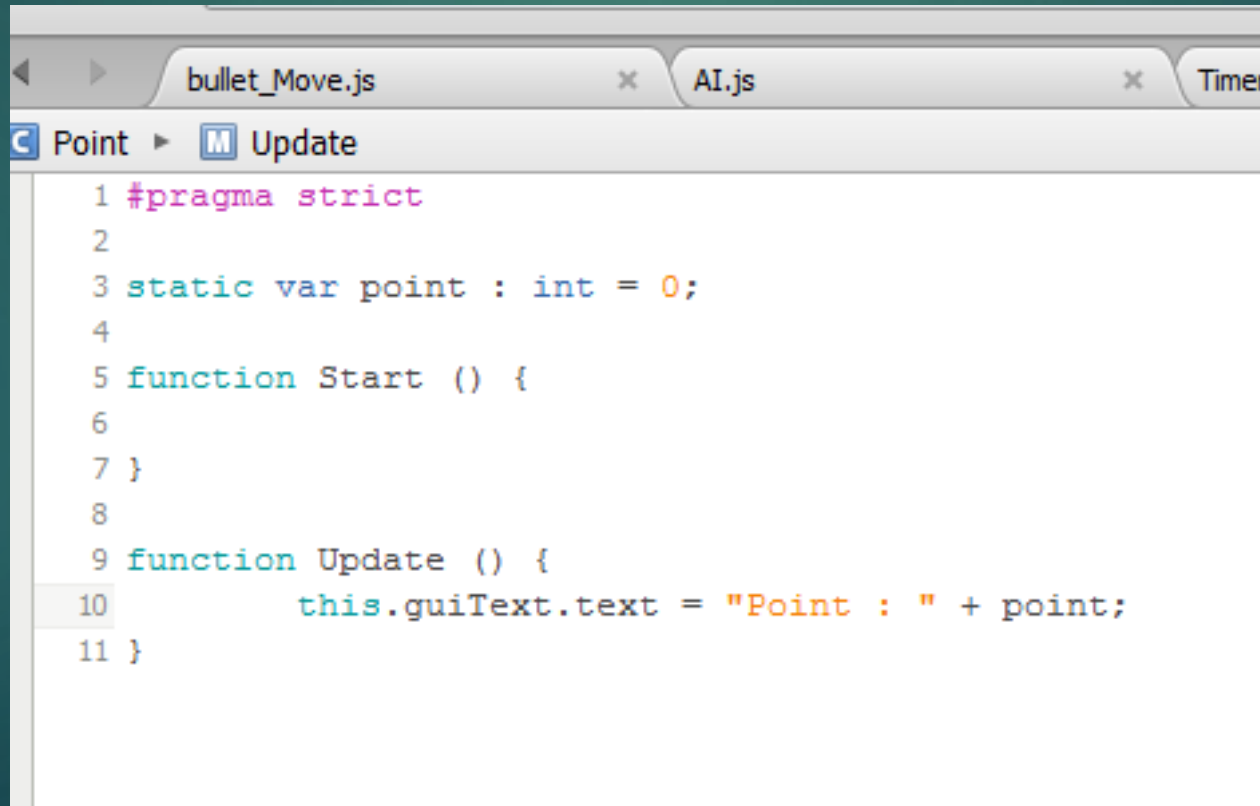
```
bullet_Move.js x AI.js x Timer.js x Shot.js x
Timer ▶ M Update
1 #pragma strict
2
3 var prefab : GameObject;
4 var timer : float = 60.0;
5
6 function Start () {
7
8 }
9
10 function Update () {
11     timer -= Time.deltaTime;
12     if (timer >= 29.9 && timer <= 30.0){
13         Spawn();
14     }
15 }
16
17 function Spawn(){
18     var xz = Random.insideUnitCircle * 100;
19     var newPosition = Vector3(xz.x,0,xz.y)+transform.position;
20     Instantiate(prefab, newPosition, transform.rotation);
21 }
```

書いたら「Game Object」 → 「Create Empty」で空のゲームオブジェクトを作成。名前は「Create」とする。「Create」に「Timer」をドラッグする。「Create」のInspectorのPosition欄の値を画像のようにする。Prefab欄の右側にある丸をクリックして、プレファブの「Zombie」を指定する。



これで、制限時間を60秒にすることができ、30秒になったら、ゾンビを複数生成することができたはず！

次にゾンビを倒したら得点が入るようにしたいので、JavaScriptを作成する。名前は「Point」とする。



```
bullet_Move.js x AI.js x Timer.  
C Point ▶ M Update  
1 #pragma strict  
2  
3 static var point : int = 0;  
4  
5 function Start () {  
6  
7 }  
8  
9 function Update () {  
10     this.guiText.text = "Point : " + point;  
11 }
```

次にスクリプトの「AI」に少し書き足す。

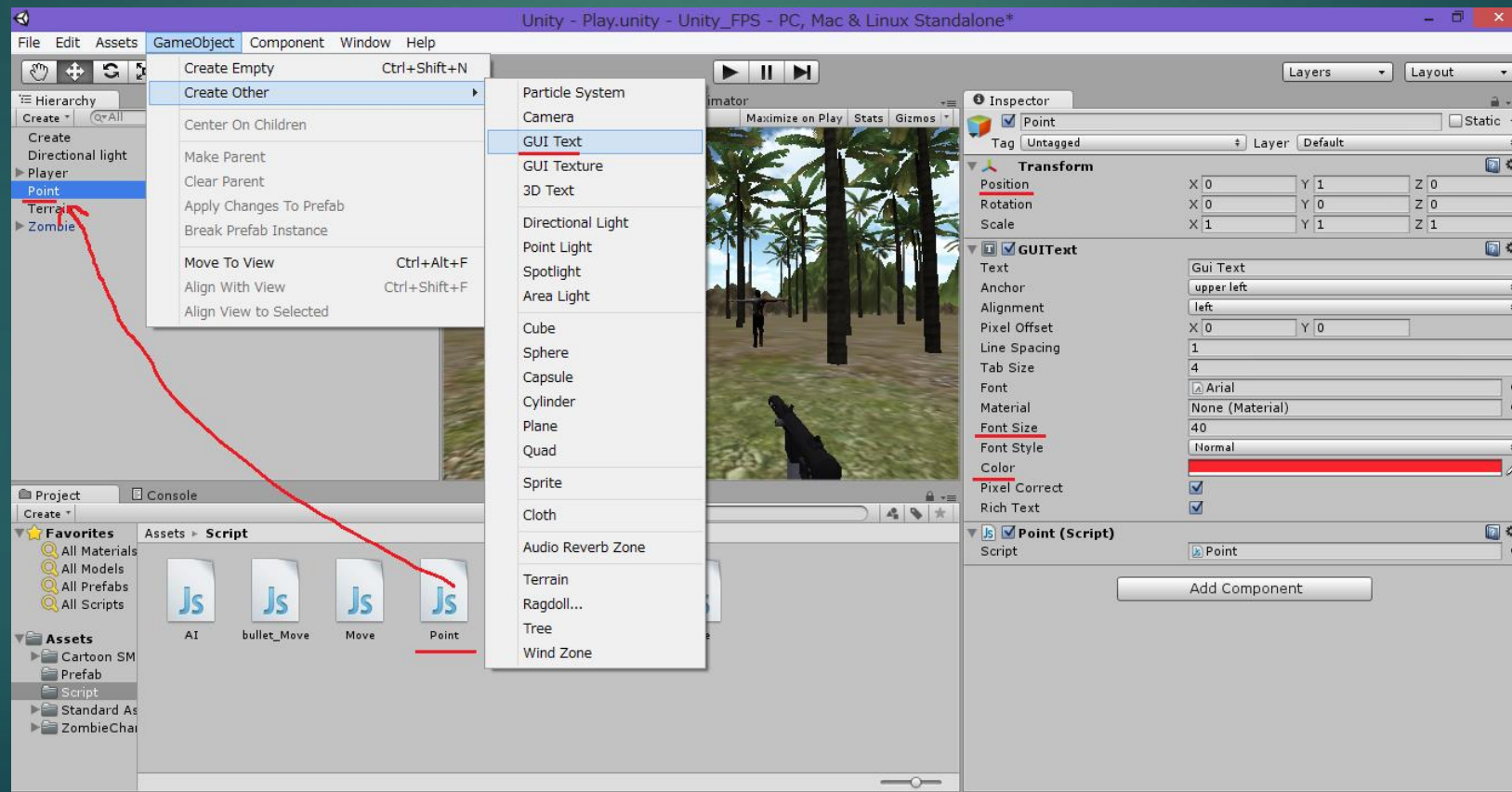
```
bullet_Move.js x AI.js x Timer.js x Shot.js
AI OnCollisionEnter
20     this.transform.position.y, target.transform.position.z);
21     this.transform.rotation = Quaternion.Slerp(this.transform.rotation,
22     Quaternion.LookRotation(targetDirection - this.transform.position), Time
23     moveDirection += transform.forward * 1;
24     moveDirection.y -= gravity * Time.deltaTime;
25     controller.Move(moveDirection * Time.deltaTime * speed);
26     }
27     }
28 }
29
30 var count : int = 0;
31
32 function OnCollisionEnter(collision : Collision) {
33     if (collision.gameObject.name == "bullet(Clone)") {
34         count = count + 1;
35         if (count == 3) {
36             Point.point = Point.point + 200;
37             Destroy(gameObject);
38         }
39     }
40 }
```

← この行だけ

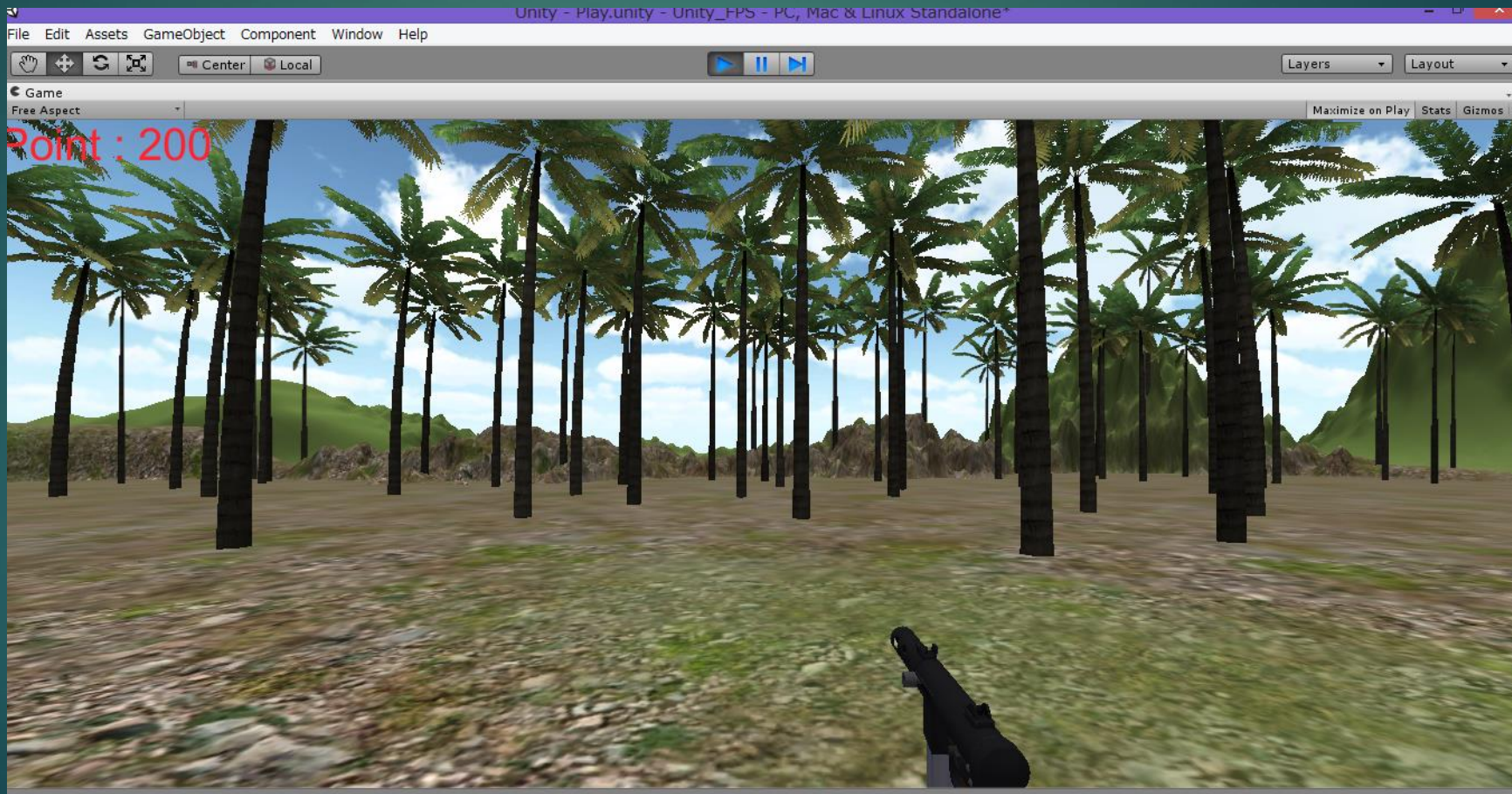
書いたら、「Game Object」 → 「Create Other」 → 「GUI Text」 をクリック

名前を「Point」とする。スクリプトの「Point」をそれにドラッグ

Inspectorの値はひとまず図のようにしてください。後で変えても大丈夫です。

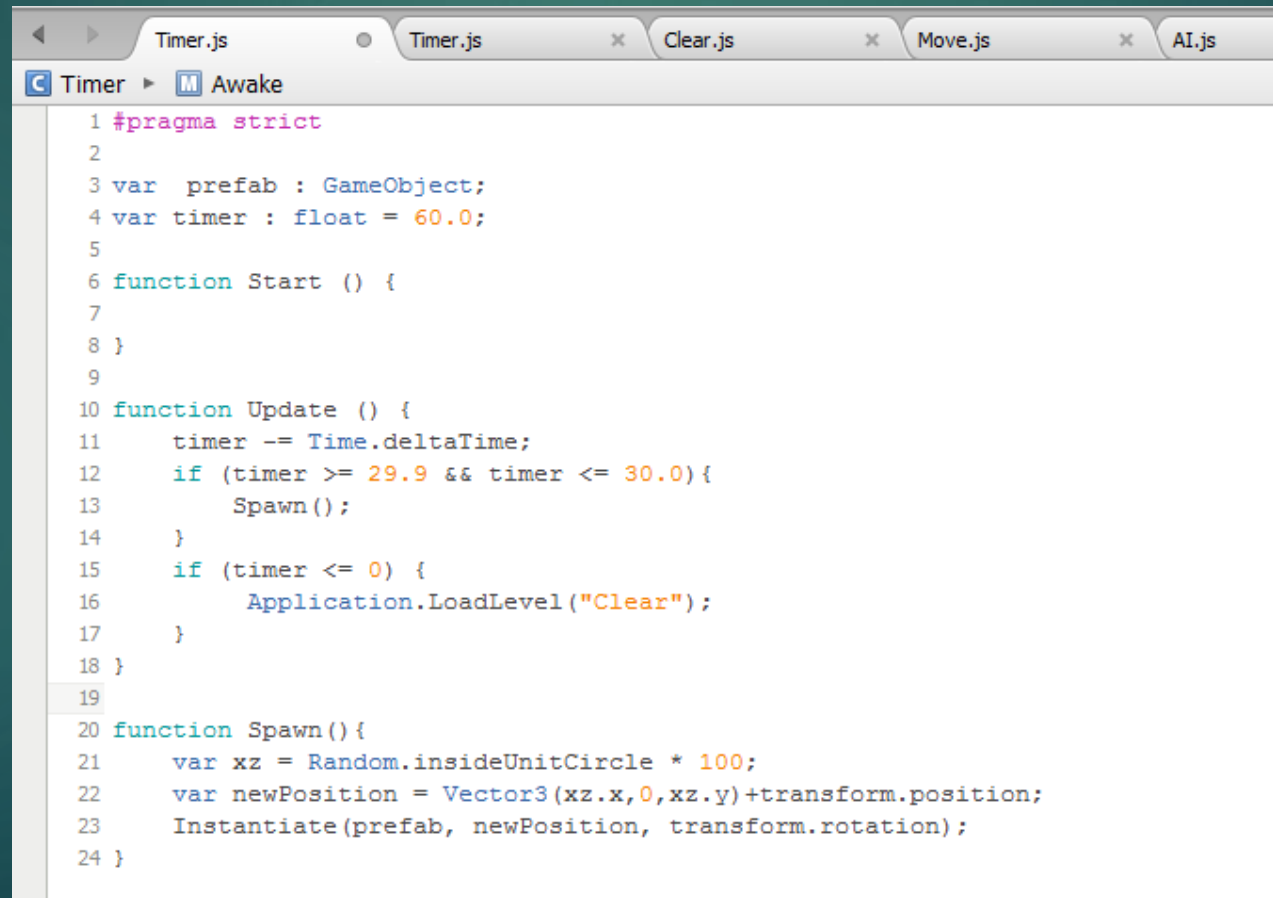


これで実行してみると、画面の左上にポイントが表示されるようになりました。ゾンビを1体倒すごとに200ポイント加わるのが確認できたら成功です。



4. 制限時間を過ぎたらゲームクリア

スクリプトの「Timer」に書き加える。

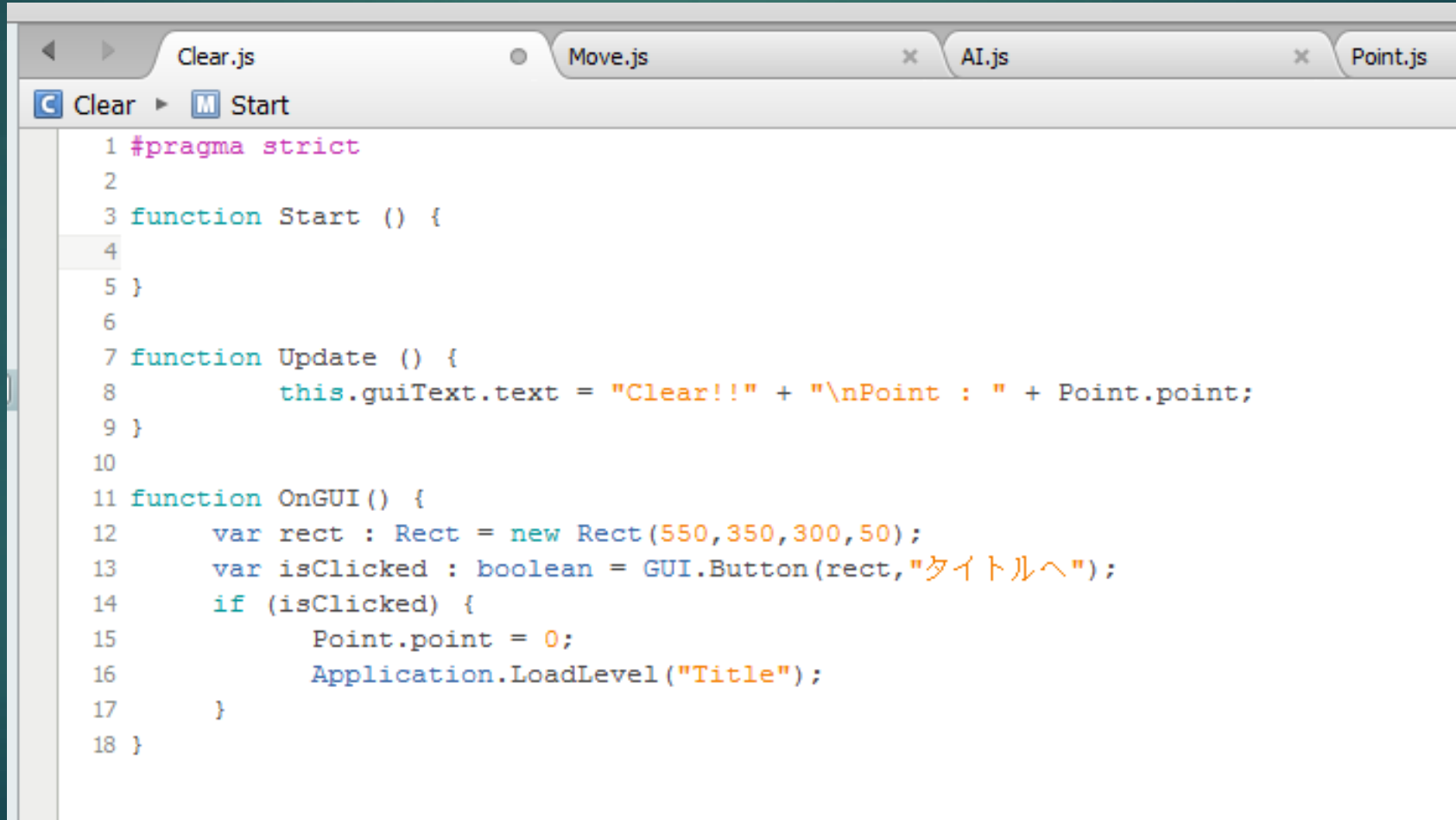


```
1 #pragma strict
2
3 var prefab : GameObject;
4 var timer : float = 60.0;
5
6 function Start () {
7
8 }
9
10 function Update () {
11     timer -= Time.deltaTime;
12     if (timer >= 29.9 && timer <= 30.0){
13         Spawn();
14     }
15     if (timer <= 0) {
16         Application.LoadLevel("Clear");
17     }
18 }
19
20 function Spawn(){
21     var xz = Random.insideUnitCircle * 100;
22     var newPosition = Vector3(xz.x,0,xz.y)+transform.position;
23     Instantiate(prefab, newPosition, transform.rotation);
24 }
```

さっきやったシーン遷移をまたやる。

「File」→「New Scene」で新しいシーンを作成。

JavaScriptを作成。名前を「Clear」とする。

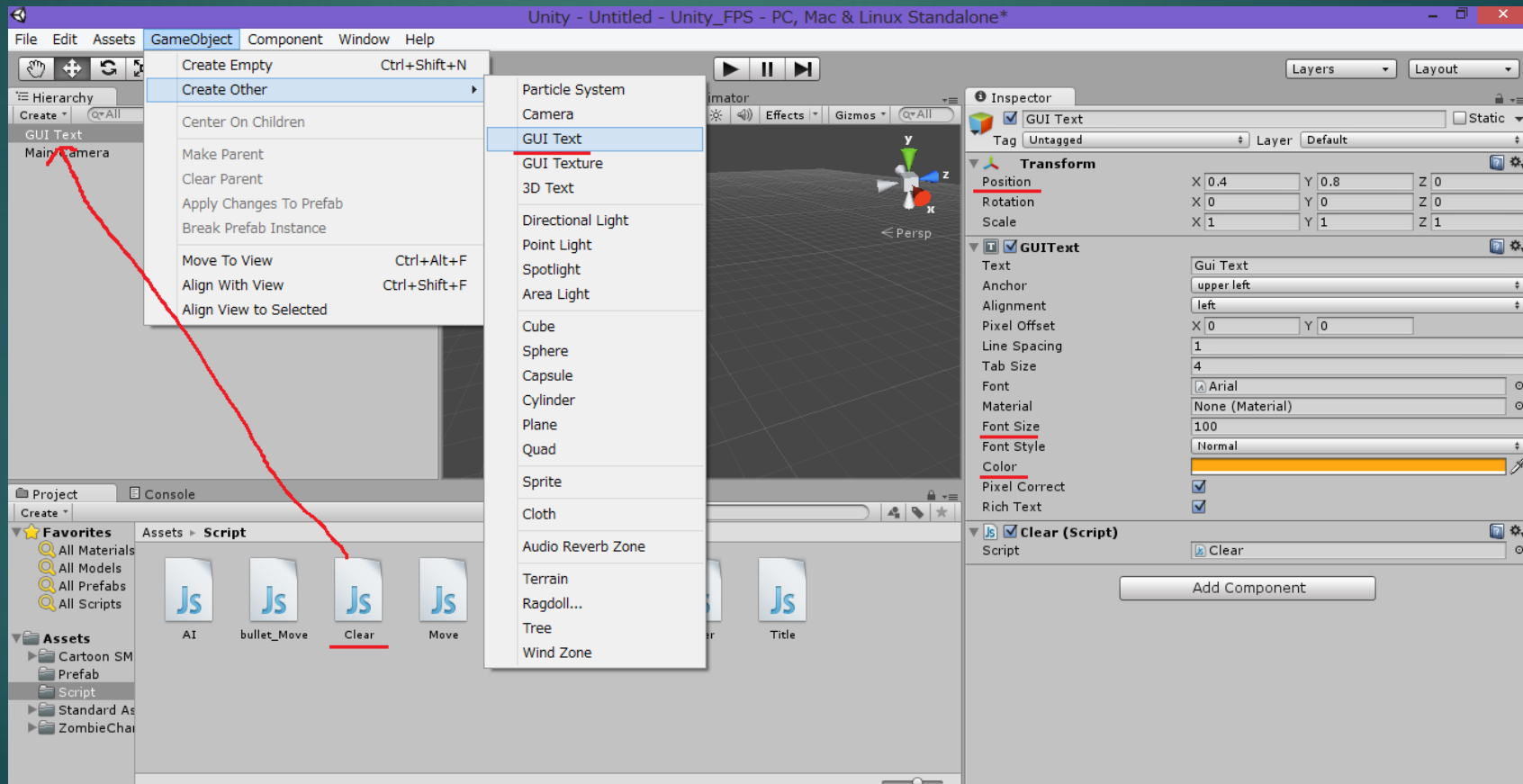


```
Clear.js  Move.js  AI.js  Point.js
Clear ▶ Start
1 #pragma strict
2
3 function Start () {
4
5 }
6
7 function Update () {
8     this.guiText.text = "Clear!!" + "\nPoint : " + Point.point;
9 }
10
11 function OnGUI () {
12     var rect : Rect = new Rect(550,350,300,50);
13     var isClicked : boolean = GUI.Button(rect,"タイトルへ");
14     if (isClicked) {
15         Point.point = 0;
16         Application.LoadLevel("Title");
17     }
18 }
```

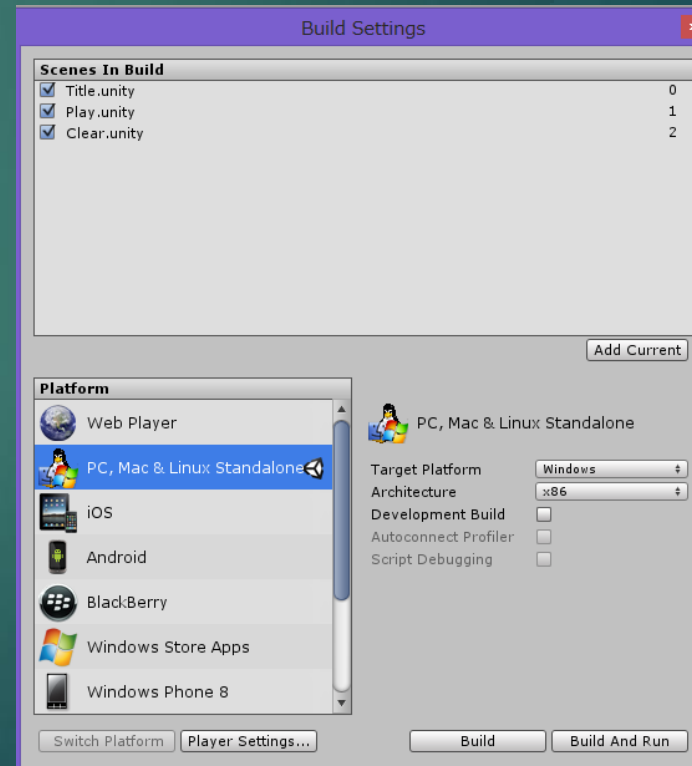
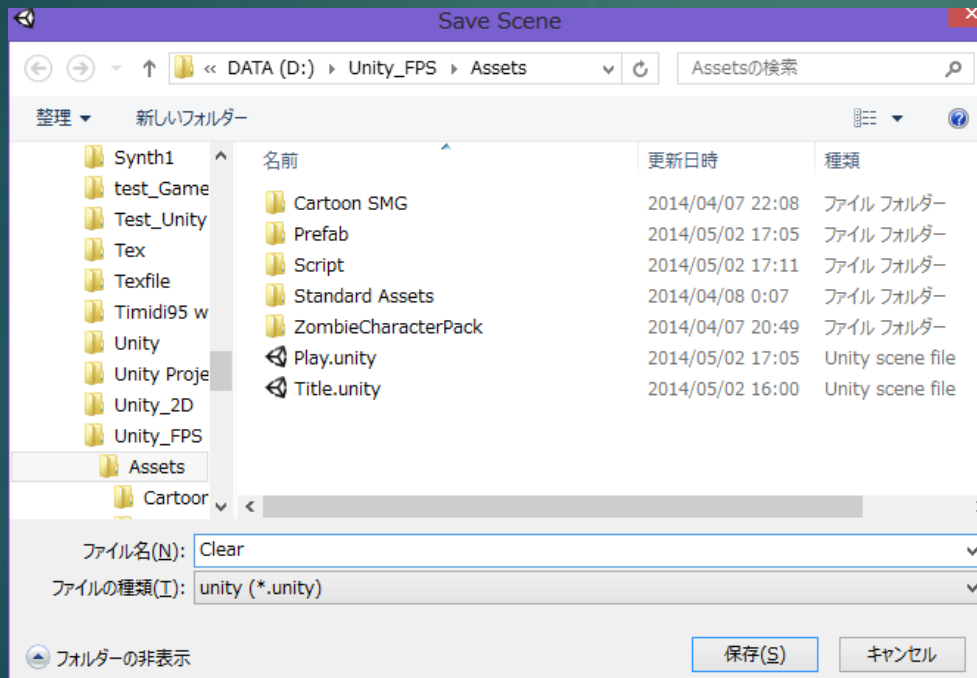
書いたら、「Game Object」 → 「Create Other」 → 「GUI Text」を選択。

「Clear」をドラッグする。

Inspectorのそれぞれの値は図のようになる。



シーンを保存する。名前は「Clear」とする。
保存したら「File」→「Build Settings」を選択。
さっきみたいに「Add Current」をクリック。
これでシーン遷移が可能になったのでクリア画面ができた！



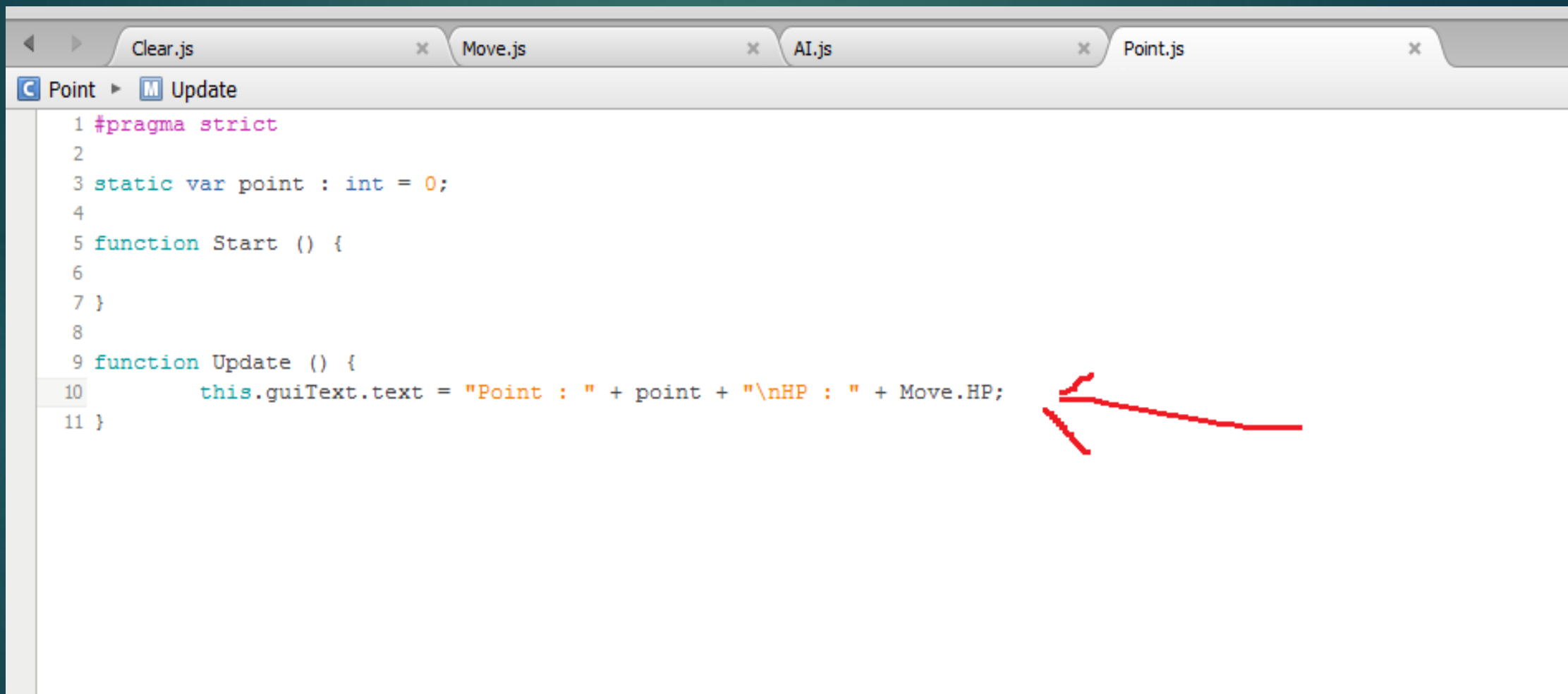
5. ゾンビに触れるとHPが減るようにする

スクリプトの「Move」に書き足す。

```
Clear.js x Move.js x AI.js
C Move ▶ F HP
1 #pragma strict
2
3 var speed : float = 10.0;
4 var rotationspeed : float = 100.0;
5 static var HP : int = 1000;
6
7 function Start () {
8
9 }
10
11 function Update () {
12     var translation : float = Input.GetAxis("Vertical")*speed;
13     var rotation : float = Input.GetAxis("Horizontal")*rotationspeed;
14     var height : float = Terrain.activeTerrain.SampleHeight(transform.position)+1-transform.position.y;
15     translation*=Time.deltaTime;
16     rotation*=Time.deltaTime;
17     transform.Translate(0,height,translation);
18     transform.Rotate(0,rotation,0);
19 }
20
21 function OnCollisionEnter(collisionInfo : Collision) {
22     if (collisionInfo.gameObject.tag == 'Zombie' || collisionInfo.gameObject.tag == 'Zombie(Clone)') {
23         HP = HP - 10;
24     }
25     if (HP <= 0) {
26         Application.LoadLevel("GameOver");
27     }
28 }
```



「Point」に書き加える。




```
1 #pragma strict
2
3 static var point : int = 0;
4
5 function Start () {
6
7 }
8
9 function Update () {
10     this.guiText.text = "Point : " + point + "\nHP : " + Move.HP;
11 }
```


「Clear」に書き加える。

```
Clear.js  Move.js  AI.js
Clear ▶ Start
1 #pragma strict
2
3 function Start () {
4     Move.HP = 1000;
5 }
6
7 function Update () {
8     this.guiText.text = "Clear!!" + "\nPoint : " + Point.point;
9 }
10
11 function OnGUI () {
12     var rect : Rect = new Rect(550,350,300,50);
13     var isClicked : boolean = GUI.Button(rect,"タイトルへ");
14     if (isClicked) {
15         Point.point = 0;
16         Application.LoadLevel("Title");
17     }
18 }
```

「File」 → 「New Scene」 で新しいシーンを作成。
JavaScriptを作成。名前は「GameOver」とする。

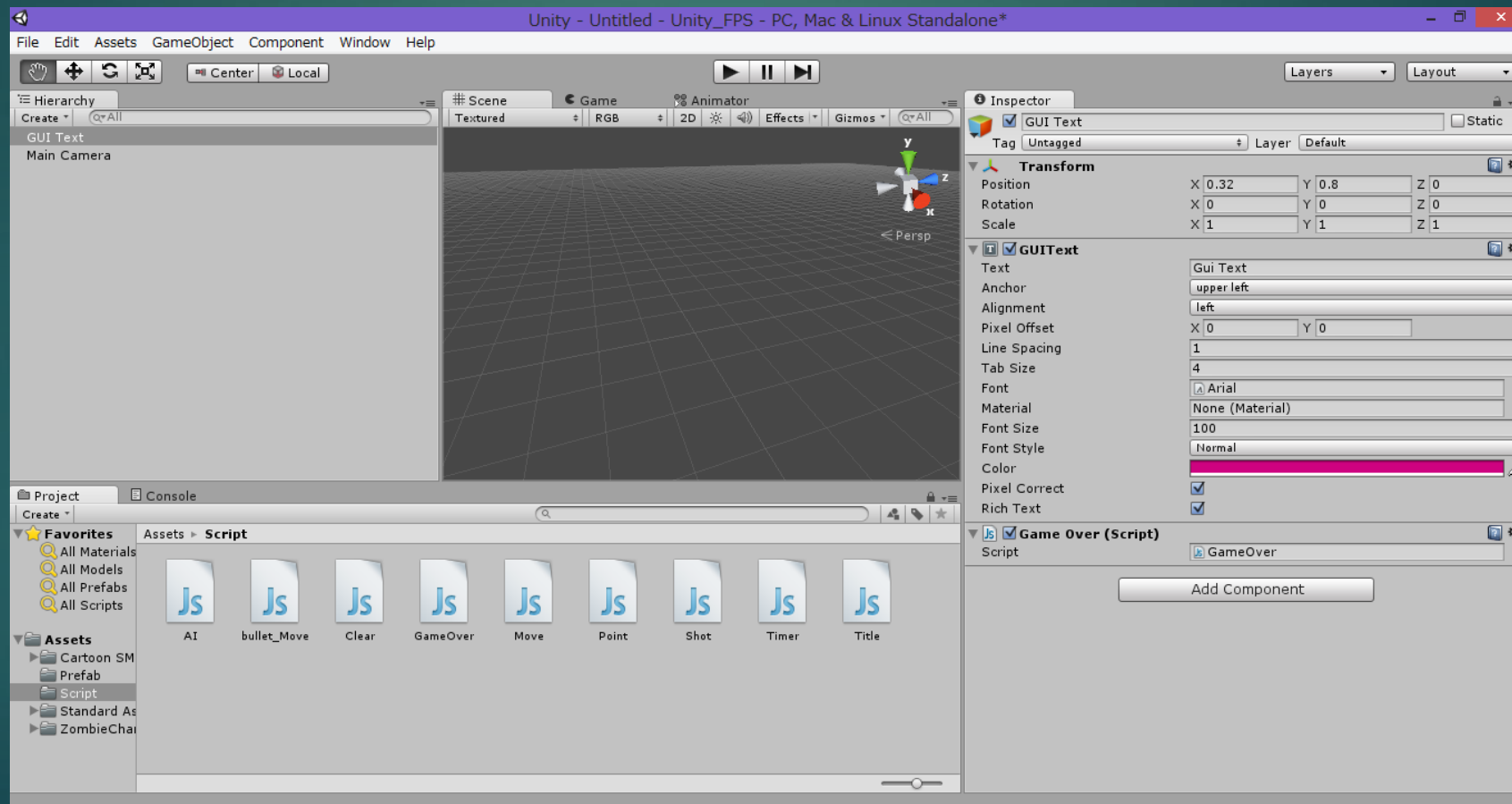


```
1 #pragma strict
2
3 function Start () {
4     Point.point = 0;
5     Move.HP = 1000;
6 }
7
8 function Update () {
9     this.guiText.text = "Game Over";
10 }
11
12 function OnGUI() {
13     var rect : Rect = new Rect(550,350,300,50);
14     var isClicked : boolean = GUI.Button(rect,"タイトルへ");
15     if (isClicked) {
16         Application.LoadLevel("Title");
17     }
18 }
```

書いたら、「Game Object」 → 「Create Other」 → 「GUI Text」 を選択。

「GameOver」 をドラッグする。

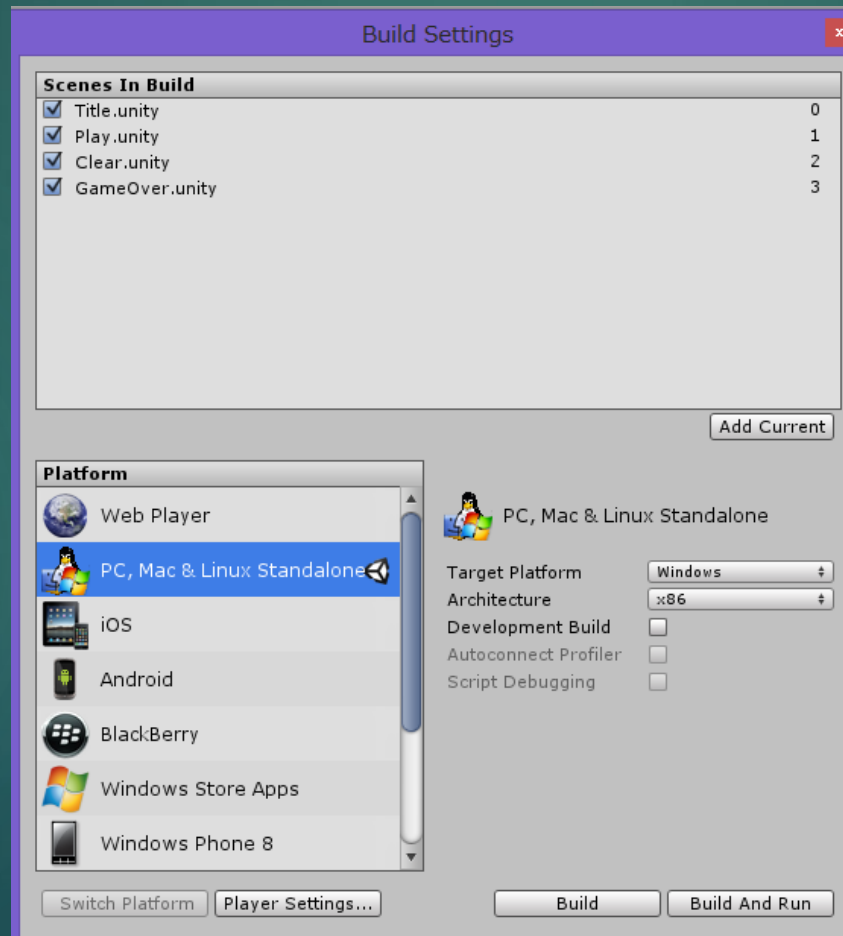
Inspectorのそれぞれの値は図のようになる。



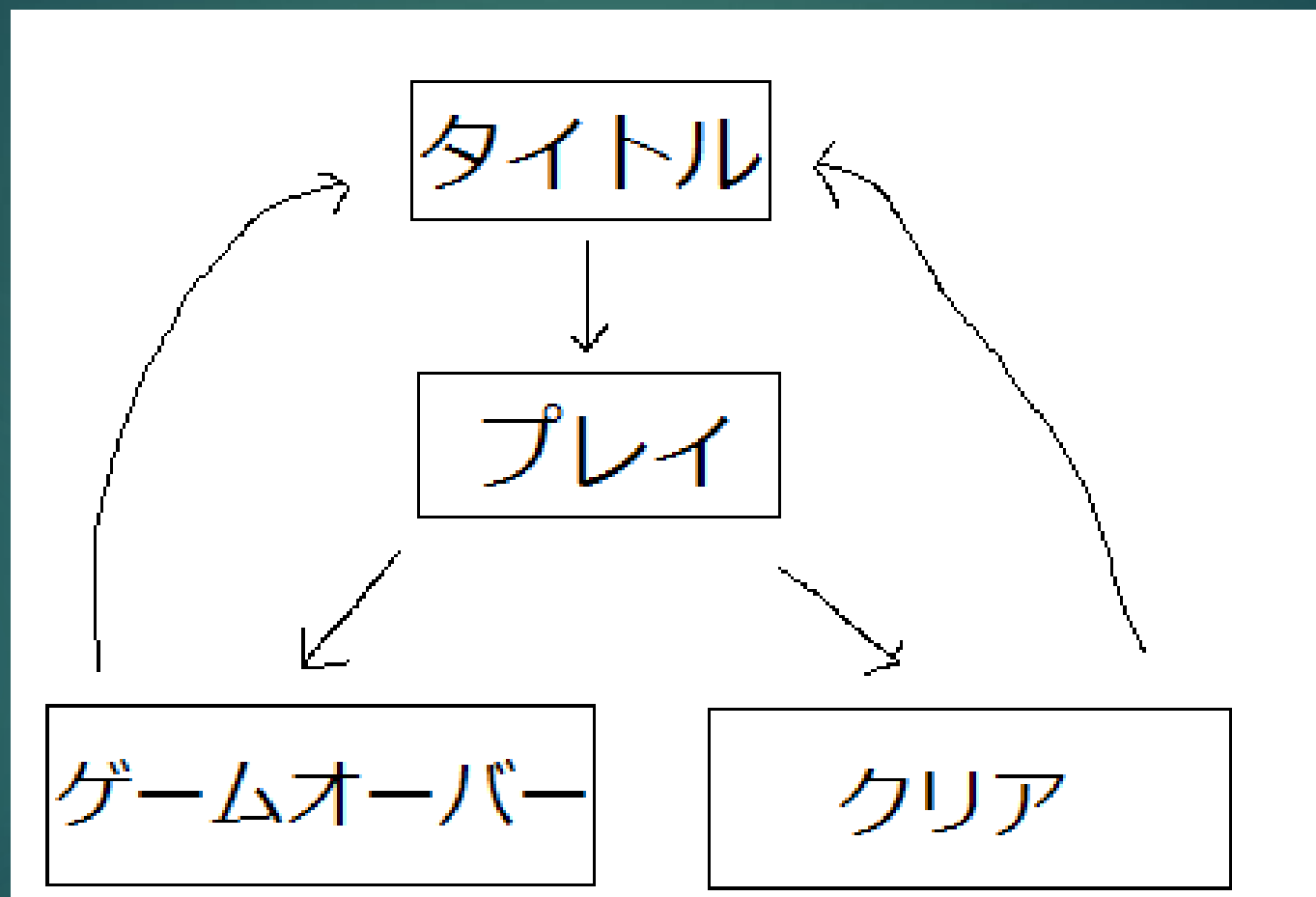
書いたら保存する。名前は「GameOver」とする。

そして「File」→「Build Settings」を選択。

「Add Current」を押す。これで4つのシーンが繋がった。

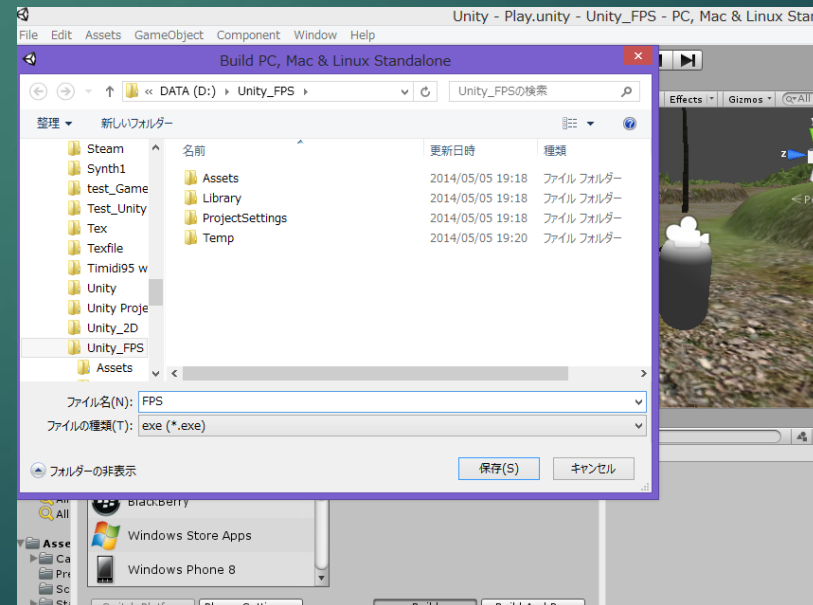
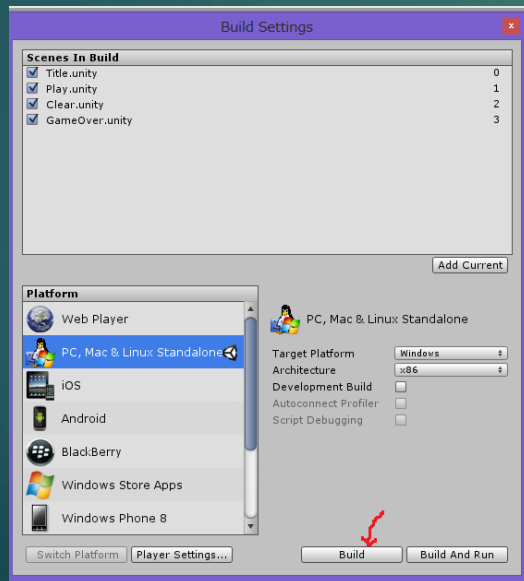


最終的にはこのようになってる。(タイトルがスタート)



6. Buildのやり方

「File」 → 「Build Settings」 を選択。ここでどのPlatformにするか決められます。Windowsでやりたいので、「PC、Mac、Linux、・・・」を選び、Target Platformがwindowsになっているのが確認できたらBuildをクリックしてください。名前はなんでもOKです。（画像だと「FPS」）Scenes In Buildの0 になっているシーンから開始されます。

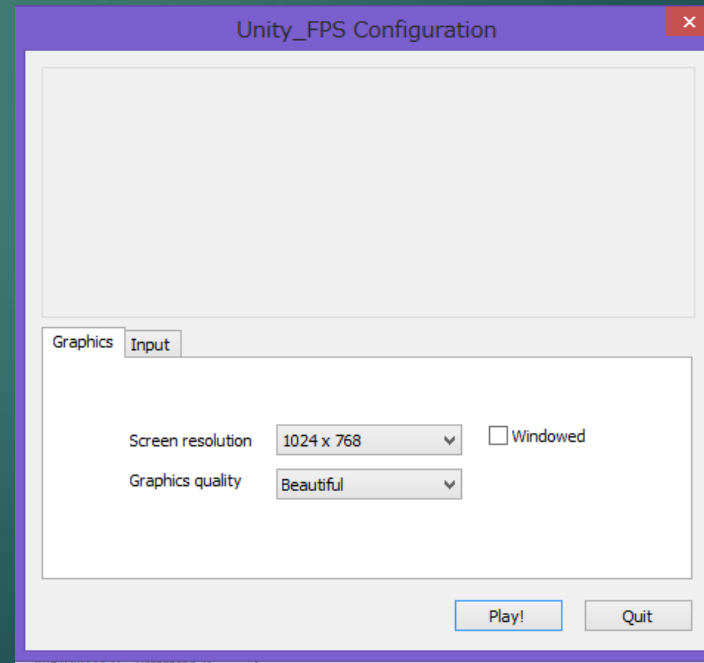
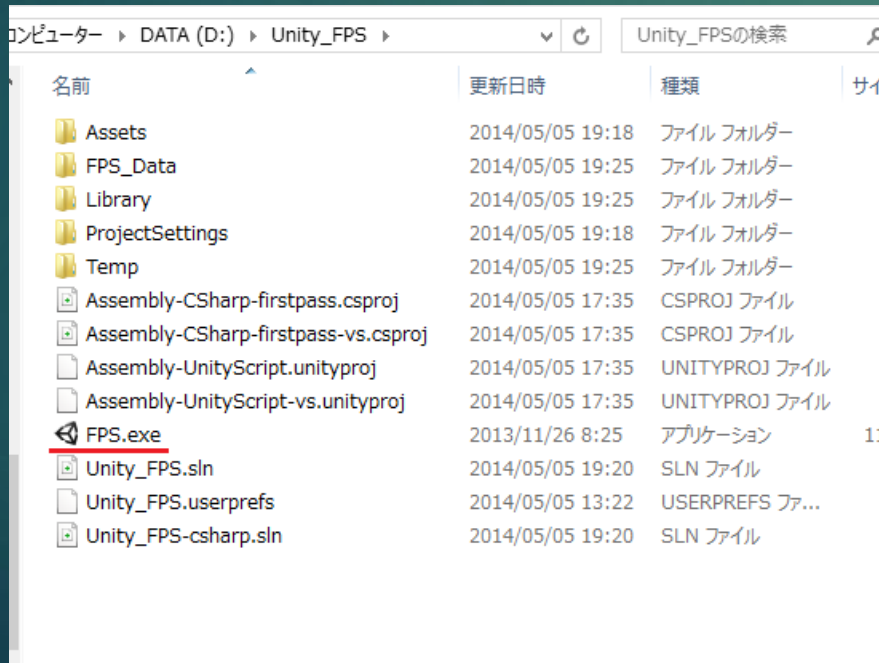


Buildできたら下のようにFPSというアプリケーションができてると
思います。これを実行したら、さっき作ったゲームが実行できます。

Graphics qualityは好きなものを選んでください。

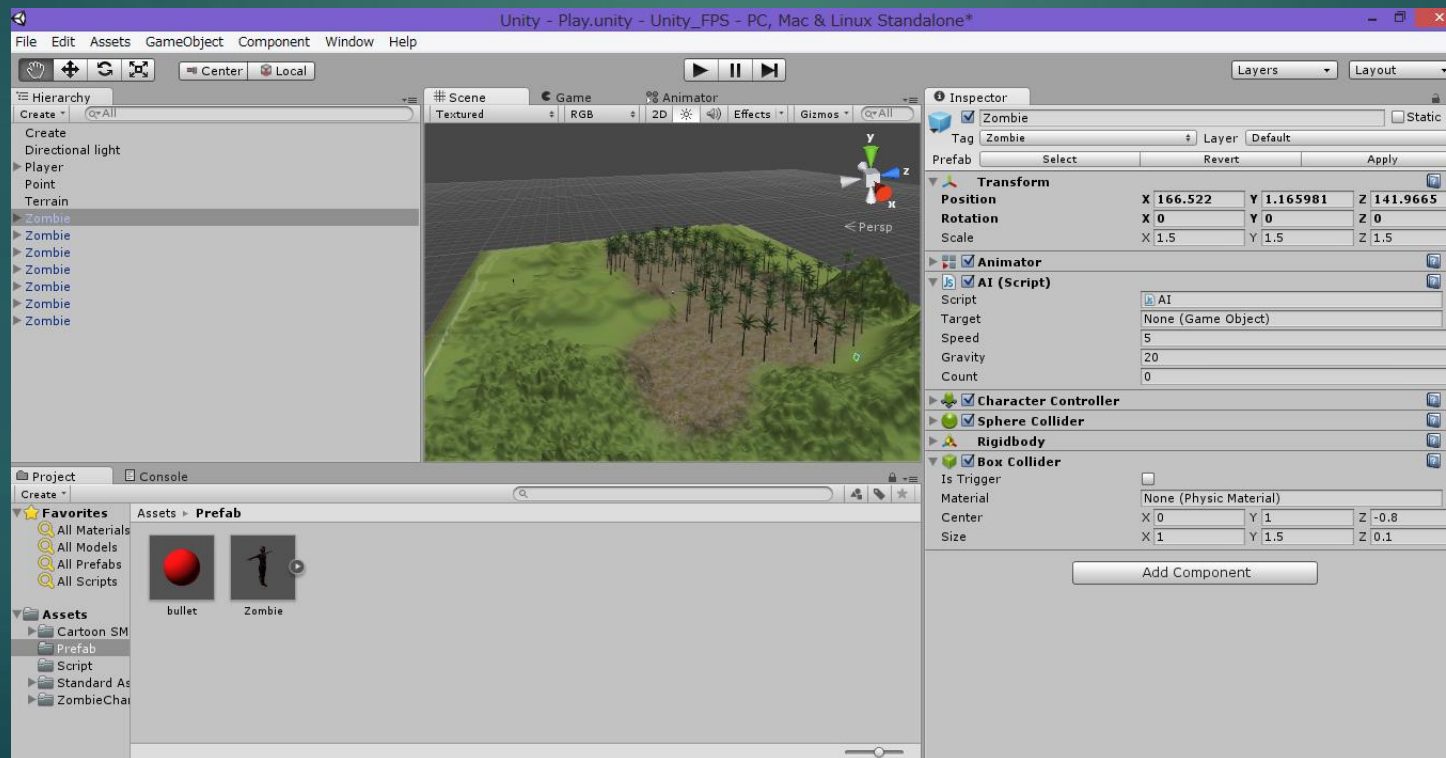
Screen resolutionは1024×768がいいと思います。

Buildする前とBuildした後のゲームの状態がたまに変わってるときがある
ので、そこは微調整してくださいm(_ _)m



完成

これでゲームは一応完成！最初にゾンビをたくさん配置したりすると難易度があがったりするのであとは各々でいじってください。



間違ってる箇所があったらすいませんm(_ _)m