

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class PlayerMovement : MonoBehaviour {
5     private Rigidbody2D player;
6     private PlayerMovement playerScript;
7     private BoxCollider2D ground;
8     public Animator playerAnim;
9     public Transform playerPos;
10    public bool facingRight;
11    public bool isJumping;
12    public bool grounded;
13    public bool attack;
14    public int lives = 3;
15    public bool inFrontOfDoor = false;
16
17    public float xValue;
18    public float yValue;
19    public float zValue;
20    // Use this for initialization
21    void Start () {
22        player = GetComponent<Rigidbody2D>();
23        playerAnim = GetComponent<Animator>();
24        playerScript = GetComponent<PlayerMovement>();
25        ground = GameObject.Find("isGrounded").GetComponent<BoxCollider2D>();
26        facingRight = true;
27        isJumping = false;
28        Vector3 playerPos = transform.localPosition;
29        this.xValue = playerPos.x; //Current x position of player. Set when level ↗
30        //begins. Used in DeathZone script
31        this.yValue = playerPos.y; //Current y position of player. Set when level ↗
32        //begins. Used in DeathZone script
33        this.zValue = playerPos.z; //Current z position of player. Set when level ↗
34        //begins. Used in DeathZone script
35    }
36    // Update is called once per frame
37    void Update () {
38        float horizontalMovement = Input.GetAxis("Horizontal");
39        //Horizontal means left or right arrow keys or the "a" and "d" keys.
40        //Left arrow and "a" set the float to -1
41        //Right arrow and "d" set the float to 1
42        checkAttack();
43        playerMove(horizontalMovement * 5);
44        Speed(horizontalMovement);
45        flipPlayer(horizontalMovement);
46        playerJump();
47        checkGrounded();
48        checkAlive();
49    }
50
51    private void checkAlive() //Checks if the player is alive
52    {
```

```
50     if(lives <= 0)
51     {
52         playerAnim.SetBool("dead", true); //Runs the death animation
53         StartCoroutine(deathDelay());
54     }
55 }
56 private void playerJump()
57 {
58     if (Input.GetKeyDown("up") && !isJumping) //Allows player to jump as long as
59         //as he is not already jumping
60     {
61         player.AddForce(Vector2.up * 540);
62     }
63 private void checkGrounded()
64 {
65     if (grounded)
66     {
67         playerAnim.SetBool("jump", false); //Stop running the jump animation
68         //if the player is on the ground
69     }
70     if (!grounded)
71     {
72         playerAnim.SetBool("jump", true); //Start running the jump animation
73         //if the player is in the air
74     }
75 }
76 private void Speed(float h)
77 {
78     playerAnim.SetFloat("speed", Mathf.Abs(h));
79     //h is the float set by Input.GetAxis above in the Update method
80     //This sets the variable "speed" in the animator in the Unity IDE to the
81     //absolute value of the speed
82     //If the value is greater than 1, the run animation plays. Otherwise, the
83     //idle animation plays
84 }
85 private void playerMove(float h)
86 {
87     if (lives > 0)
88     {
89         player.velocity = new Vector2(h * 1.5f, player.velocity.y);
90         //Sets player speed on the x and the y axis
91     }
92 }
93 private void flipPlayer(float h)
94 {
95     if ((h > 0 && !facingRight) || (h < 0 && facingRight)) //Flips the player
96         //so that he faces the direction he's going
97     {
98         facingRight = !facingRight;
99         Vector3 playerFlip = transform.localScale;
100         playerFlip.x *= -1;
```

```
96         transform.localScale = playerFlip;
97     }
98 }
99 private void checkAttack()
100 {
101     if(Input.GetKeyDown("space") && !playerAnim.GetBool("attack"))
102     {
103         playerAnim.SetBool("attack", true); //Runs attack animation
104         attack = true; //Sets attack to true for use in other scripts
105         StartCoroutine(Delay());
106     }
107 }
108 IEnumerator Delay()
109 {
110     yield return new WaitForSeconds(0.8f); //Waits 0.8 seconds then runs the ↗
111         next line of code
112     playerAnim.SetBool("attack", false); //Stop the attack animation
113     attack = false; //Sets attack to false
114 }
115 public void deathByVoid()
116 {
117     lives -= 1; //Player loses a life
118     transform.localPosition = new Vector3(xValue,yValue,zValue);
119     //Teleports player to his beginning position
120 }
121 public void attackByZombie()
122 {
123     GameObject Shield = GameObject.Find("shieldIcon");
124     ShieldIcon shieldScript = Shield.GetComponent<ShieldIcon>();
125     if(shieldScript.pickup)
126     {
127         shieldScript.pickup = false; //Removes shield if a shield is attached
128     } else
129     {
130         lives -= 1; //If there is no shield, then the player loses a life
131     }
132 }
133 public void attackByBoss()
134 {
135     GameObject Shield = GameObject.Find("shieldIcon");
136     ShieldIcon shieldScript = Shield.GetComponent<ShieldIcon>();
137     if (shieldScript.pickup)
138     {
139         shieldScript.pickup = false; //Removes shield if a the player has ↗
140             shield
141         lives -= 2; //Player loses 2 lives
142     }
143     else
144     {
145         lives -= 3; //Player loses 3 lives
146     }
147 }
```

```
146     }
147     IEnumerator deathDelay()
148     {
149         GameObject title = GameObject.Find("TitleScreen");
150         LevelComplete titlescript = title.GetComponent<LevelComplete>();
151         Destroy(ground);
152         grounded = false; //Prevents player from jumping when dead
153         yield return new WaitForSeconds(1.5f); //Wait 1.5 seconds
154         playerAnim.SetBool("dead", false); //Stops the death animation
155         Application.LoadLevel(titlescript.titleScreen); //Loads the title screen ↗
156         //with levels that the player has unlocked
157     }
158     public void levelComplete()
159     {
160         GameObject title = GameObject.Find("TitleScreen");
161         LevelComplete titlescript = title.GetComponent<LevelComplete>();
162         titlescript.levelComplete(); //Runs the levelComplete method in the ↗
163         //LevelComplete script
164     }
165     public void addKillStat()
166     {
167         GameObject stat = GameObject.Find("KStat");
168         zombiekstat zstat = stat.GetComponent<zombiekstat>();
169         zstat.zombieDeadNum += 1; //Adds a zombie kill statistic
170     }
171 }
```