

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class PlayerController : MonoBehaviour
7  {
8      public AudioSource audioSource;
9      public AudioClip jump, attack, getItem;
10     public Text timeLimitText, itemCountText;
11     public GameManager gameManager;
12     Rigidbody2D rigidbody2D;
13     Animator animator;
14
15     public float moveSpeed;
16     public float jumpPower;
17
18     float vx; // プレイヤーのxの速度
19     float vy; // プレイヤーのyの速度
20
21     public bool isGrounded;
22     bool canMove = true;
23
24     public float timeLimit;
25     int itemCount;
26
27     // ゲームが再生されたときに、一度だけ実行される
28     void Start()
29     {
30         rigidbody2D = this.GetComponent<Rigidbody2D>();
31         animator = this.GetComponent<Animator>();
32     }
33
34     // スタート関数が実行された後、繰り返し実行される
35     void Update()
36     {
37
38
39         if (canMove)
40         {
41             CountDown();
```

```
42         //プレイヤーの移動処理
43         MovePlayer();
44
45         // アニメーションの切り替え
46         ChangeAnimation();
47     }
48 }
49
50 private void LateUpdate()
51 {
52     CameraScroll();
53 }
54
55 void CameraScroll()
56 {
57     Vector3 cameraPos = Camera.main.transform.position;
58     cameraPos.x = Mathf.Clamp(this.transform.position.x, 0, 19);
59     Camera.main.transform.position = cameraPos;
60 }
61
62 void Countdown()
63 {
64     if (timeLimit >= 0)
65     {
66         timeLimit -= Time.deltaTime;
67         timeLimitText.text = Mathf.Ceil(timeLimit).ToString();
68     }
69     else
70     {
71         GameOver();
72     }
73 }
74
75 //プレイヤーの移動処理
76 void MovePlayer()
77 {
78     if (Input.GetKey("right"))//右が押されている時
79     {
80         vx = moveSpeed;
81         this.transform.localScale = new Vector3(1, 1, 1);
82     }
```

```

83         else if (Input.GetKey("left"))//左が押されている時
84         {
85             vx = -moveSpeed;
86             this.transform.localScale = new Vector3(-1, 1, 1);
87         }
88         else//右も左も押されていない時
89         {
90             vx = 0;
91         }
92
93         if (Input.GetKeyDown("space") && isGrounded)//スペースが押された時、地面の上にいたら
94         {
95             Jump(jumpPower);
96             audioSource.PlayOneShot(jump);
97         }
98         else
99         {
100             vy = rigidbody2D.velocity.y;
101         }
102
103         rigidbody2D.velocity = new Vector2(vx, vy);
104     }
105
106     // アニメーションの切り替え
107     void ChangeAnimation()
108     {
109         if (isGrounded)//地面の上にいる時
110         {
111             if (vx == 0)//止まっている時
112             {
113                 animator.Play("IdleAnimation");
114             }
115             else//動いている時
116             {
117                 animator.Play("RunAnimation");
118             }
119         }
120         else//空中にいる時
121         {
122             if (vy > 0)//上に移動している時
123             {

```

```
124         animator.Play("JumpAnimation");
125     }
126     else//落ちている時
127     {
128         animator.Play("FallAnimation");
129     }
130 }
131 }
132 void Jump(float jumpPower)
133 {
134     vy = jumpPower;
135     rigidbody2D.velocity = new Vector2(vx, vy);
136 }
137
138
139 private void OnCollisionEnter2D(Collision2D collision)
140 {
141     if (collision.gameObject.tag == "Chicken")
142     {
143         //上から踏んだら
144         if (this.transform.position.y > collision.transform.position.y + 0.7f)
145         {
146             audioSource.PlayOneShot(attack);
147             ChickenController chickenController;
148             chickenController = collision.gameObject.GetComponent<ChickenController>();
149             chickenController.PlayHitAnimation();
150             Jump(jumpPower / 2);
151         }
152         //横から当たった時
153         else
154         {
155             GameOver();
156         }
157     }
158 }
159
160 private void OnTriggerEnter2D(Collider2D collision)
161 {
162     if (collision.name == "Goal")
163     {
164         if (itemCount == 5)
```

```

165         {
166             GameClear();
167         }
168     }
169
170     if (collision.tag == "Item")
171     {
172         audioSource.PlayOneShot(getItem);
173         Destroy(collision.gameObject);
174         itemCount++;
175         itemCountText.text = itemCount + " / 5";
176     }
177
178     if (collision.tag == "Trap")
179     {
180         GameOver();
181     }
182
183     if (collision.tag == "Stage")
184     {
185         isGrounded = true;
186     }
187 }
188
189 private void OnTriggerExit2D(Collider2D collision)
190 {
191     if (collision.name == "Stage")
192     {
193         isGrounded = false;
194     }
195 }
196
197 public void DestroyThisGameObject()
198 {
199     Destroy(this.gameObject);
200 }
201
202 void GameOver()
203 {
204     gameManager.GameOver();
205     canMove = false;

```

```
206         Destroy(this.rigidbody2D);
207         Destroy(this.GetComponent<BoxCollider2D>());
208         animator.Play("DeathAnimation");
209     }
210
211     void GameClear()
212     {
213         gameManager.GameClear();
214         canMove = false;
215         Destroy(this.rigidbody2D);
216         Destroy(this.GetComponent<BoxCollider2D>());
217     }
218 }
```