

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class AI_AttackPlayer_Level1 : MonoBehaviour {
6
7     GameObject player;
8     public GameObject go;
9     bool movedDown = false;
10    float playerZ;
11    float myPos;
12    float maxMax = 0.0000f;
13    float minMin = 0.0000f;
14    float timerTime;
15    Vector3 cubeQuart = new Vector3(0.25f, 0.25f, 0.25f);
16    Vector3 playerPos;
17    GameController.playerStatus glitchStatus;
18
19
20    // Start is called before the first frame update
21    void Start()
22    {
23        glitchStatus = GameController.playerStatus.Empty;
24        player = GameObject.FindGameObjectWithTag("Player");
25        go = this.gameObject;
26
27        //If there's a player object, get its bounds.
28        if (player != null)
29        {
30            Collider playerCollider = player.GetComponent<Collider>();
31            playerZ = playerCollider.bounds.center.z;
32            glitchStatus = GameController.playerStatus.Unknown;
33            playerPos = player.transform.position;
34        }
35
36        //Rotate the block to a good orientation
37        switch (go.name)
38        {
39            case "Ell1CC(Clone)":
40                Rotate();
41                break;
42            case "Ell2CC(Clone)":
43                Rotate();
44                Rotate();
45                Rotate();
46                break;
47            case "TallCC(Clone)":
48                Rotate();
49                break;
```

```
50         case "TeeCC(Clone)":
51             break;
52         case "Zed1CC(Clone)":
53             Rotate();
54             break;
55         case "Zed2CC(Clone)":
56             Rotate();
57             break;
58     }
59 }
60
61 // Update is called once per frame
62 void Update()
63 {
64     timerTime -= Time.deltaTime;
65     if (player != null) {
66
67         //Get the player's bounds
68         Collider playerCollider = player.GetComponent<Collider>();
69         playerZ = playerCollider.bounds.center.z;
70         playerPos = player.transform.position;
71         glitchStatus = GameController.playerStatus.Unknown;
72         //Check above the player
73         if (glitchStatus == GameController.playerStatus.Unknown ||
74             glitchStatus == GameController.playerStatus.Covered)
75         {
76             CheckAbove();
77         }
78         //If the timer is 0, move toward the player
79         if (timerTime <= 0)
80         {
81             if (glitchStatus == GameController.playerStatus.Uncovered)
82             {
83                 MoveToPlayer();
84             }
85         }
86     }
87     else player = GameObject.FindGameObjectWithTag("Player");
88
89 }
90 //Check above the player's z position to see if there's anything there
91 //or not.
92 void CheckAbove()
93 {
94     Vector3 vecPlus1 = new Vector3(playerPos.x, playerPos.y + 1,
95         playerPos.z);
96     Vector3 vecPlus2 = new Vector3(playerPos.x, playerPos.y + 2,
97         playerPos.z);
```

```

...ts\Scripts\AI\Attack Player\AI_AttackPlayer_Level1.cs 3
95     Vector3 vecPlus3 = new Vector3(playerPos.x, playerPos.y + 3,  ↗
        playerPos.z);
96     Vector3 vecPlus4 = new Vector3(playerPos.x, playerPos.y + 4,  ↗
        playerPos.z);
97     Vector3 vecPlus5 = new Vector3(playerPos.x, playerPos.y + 5,  ↗
        playerPos.z);
98     Vector3 vecPlus6 = new Vector3(playerPos.x, playerPos.y + 6,  ↗
        playerPos.z);
99     Vector3 vecPlus7 = new Vector3(playerPos.x, playerPos.y + 7,  ↗
        playerPos.z);
100    Vector3 vecPlus8 = new Vector3(playerPos.x, playerPos.y + 8,  ↗
        playerPos.z);
101    Vector3 vecPlus9 = new Vector3(playerPos.x, playerPos.y + 9,  ↗
        playerPos.z);
102    Vector3 vecPlus10 = new Vector3(playerPos.x, playerPos.y + 10, ↗
        playerPos.z);
103
104    if (!Physics.CheckBox(vecPlus1, cubeQuart) && !Physics.CheckBox  ↗
        (vecPlus2, cubeQuart) && !Physics.CheckBox(vecPlus3, cubeQuart)
105        && !Physics.CheckBox(vecPlus4, cubeQuart) && !Physics.CheckBox  ↗
        (vecPlus5, cubeQuart) && !Physics.CheckBox(vecPlus6,
106        cubeQuart) &&
        !Physics.CheckBox(vecPlus7, cubeQuart) && !Physics.CheckBox  ↗
        (vecPlus8, cubeQuart) && !Physics.CheckBox(vecPlus9,
107        cubeQuart) &&
        !Physics.CheckBox(vecPlus10, cubeQuart))
108    {
109        glitchStatus = GameController.playerStatus.Uncovered;
110    }
111    else glitchStatus = GameController.playerStatus.Covered;
112 }
113 //Move toward the player
114 void MoveToPlayer()
115 {
116     minMin = 0.0000f;
117     maxMax = 0.0000f;
118     foreach (Transform child in transform)
119     {
120         Collider myCollider = child.GetComponent<Collider>();
121         float myMax = myCollider.bounds.max.z;
122         float myMin = myCollider.bounds.min.z;
123         //Set the block's min and max bounds
124         if (maxMax != 0.0000f)
125         {
126             maxMax = myMax;
127         }
128         else if (myMax > maxMax)
129         {
130             maxMax = myMax;

```

```
131     }
132     if (minMin == 0.0000f)
133     {
134         minMin = myMin;
135     }
136     else if (myMin < minMin)
137     {
138         minMin = myMin;
139     }
140     //print("playerZ: " + playerZ + " minMin: " + minMin + "
141         maxMax: " + maxMax);
142 }
143     if (playerZ < minMin)
144     {
145         MoveLeft();
146     }
147     else if (playerZ > maxMax)
148     {
149         MoveRight();
150     }
151     else if (playerZ >= minMin && playerZ <= maxMax)
152     {
153         if (!movedDown)
154         {
155             MoveDown();
156             movedDown = true;
157             Destroy(this);
158         }
159     }
160     timerTime = Random.Range(0.2f, 0.6f);
161     glitchStatus = GameController.playerStatus.Unknown;
162 }
163 void MoveRight()
164 {
165     switch (go.name)
166     {
167         case "BlokCC(Clone)":
168             gameObject.GetComponent<Blok_Movement>().MoveRight();
169             break;
170         case "Ell1CC(Clone)":
171             gameObject.GetComponent<Ell1_Movement>().MoveRight();
172             break;
173         case "Ell2CC(Clone)":
174             gameObject.GetComponent<Ell2_Movement>().MoveRight();
175             break;
176         case "TallCC(Clone)":
177             gameObject.GetComponent<Tall_Movement>().MoveRight();
178             break;
```

```
179         case "TeeCC(Clone)":
180             gameObject.GetComponent<Tee_Movement>().MoveRight();
181             break;
182         case "Zed1CC(Clone)":
183             gameObject.GetComponent<Zed1_Movement>().MoveRight();
184             break;
185         case "Zed2CC(Clone)":
186             gameObject.GetComponent<Zed2_Movement>().MoveRight();
187             break;
188     }
189 }
190
191 void MoveLeft()
192 {
193     switch (go.name)
194     {
195         case "BlokCC(Clone)":
196             gameObject.GetComponent<Blok_Movement>().MoveLeft();
197             break;
198         case "Ell1CC(Clone)":
199             gameObject.GetComponent<Ell1_Movement>().MoveLeft();
200             break;
201         case "Ell2CC(Clone)":
202             gameObject.GetComponent<Ell2_Movement>().MoveLeft();
203             break;
204         case "TallCC(Clone)":
205             gameObject.GetComponent<Tall_Movement>().MoveLeft();
206             break;
207         case "TeeCC(Clone)":
208             gameObject.GetComponent<Tee_Movement>().MoveLeft();
209             break;
210         case "Zed1CC(Clone)":
211             gameObject.GetComponent<Zed1_Movement>().MoveLeft();
212             break;
213         case "Zed2CC(Clone)":
214             gameObject.GetComponent<Zed2_Movement>().MoveLeft();
215             break;
216     }
217 }
218
219 void MoveDown()
220 {
221     switch (go.name)
222     {
223         case "BlokCC(Clone)":
224             gameObject.GetComponent<Blok_Movement>().MoveDown();
225             break;
226         case "Ell1CC(Clone)":
227             gameObject.GetComponent<Ell1_Movement>().MoveDown();
```

```
228         break;
229     case "Ell2CC(Clone)":
230         gameObject.GetComponent<Ell2_Movement>().MoveDown();
231         break;
232     case "TallCC(Clone)":
233         gameObject.GetComponent<Tall_Movement>().MoveDown();
234         break;
235     case "TeeCC(Clone)":
236         gameObject.GetComponent<Tee_Movement>().MoveDown();
237         break;
238     case "Zed1CC(Clone)":
239         gameObject.GetComponent<Zed1_Movement>().MoveDown();
240         break;
241     case "Zed2CC(Clone)":
242         gameObject.GetComponent<Zed2_Movement>().MoveDown();
243         break;
244     }
245 }
246
247 void Rotate()
248 {
249     switch (go.name)
250     {
251     case "Ell1CC(Clone)":
252         gameObject.GetComponent<Ell1_Movement>().Rotate();
253         break;
254     case "Ell2CC(Clone)":
255         gameObject.GetComponent<Ell2_Movement>().Rotate();
256         break;
257     case "TallCC(Clone)":
258         gameObject.GetComponent<Tall_Movement>().Rotate();
259         break;
260     case "TeeCC(Clone)":
261         gameObject.GetComponent<Tee_Movement>().Rotate();
262         break;
263     case "Zed1CC(Clone)":
264         gameObject.GetComponent<Zed1_Movement>().Rotate();
265         break;
266     case "Zed2CC(Clone)":
267         gameObject.GetComponent<Zed2_Movement>().Rotate();
268         break;
269     }
270 }
271 }
272
```