

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using DG.Tweening;
4 using UnityEngine;
5 using static GameManager;
6
7 public class PlayerMovement : MonoBehaviour
8 {
9     private static PlayerMovement instance;
10    public int playerIndex = 0;
11    private GameManager gameManager;
12    public float moveSpeed = 2f;
13    public Transform targetPoint;
14    public Way currentNextWay;
15
16    public AudioSource moveSound;
17    public AudioSource cezaSound;
18    public AudioSource finish;
19
20    public Animator playeranim;
21
22    public GameObject dice;
23    public GameObject buttonText;
24
25    public float jumpHeight = 20f;
26    public float jumpDuration = 1f;
27
28    public bool inJail = false;
29    public bool onTurtle = false;
30    public bool tourWait = false;
31
32    public Transform hospital;
33    public int hospitalPointCount = 0;
34
35    public Transform jail;
36    public int jailPointCount = 0;
37
38    public Transform redTel;
39    public int redTelPointCount = 0;
40
41    public Transform OutTel;
42    public Way nextWayAfterOut;
43
44    public bool isStartGame = false;
45    public bool isGucluKol = false;
46
47    public static PlayerMovement Instance
48    {
49        get
50        {
51            if (instance == null)
52            {
53                instance = FindObjectOfType<PlayerMovement>();
```

```
54         }
55         return instance;
56     }
57 }
58
59 private void Start()
60 {
61     gameManager = GameManager.instance;
62     currentNextWay = gameManager.startWay.nextWay[0];
63 }
64
65 public void MoveForward(float moveCount)
66 {
67     if (currentNextWay.playerMovement.Contains(this))
68         currentNextWay.playerMovement.Remove(this);
69
70     if (isStartGame)
71         currentNextWay = currentNextWay.nextWay[0];
72
73     Move(moveCount);
74     isStartGame = true;
75 }
76
77 public void Move(float moveCount)
78 {
79     if (inJail && moveCount >= 3)
80         inJail = false;
81
82     if (moveCount <= 0 || (inJail && moveCount < 3))
83     {
84         NextPlayer();
85         return;
86     }
87
88     LeanTween.cancel(gameObject);
89     transform.DOJump(
90         new Vector3(currentNextWay.transform.position.x,
91                     currentNextWay.transform.position.y + 0.5f,
92                     currentNextWay.transform.position.z),
93         1, 1, 0.5f).OnComplete(() =>
94     {
95         moveSound.Play();
96         moveCount--;
97
98         if (moveCount == 0)
99         {
100             HandleCardEffect();
101             return;
102         }
103
104         if (hospital != null)
105             hospitalPointCount++;
```

```
105
106         if (currentNextWay.nextWay.Count > 1)
107             currentNextWay = currentNextWay.nextWay[1];
108         else
109             currentNextWay = currentNextWay.nextWay[0];
110
111         Move(moveCount);
112     });
113 }
114
115 private void HandleCardEffect()
116 {
117     switch (currentNextWay.cardType)
118     {
119         case GameManager.CardType.TeleportRed:
120             Diced.Instance.TeleportRed();
121             break;
122
123         case GameManager.CardType.Green:
124             GreenCard();
125             break;
126
127         case GameManager.CardType.Blue:
128             BlueCard();
129             break;
130
131         case GameManager.CardType.TeleportBlue:
132             TeleportPlayer();
133             break;
134
135         case GameManager.CardType.Red:
136         case GameManager.CardType.Bomb:
137             cezaSound.Play();
138             tourWait = true;
139             if (currentNextWay.cardType ==
140                 GameManager.CardType.Bomb)
141                 currentNextWay.CheckTrap();
142             NextPlayer();
143             break;
144
145         case GameManager.CardType.Bomb5Steps:
146             gameManager.players[gameManager.currentPlayer].BckMove
147                 (5);
148             currentNextWay.CheckTrap();
149             // NextPlayer();
150             break;
151
152         case GameManager.CardType.Jail:
153             inJail = true;
154             NextPlayer();
155             break;
156
157         case GameManager.CardType.Turtle:
```

```
156         onTurtle = true;
157         GameManager.instance.TurtlePanel();
158         NextPlayer();
159         break;
160
161         default:
162
163             if (currentNextWay.isFinish)
164             {
165                 finish.Play();
166                 GameManager.players.RemoveAt
167                     (GameManager.currentPlayer);
168                 GameManager.StartNextTurn();
169             }
170             else
171             {
172                 NextPlayer();
173             }
174             break;
175         }
176     }
177
178     private void TeleportPlayer()
179     {
180         int cp = GameManager.currentPlayer;
181         var player = GameManager.players[cp];
182         player.transform.position = OutTel.position;
183         player.currentNextWay = nextWayAfterOut;
184         NextPlayer();
185     }
186
187     private void NextPlayer()
188     {
189         GameManager.StartNextTurn();
190     }
191
192     private void GreenCard()
193     {
194         NextPlayer();
195     }
196
197     private void BlueCard()
198     {
199         NextPlayer();
200     }
201 }
202
203 // This is one of the games I developed for ParsGameStudio.
204 // Out of respect for the confidentiality of the company's unreleased
205 // project, I won't be sharing the entire script, but you can review a
206 // small portion of it here.
207 // Thank you for your interest!
```