

PlayerController.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PlayerController : MonoBehaviour
6 {
7     [SerializeField] LayerMask BlockLayer;
8     private enum DIRECTION_TYPE
9     {
10         DEFAULT,
11         RIGHT,
12         LEFT
13     }
14     private DIRECTION_TYPE dir;
15     private Rigidbody2D rb2;
16     private float _speed;
17     private Animator anim;
18     private float Speed;
19     private float JumpPower;
20
21     // Start is called before the first frame update
22     protected virtual void Start()
23     {
24         //向きの設定、初期はデフォルト状態
25         dir = DIRECTION_TYPE.DEFAULT;
26
27         //コンポーネント取得
28         rb2 = GetComponent<Rigidbody2D>();
29         anim = GetComponent<Animator>();
30
31         //移動とジャンプ値を設定
32         Speed = 5.0f;
33         JumpPower = 500.0f;
```

```
34     }
35
36     // Update is called once per frame
37     protected virtual void Update()
38     {
39         InputKey();
40         CheckJump();
41     }
42
43     void FixedUpdate()
44     {
45         Move();
46     }
47
48     private void InputKey()
49     {
50         //キーの入力を取得し、アニメーションに反映
51         float x = Input.GetAxisRaw("Horizontal");
52         anim.SetFloat("AnimSpeed", Mathf.Abs(x));
53
54         if( x == 0)
55         {
56             //向きはそのまま
57             dir = DIRECTION_TYPE.DEFAULT;
58         }
59         else if( x > 0)
60         {
61             //右向き
62             dir = DIRECTION_TYPE.RIGHT;
63         }
64         else
65         {
66             //左向き
67             dir = DIRECTION_TYPE.LEFT;
68         }
69     }
70 }
```

```
69     }
70
71     private void CheckJump()
72     {
73         //地面に設置していたら
74         if(IsGround())
75         {
76             //スペース押されていたらジャンプ処理
77             if(Input.GetButtonDown("Jump"))
78             {
79                 rb2.AddForce(Vector2.up * JumpPower);
80             }
81         }
82     }
83
84     private void Move()
85     {
86         switch(dir)
87         {
88             case DIRECTION_TYPE.DEFAULT:
89                 //スピード無し
90                 _speed = 0;
91                 break;
92             case DIRECTION_TYPE.RIGHT:
93                 //右に移動
94                 _speed = Speed;
95                 transform.localScale = new Vector3(2,2,2);
96                 break;
97             case DIRECTION_TYPE.LEFT:
98                 //左に移動
99                 _speed = -(Speed);
100                transform.localScale = new Vector3(-2,2,2);
101                break;
102             default:
103                 //スピード無し
```

```
104     _speed = 0;
105     break;
106 }
107
108 //移動処理
109 rb2.velocity = new Vector2(_speed, rb2.velocity.y);
110 }
111
112 private bool IsGround()
113 {
114     bool isBlock = false;
115     Vector3 rightStartPoint = transform.position + Vector3.right * 0.4f;
116     Vector3 leftStartPoint = transform.position - Vector3.right * 0.4f;
117     Vector3 endPoint = transform.position - Vector3.up * 0.2f;
118     Debug.DrawLine(rightStartPoint , endPoint);
119     Debug.DrawLine(leftStartPoint , endPoint);
120
121     //地面に当たっているか判定
122     if( Physics2D.Linecast(rightStartPoint , endPoint, BlockLayer) ||
123         Physics2D.Linecast(leftStartPoint , endPoint ,BlockLayer) )
124     {
125         isBlock = true;
126     }
127
128     return isBlock;
129 }
130 }
131
```

PlayerController2.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PlayerController2 : PlayerController
6 {
7     // Start is called before the first frame update
8     protected override void Start()
9     {
10         base.Start();
11     }
12
13     // Update is called once per frame
14     protected override void Update()
15     {
16         base.Update();
17     }
18 }
19
```

PlayerController.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PlayerController : MonoBehaviour
6 {
7     [SerializeField] LayerMask BlockLayer;
8     private enum DIRECTION_TYPE
9     {
10         DEFAULT,
11         RIGHT,
12         LEFT
13     }
14     private DIRECTION_TYPE dir;
15     private Rigidbody2D rb2;
16     private float _speed;
17     private Animator anim;
18     protected float Speed;
19     protected float JumpPower;
20
21     // Start is called before the first frame update
22     protected virtual void Start()
23     {
24         //向きの設定、初期はデフォルト状態
25         dir = DIRECTION_TYPE.DEFAULT;
26
27         //コンポーネント取得
28         rb2 = GetComponent<Rigidbody2D>();
29         anim = GetComponent<Animator>();
30
31         //移動とジャンプ値を設定
32         Speed = 5.0f;
33         JumpPower = 500.0f;
```

```
34 }
35
36 // Update is called once per frame
37 protected virtual void Update()
38 {
39     InputKey();
40     CheckJump();
41 }
42
43 void FixedUpdate()
44 {
45     Move();
46 }
47
48 private void InputKey()
49 {
50     //キーの入力を取得し、アニメーションに反映
51     float x = Input.GetAxisRaw("Horizontal");
52     anim.SetFloat("AnimSpeed", Mathf.Abs(x));
53
54     if( x == 0)
55     {
56         //向きはそのまま
57         dir = DIRECTION_TYPE.DEFAULT;
58     }
59     else if( x > 0)
60     {
61         //右向き
62         dir = DIRECTION_TYPE.RIGHT;
63     }
64     else
65     {
66         //左向き
67         dir = DIRECTION_TYPE.LEFT;
68     }
```

```
69     }
70
71     private void CheckJump()
72     {
73         //地面に設置していたら
74         if(IsGround())
75         {
76             //スペース押されていたらジャンプ処理
77             if(Input.GetButtonDown("Jump"))
78             {
79                 rb2.AddForce(Vector2.up * JumpPower);
80             }
81         }
82     }
83
84     private void Move()
85     {
86         switch(dir)
87         {
88             case DIRECTION_TYPE.DEFAULT:
89                 //スピード無し
90                 _speed = 0;
91                 break;
92             case DIRECTION_TYPE.RIGHT:
93                 //右に移動
94                 _speed = Speed;
95                 transform.localScale = new Vector3(2,2,2);
96                 break;
97             case DIRECTION_TYPE.LEFT:
98                 //左に移動
99                 _speed = -(Speed);
100                transform.localScale = new Vector3(-2,2,2);
101                break;
102            default:
103                //スピード無し
```

```
104     _speed = 0;
105     break;
106 }
107
108 //移動処理
109 rb2.velocity = new Vector2(_speed, rb2.velocity.y);
110 }
111
112 private bool IsGround()
113 {
114     bool isBlock = false;
115     Vector3 rightStartPoint = transform.position + Vector3.right * 0.4f;
116     Vector3 leftStartPoint = transform.position - Vector3.right * 0.4f;
117     Vector3 endPoint = transform.position - Vector3.up * 0.2f;
118     Debug.DrawLine(rightStartPoint , endPoint);
119     Debug.DrawLine(leftStartPoint , endPoint);
120
121     //地面に当たっているか判定
122     if( Physics2D.Linecast(rightStartPoint , endPoint, BlockLayer) ||
123         Physics2D.Linecast(leftStartPoint , endPoint ,BlockLayer) )
124     {
125         isBlock = true;
126     }
127
128     return isBlock;
129 }
130 }
131
```

PlayerController2.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PlayerController2 : PlayerController
6 {
7     // Start is called before the first frame update
8     protected override void Start()
9     {
10         base.Start();
11     }
12
13     // Update is called once per frame
14     protected override void Update()
15     {
16         base.Update();
17
18         //左シフトキーを押すとSpeedの値を変更する
19         if (Input.GetKey(KeyCode.LeftShift))
20         {
21             Speed = 10.0f;
22         }
23         else
24         {
25             Speed = 5.0f;
26         }
27
28         //右シフトキーを押すとJumpPowerの値を変更する
29         if (Input.GetKey(KeyCode.RightShift))
30         {
31             JumpPower = 1000.0f;
32         }
33         else
```

```
34 {  
35     JumpPower = 500.0f;  
36 }  
37 }  
38 }  
39
```