```
 1  using UnityEngine;
 2
 3  public abstract class Movement : MonoBehaviour
 4  {
 5      // This controls the horizontal / vertical movement of all players
 6      protected float Speed;
 7      protected float JumpHeight;
 8      private bool OnGround = false;
 9      private bool LeftCheck = false;
10      private bool RightCheck = false;
11      public Master M;
12      public string Name;
13
14      protected virtual void Start()
15      {
16          // Initialises the player object
17          M.InitialisePlayer(Name);
18
19          Speed = M.GetSpeed();
20          JumpHeight = M.GetJumpHeight();
21      }
22
23      public void Update()
24      {
25          // Gets an array containing the moves
26          // [0] = whether or not the player is jumping
27          // [1] = whether the player is moving left / right / not at all
28          int[] move = SelectedMove();
29
30          Jump(move[0]);
31
32          HorizontalMove(move[1]);
33
34          // Checks if the player has died
35          if (M.CheckDeath(Name, transform.position.y))
36          {
37              gameObject.SetActive(false);
38          }
39      }
40
41      public void TouchGround()
42      {
43          OnGround = true;
44      }
45
46      public void LeaveGround()
47      {
48          OnGround = false;
49      }
50
51      public void SideCheckInput (bool left, bool value)
52      {
53          if (left)
```

```
54          {
55              LeftCheck = value;
56          }
57          else
58          {
59              RightCheck = value;
60          }
61      }
62
63      private void HorizontalMove(int direction)
64      {
65          Vector3 movement;
66
67          float horizontal = direction;
68
69          // Checks to see if there is anything left or right in order to
              prevent jittering
70          if (horizontal < 0 && LeftCheck == false)
71          {
72              horizontal = Time.deltaTime * Speed * -1;
73
74              // If the player is in the air, they move slower
75              if (OnGround)
76              {
77                  horizontal *= 1.2f;
78              }
79              // Declares how far the player will move
80              movement = new Vector2(horizontal, 0f);
81              // Resolves the movement
82              M.MovePlayer(Name, horizontal);
83          }
84          else if (horizontal > 0 && RightCheck == false)
85          {
86              horizontal = Time.deltaTime * Speed;
87              if (OnGround)
88              {
89                  horizontal *= 1.2f;
90              }
91              movement = new Vector2(horizontal, 0f);
92              M.MovePlayer(Name, horizontal);
93          }
94          else
95          {
96              movement = new Vector2(0f, 0f);
97          }
98
99          // Moves the player by that amount
100         transform.position += movement;
101     }
102
103     private void Jump(int jump)
104     {
105         var body = gameObject.GetComponent<Rigidbody2D>();
```

```
106         var vel = body.velocity.y;
107
108         // Checks that the player is on the ground and wants to jump and
              isnt already jumping (this is due to a weird bug)
109         if (jump == 1 && OnGround && vel <= 0 && OnGround)
110         {
111             // Adds an upwards force to create a realistic jump
112             body.AddForce(new Vector2(0f, JumpHeight), ForceMode2D.Impulse);
113             LeaveGround();
114         }
115     }
116
117     protected abstract int[] SelectedMove();
118 }
```