

Enemy.cs

```
1 using UnityEngine;
2
3 public class Enemy : MonoBehaviour
4 {
5     [Header("1秒ごとのダメージ量")]
6     public float Damage = 0.3f;
7 }
8
```

EnemyGenerator.cs

```
1 using UnityEngine;
2
3 public class EnemyGenerator : MonoBehaviour
4 {
5     [Header("敵のPrefab")]
6     [SerializeField] private GameObject enemyPrefab;
7
8     [Header("横スクロールのリミット位置")]
9     [SerializeField] private float limit;
10
11     [Header("敵の配置スパン")]
12     [SerializeField] private float minSpan;
13     [SerializeField] private float maxSpan;
14
15     void Awake()
16     {
17         //初期位置設定
18         Vector3 pos = new Vector3(10f, 0, 0);
19
20         //設定したxのリミット位置まで敵を配置する
21         while (pos.x <= limit)
22         {
23             //敵を生成
24             GameObject obj = Instantiate(enemyPrefab, transform);
25
26             //地面か空中のどちらに配置するか設定
27             pos.y = 0.4f;
28             if(Random.Range(0, 2) == 1)
29             {
30                 pos.y = 2.0f;
31             }
32             obj.transform.position = pos;
33
34             //横にずらす
35             pos.x += Random.Range(minSpan, maxSpan);
36         }
37     }
38 }
39
```

Player.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 #if STATE_LV1
6 public partial class Player : MonoBehaviour
7 {
8     private enum State
9     {
10         Standing, //通常
11         Squat, //しゃがみ
12         Jump, //ジャンプ
13         Dive //ダイブ
14     };
15     State PlayState;
16
17     //Player_Common.csからStartが呼ばれる
18     void OnStart()
19     {
20         PlayState = State.Standing;
21     }
22
23     //Player_Common.csからUpdateが呼ばれる
24     void OnUpdate()
25     {
26         switch(PlayState)
27         {
28             case State.Standing:
29                 //座り判定
30                 if (Input.GetKey(KeyCode.DownArrow))
31                 {
32                     Squat();
33                     PlayState = State.Squat;
34                 }
35
36                 //ジャンプ判定
37                 if (Input.GetKey(KeyCode.Space))
38                 {
39                     Jump();
40                     PlayState = State.Jump;
41                 }
42                 break;
43
44             case State.Squat:
45                 //立っている判定
46                 if (!Input.GetKey(KeyCode.DownArrow))
47                 {
48                     EndSquat();
49                     PlayState = State.Standing;
50                 }
51                 break;
```

```
52
53     case State.Jump:
54         //ダイブ判定
55         if (Input.GetKey(KeyCode.DownArrow))
56             {
57                 Dive();
58                 PlayState = State.Dive;
59             }
60         break;
61
62     case State.Dive:
63         break;
64     }
65 }
66
67 private void OnCollisionEnter(Collision collision)
68 {
69     PlayState = State.Standing;
70 }
71 }
72 #endif
73
74
```

Player_Common.cs

```
1 using UnityEngine;
2
3 // ステート以外の処理部分
4 public partial class Player : MonoBehaviour
5 {
6     public bool IsDead;
7     public float HP;
8
9     [Header("スクロール速度")]
10    [SerializeField] private float moveSpeed;
11
12    [Header("死んだ時にRigidbodyに与える力")]
13    [SerializeField] private Vector3 deathAddForce = Vector3.zero;
14    [SerializeField] private Vector3 deathAddTorque = Vector3.zero;
15
16    private Vector3 defaultScale;
17    private Material materialInstance;
18    private Rigidbody rb;
19
20    private void Start()
21    {
22        //初期値設定、コンポーネント取得
23        HP = 1.0f;
24        defaultScale = transform.localScale;
25        rb = GetComponent<Rigidbody>();
26        materialInstance = GetComponent<MeshRenderer>().material;
27
28        OnStart();
29    }
30
31    private void Update()
32    {
33        // 自動で右方向に移動
34        transform.position += Vector3.right * moveSpeed * Time.deltaTime;
35
36        OnUpdate();
37    }
38
39    //ジャンプ
40    private void Jump()
41    {
42        rb.AddForce(Vector3.up * 7f, ForceMode.Impulse);
43    }
44
45    //素早く地面に接地
46    private void AdjustGround()
47    {
48        var pos = transform.position;
49        pos.y = transform.localScale.y * 0.5f;
50        transform.position = pos;
51    }
```

```
52
53 //しゃがみ
54 private void Squat()
55 {
56     transform.localScale = Vector3.one;
57     AdjustGround();
58 }
59
60 //立つ
61 private void EndSquat()
62 {
63     transform.localScale = defaultScale;
64 }
65
66 //ダイブ
67 private void Dive()
68 {
69     rb.AddForce(Vector3.down * 14f, ForceMode.Impulse);
70 }
71
72 //敵と当たった時
73 private void OnTriggerEnter(Collider other)
74 {
75     materialInstance.color = Color.red;
76 }
77
78 //敵と当たり続けている時
79 private void OnTriggerStay(Collider other)
80 {
81     if (other.TryGetComponent(out Enemy enemy))
82     {
83         HP -= enemy.Damage * Time.deltaTime;
84
85         if(HP <= 0)
86         {
87             DeathAction();
88         }
89     }
90 }
91
92 //敵と離れた時
93 private void OnTriggerExit(Collider other)
94 {
95     materialInstance.color = Color.blue;
96 }
97
98 // 死亡した時に呼ばれる
99 private void DeathAction()
100 {
101     IsDead = true;
102
103     // 動作停止
104     GetComponent<Collider>().enabled = false;
105
106     // 吹っ飛ばす
```

```
107     rb.constraints = RigidbodyConstraints.FreezePositionZ | RigidbodyConstraints.FreezeRotationX;
108     rb.AddForce(deathAddForce, ForceMode.Impulse);
109     rb.AddTorque(deathAddTorque, ForceMode.Impulse);
110 }
111 }
112
```

CameraMove.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class CameraMove : MonoBehaviour
6 {
7     private Player player = null;
8
9     private void Start()
10    {
11        player = GameObject.FindWithTag("Player").GetComponent<Player>();
12    }
13
14    private void Update()
15    {
16        if (player.IsDead)
17        {
18            return;
19        }
20
21        var pos = transform.position;
22        pos.x = player.transform.position.x;
23        transform.position = pos;
24    }
25 }
26
```

HPGauge.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class HPGauge : MonoBehaviour
7 {
8     private Player player;
9
10    // Start is called before the first frame update
11    void Start()
12    {
13        player = GameObject.FindWithTag("Player").GetComponent<Player>();
14    }
15
16    // Update is called once per frame
17    void Update()
18    {
19        GetComponent<Image>().fillAmount = player.HP;
20    }
21 }
22
```

PlayerStateBase.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 #if STATE_LV2
6 public abstract class PlayerStateBase
7 {
8     //ステータス開始時
9     public virtual void OnEnter(Player owner, PlayerStateBase prevState) { }
10
11     //ステータス中の判定
12     public virtual void OnUpdate(Player owner) { }
13
14     //ステータス終了判定
15     public virtual void OnExit(Player owner, PlayerStateBase nextState) { }
16 }
17 #endif
18
19
```

PlayerStateDive.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 #if STATE_LV2
6 public partial class Player
7 {
8     public class StateDive : PlayerStateBase
9     {
10         public override void OnEnter(Player owner, PlayerStateBase prevState)
11         {
12             owner.Dive();
13         }
14     }
15 }
16 #endif
```

PlayerStateJump.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 #if STATE_LV2
6 public partial class Player
7 {
8     public class StateJump : PlayerStateBase
9     {
10         public override void OnEnter(Player owner, PlayerStateBase prevState)
11         {
12             owner.Jump();
13         }
14
15         public override void OnUpdate(Player owner)
16         {
17             if (Input.GetKeyDown(KeyCode.DownArrow))
18             {
19                 owner.ChangeState(stateDive);
20             }
21         }
22     }
23 }
24 #endif
```

PlayerStateSquat.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 #if STATE_LV2
6 public partial class Player
7 {
8     public class StateSquat : PlayerStateBase
9     {
10         public override void OnEnter(Player owner, PlayerStateBase prevState)
11         {
12             owner.Squat();
13         }
14
15         public override void OnUpdate(Player owner)
16         {
17             if (!Input.GetKey(KeyCode.DownArrow))
18             {
19                 owner.ChangeState(stateStanding);
20             }
21         }
22
23         public override void OnExit(Player owner, PlayerStateBase nextState)
24         {
25             owner.EndSquat();
26         }
27     }
28 }
29 #endif
```

PlayerStateStanding.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 #if STATE_LV2
6 public partial class Player
7 {
8     public class StateStanding : PlayerStateBase
9     {
10         public override void OnUpdate(Player owner)
11         {
12             if (Input.GetKey(KeyCode.DownArrow))
13             {
14                 owner.ChangeState(stateSquat);
15             }
16
17             if (Input.GetKey(KeyCode.Space))
18             {
19                 owner.ChangeState(stateJump);
20             }
21         }
22     }
23 }
24 #endif
```

Player.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 #if STATE_LV2
6 public partial class Player : MonoBehaviour
7 {
8     //ステータスをクラスで定義
9     private static StateStanding stateStanding = new StateStanding();
10    private static StateSquat stateSquat = new StateSquat();
11    private static StateJump stateJump = new StateJump();
12    private static StateDive stateDive = new StateDive();
13
14    private PlayerStateBase currentState;
15
16    //Player_Common.csからStartが呼ばれる
17    void OnStart()
18    {
19        //初期ステータス設定
20        currentState = stateStanding;
21        currentState.OnEnter(this, null);
22    }
23
24    //Player_Common.csからUpdateが呼ばれる
25    void OnUpdate()
26    {
27        //今のステータスのUpdateを呼ぶ
28        currentState.OnUpdate(this);
29    }
30
31    // ステート変更
32    private void ChangeState(PlayerStateBase nextState)
33    {
34        //今のステータスの終了処理をして、次のステータスの開始処理をする
35        currentState.OnExit(this, nextState);
36        nextState.OnEnter(this, currentState);
37        currentState = nextState;
38    }
39
40    private void OnCollisionEnter(Collision collision)
41    {
42        //立ち判定
43        ChangeState(stateStanding);
44    }
45 }
46 #endif
47
```