```
 1  using System.Collections;
 2  using System.Collections.Generic;
 3  using UnityEngine;
 4  using UnityEngine.Events;
 5
 6  public class DogMovement : MonoBehaviour
 7  {
 8      [SerializeField]
 9      float forwardSpeed;
10
11      //[SerializeField]
12      //float backwardSpeed;
13
14      //[SerializeField]
15      //float strafeSpeed;
16
17      [SerializeField]
18      float sprintMultiplier = 2f;
19
20      Rigidbody rigBod;
21
22      bool isMoving = false;
23
24      [SerializeField]
25      float sprintTime = 5f;
26
27      float sprintTimer;
28
29      [SerializeField]
30      float sprintCooldownTime = 15f;
31
32      [SerializeField]
33      float rayLength = 1f;
34
35      float sprintCooldownTimer;
36
37      Vector3 DownwardForce = new Vector3(0, -10, 0);
38
39      bool isGrounded;
40
41
42      // Start is called before the first frame update
43      void Start()
44      {
45          rigBod = GetComponent<Rigidbody>();
46
47          sprintTimer = sprintTime;
48          sprintCooldownTimer = sprintCooldownTime;
49      }
```

```
50
51      // Update is called once per frame
52      void Update()
53      {
54          followCursor();
55
56          //Check to see if the Dog should be moving
57          if (Input.GetKeyUp(KeyCode.W))
58          {
59              rigBod.velocity = Vector3.zero;
60          }
61
62          isGrounded = GroundCheck();
63          //Debug.Log(isGrounded);
64
65      }
66
67      private void FixedUpdate()
68      {
69
70          //Move Forward
71          if (Input.GetKey(KeyCode.W))
72          {
73              moveForward(forwardSpeed);
74              isMoving = true;
75          }
76          //Sprint
77          if (Input.GetKey(KeyCode.W) && Input.GetKey(KeyCode.LeftShift) &&
              sprintTimer > 0f)
78          {
79              sprint(forwardSpeed);
80              sprintTimer -= Time.deltaTime;
81          }
82
83          if (!Input.GetKey(KeyCode.LeftShift) && sprintTimer < sprintTime)
84          {
85              sprintCooldownTimer -= Time.deltaTime;
86          }
87
88          if (sprintCooldownTimer <= 0f)
89          {
90              sprintTimer = sprintTime;
91              sprintCooldownTimer = sprintCooldownTime;
92          }
93
94          ////Move Backward
95          //if (Input.GetKey(KeyCode.S))
96          //{
97          //    moveBackward(backwardSpeed);
```

```
 98          //    isMoving = true;
 99          //}
100
101          ////Strafe Left
102          //if (Input.GetKey(KeyCode.A))
103          //{
104          //    strafeLeft(strafeSpeed);
105          //    isMoving = true;
106          //}
107
108          ////Strafe Right
109          //if (Input.GetKey(KeyCode.D))
110          //{
111          //    strafeRight(strafeSpeed);
112          //    isMoving = true;
113          //}
114
115          // Stop the rigidbody from moving if no buttons are pressed
116          if (!isMoving)
117          {
118              rigBod.velocity = Vector3.zero;
119          }
120          if (!isGrounded)
121          {
122              Vector3 newVelocity = new Vector3(rigBod.velocity.x,
                   rigBod.velocity.y - 2f, rigBod.velocity.z);
123              rigBod.velocity = newVelocity;
124          }
125
126      }
127
128      /** Make Dag follow the player's cursor*/
129      void followCursor()
130      {
131          //Get mouse position
132          Vector3 mousePos = Input.mousePosition;
133
134          //Convert mouse position to a point in the world
135          Ray ray = Camera.main.ScreenPointToRay(mousePos);
136          RaycastHit hit;
137
138          if (Physics.Raycast(ray, out hit))
139          {
140              //Set targetPos to mouse position point in world
141              Vector3 targetPos = hit.point;
142
143              // Rotate Dag
144              transform.LookAt(targetPos);
145
```

```
146             //Block x and z rotations
147             transform.eulerAngles = new Vector3(0,
                  transform.eulerAngles.y, 0);
148         }
149     }
150
151     void moveForward(float forwardSpeed)
152     {
153         rigBod.velocity = (transform.forward - (.1f * transform.up)) *
              forwardSpeed;
154     }
155
156     void moveBackward(float backwardSpeed)
157     {
158         rigBod.velocity = -transform.forward * backwardSpeed;
159     }
160
161     void strafeLeft(float strafeSpeed)
162     {
163         Vector3 left = new Vector3(-1, 0, 0);
164         rigBod.velocity = left * strafeSpeed;
165     }
166
167     void strafeRight(float strafeSpeed)
168     {
169         Vector3 right = new Vector3(1, 0, 0);
170         rigBod.velocity = right * strafeSpeed;
171     }
172
173     void sprint(float forwardSpeed)
174     {
175         rigBod.velocity = transform.forward * forwardSpeed *
              sprintMultiplier;
176     }
177
178     bool GroundCheck()
179     {
180         RaycastHit hit;
181
182         if (Physics.Raycast(transform.position, Vector3.down, out hit,
              rayLength))
183         {
184             return true;
185         }
186         Debug.Log(transform.position.y);
187         return false;
188     }
189 }
190
```