

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5 using TMPro;
6 using Cinemachine;
7 using UnityEngine.UI;
8 using DG.Tweening;
9 using UnityEngine.Rendering;
10
11 public class GameManager : MonoBehaviour
12 {
13     [Range(30, 1000)]
14     public int fps;
15
16     public static GameManager instance;
17     public GameObject stopButton;
18     public GameObject ddPanel;
19     public Animator anim;
20     public Way startWay;
21     public List<PlayerMovement> players;
22     public List<CinemachineVirtualCamera> cameras = new
23         List<CinemachineVirtualCamera>();
24     public int currentPlayer;
25     public int diceSide = 0;
26
27
28     public GameObject isTwoPlayerPanel;
29
30     public GameObject twoPlayerPanel;
31
32     public float rotationDuration = 2f;
33     public float moveDuration = 2f;
34
35     private Transform cameraTarget;
36     public Camera camera;
37
38     public TextMeshProUGUI[] characterNameTexts;
39     public TextMeshPro[] characterNameTextsTwo;
40
41     public int twoTourCount = 2;
42
43     public GameObject greenCard;
44     public GameObject blueCard;
45     public GameObject bomb;
46     public GameObject bomb5stepback;
47
48     public GameObject p1turtle;
49     public GameObject p2turtle;
50     public GameObject p3turtle;
51     public GameObject p4turtle;
52
```

```
53
54     public GameObject p1redcard;
55     public GameObject p2redcard;
56     public GameObject p3redcard;
57     public GameObject p4redcard;
58
59     public GameObject diceValuePanel;
60     public GameObject diceValuePanelTwo;
61     public GameObject diceValuePanelThree;
62     public GameObject diceValuePanelFour;
63
64     public TextMeshProUGUI diceValueText;
65     public TextMeshProUGUI diceValueTextTwo;
66     public TextMeshProUGUI diceValueTextThree;
67     public TextMeshProUGUI diceValueTextFour;
68
69
70     public GameObject p1jail;
71     public GameObject p2jail;
72     public GameObject p3jail;
73     public GameObject p4jail;
74
75     public GameObject buttonText;
76     public GameObject dice;
77     public GameObject button;
78
79     public Button zarButton;
80
81     public AudioSource panelSound;
82     public AudioSource buttonSound;
83
84     public List<PlayerMovement> twoPlayers = new List<PlayerMovement>  ↗
85         ();
86     public int[] diceValues = new int[2];
87     public enum CardType
88     {
89         None, Green, Blue, Red, Turtle, Jail, Hospital, Hospital1,  ↗
90         Hospital2, Hospital3, Hospital4, Hospital5, Bomb, Bomb5Steps,  ↗
91         TeleportBlue, TeleportRed
92     }
93     private void Awake()
94     {
95         instance = this;
96     }
97
98     public void DisableRaycastOnButton()
99     {
100         GraphicRaycaster graphicRaycaster =  ↗
101             button.GetComponentInParent<GraphicRaycaster>();
102
103         if (graphicRaycaster != null)
104         {
105             graphicRaycaster.enabled = false;
106         }
107     }
108 }
```

```
102     }
103 }
104
105     private void Start()
106     {
107         Application.targetFrameRate = fps;
108
109         SetCameraTarget(currentPlayer);
110
111         PlayersAnim();
112         for (int i = 0; i < characterNameTexts.Length; i++)
113         {
114             string savedName = PlayerPrefs.GetString(i.ToString());
115             characterNameTexts[i].text = savedName;
116         }
117
118         for (int i = 0; i < characterNameTextsTwo.Length; i++)
119         {
120             string savedName = PlayerPrefs.GetString(i.ToString());
121             characterNameTextsTwo[i].text = savedName;
122         }
123
124         NickOnOff.Instance.currentPl();
125     }
126
127     public void OffButton()
128     {
129         zarButton.interactable = false;
130     }
131
132     public void EnableButton()
133     {
134         zarButton.interactable = true;
135     }
136
137     private void Update()
138     {
139
140     }
141
142     public void OpenPanel()
143     {
144         anim.SetTrigger("openPanel");
145     }
146
147     public void ClosePanel()
148     {
149         anim.SetTrigger("closePanel");
150     }
151
152     public void PlayButtonSound()
153     {
154         buttonSound.Play();
```

```
155     }
156
157     public void Cagir()
158     {
159
160     }
161
162     public void SetCameraTarget(int cameraIndex)
163     {
164         foreach (CinemachineVirtualCamera cam in cameras)
165         {
166             cam.Priority = 0;
167         }
168         cameras[cameraIndex].Priority = 1;
169
170         zarButton.interactable = true;
171     }
172
173
174     public void StartNextTurn()
175     {
176         players[currentPlayer].gameObject.GetComponent<PlayersAniim>
            ().StopAnim();
177
178         dice.SetActive(false);
179         buttonText.SetActive(true);
180
181     yeniden:
182         currentPlayer = (currentPlayer + 1) % players.Count;
183         SetCameraTarget(currentPlayer);
184         PlayersAnim();
185         if (players[currentPlayer].tourWait == true)
186         {
187             players[currentPlayer].tourWait = false;
188             goto yeniden;
189         }
190
191         if (players[currentPlayer].twoTourWait == true && twoTourCount
            == 0)
192         {
193             players[currentPlayer].twoTourWait = false;
194         }
195
196         else if (players[currentPlayer].twoTourWait == true &&
            twoTourCount != 0)
197         {
198             twoTourCount--;
199             goto yeniden;
200         }
201
202         PlayersAnim();
203         NickOnOff.Instance.currentPl();
204         //if (players[0].tourWait == true)
```

```
205         //{
206         //     p1redcard.SetActive(true);
207         //}
208
209         //else if (players[0].tourWait == false)
210         //{
211         //     p1redcard.SetActive(false);
212         //}
213
214         //if (players[1].tourWait == true)
215         //{
216         //     p2redcard.SetActive(true);
217         //}
218
219         //else if (players[1].tourWait == false)
220         //{
221         //     p2redcard.SetActive(false);
222         //}
223
224         //if (players[2].tourWait == true)
225         //{
226         //     p3redcard.SetActive(true);
227         //}
228
229         //else if (players[2].tourWait == false)
230         //{
231         //     p3redcard.SetActive(false);
232         //}
233
234         //if (players[3].tourWait == true)
235         //{
236         //     p4redcard.SetActive(true);
237         //}
238
239         //else if (players[3].tourWait == false)
240         //{
241         //     p4redcard.SetActive(false);
242         //}
243
244         if (players[currentPlayer].isSuspen == true)
245         {
246             players[currentPlayer].isSuspen = false;
247             CardEvent.instance.BackComeGreenCard();
248         }
249
250         if (players[currentPlayer].haveCard == true)
251         {
252             CardEvent.instance.WhenComeBack();
253         }
254         if (currentPlayer == CheckCards.instance.currentPlayerr)
255         {
256
257         }
```

```
258
259     PlayersAnim();
260
261 }
262
263 public void PlayersAnim()
264 {
265     players[currentPlayer].gameObject.GetComponent<PlayersAniim>
        ().StartAnim();
266
267 }
268
269 public void TwoPlayersAnim()
270 {
271     players
        [Way.instance.twoPlayerOne].gameObject.GetComponent<PlayersAn
        iim>().StartAnim();
272 }
273
274 public void TwoPlayersAnimSecond()
275 {
276     players
        [Way.instance.twoPlayerOne].gameObject.GetComponent<PlayersAn
        iim>().StopAnim();
277     players
        [Way.instance.twoPlayerTwo].gameObject.GetComponent<PlayersAn
        iim>().StartAnim();
278 }
279
280 public void TwoPlayersAnimSecondStop()
281 {
282     players
        [Way.instance.twoPlayerTwo].gameObject.GetComponent<PlayersAn
        iim>().StopAnim();
283 }
284
285 public void SetDiceSide(int side)
286 {
287     diceSide = side;
288 }
289
290 public int GetDiceSide()
291 {
292     return diceSide;
293 }
294 int diceValuesIndex = 1;
295 public void OnDiceRolled(float result)
296 {
297     PlayersAnim();
298     diceSide = (int)result;
299     if (players[currentPlayer].isTwoPlayer == true)
300     {
301         PlayersAnim();
```

```
302 //StartCoroutine(Diced.Instance.RollDiceforTwoPlayer);
303 diceValues[diceValuesIndex] = (int)result;
304 diceValuesIndex--;
305 currentPlayer = twoPlayers[0].playerIndex;
306 if (diceValuesIndex < 0)
307 {
308
309     if (diceValues[0] < diceValues[1] && twoPlayers[0].haveArmor)
310     {
311         CardEvent.instance.ComeBackCard(players
312         [currentPlayer].cardTypeeee.gameObject);
313     }
314     else if (diceValues[0] < diceValues[1])
315     {
316         twoPlayers[0].GoToHospital();
317         twoPlayers.Clear();
318     }
319 }
320
321 else if (diceValues[0] > diceValues[1] && twoPlayers[1].haveArmor)
322 {
323     CardEvent.instance.ComeBackCard(players
324     [currentPlayer].cardTypeeee.gameObject);
325 }
326 else if (diceValues[0] > diceValues[1])
327 {
328     twoPlayers[1].GoToHospital();
329     twoPlayers.Clear();
330 }
331
332
333
334
335     diceValuesIndex = 1;
336     currentPlayer = twoPlayers[1].playerIndex;
337     twoPlayers[0].isTwoPlayer = false;
338     twoPlayers[1].isTwoPlayer = false;
339     twoPlayers.Clear();
340 }
341 // StartNextTurn();
342 }
343
344 else if (players[currentPlayer].onTurtle == true)
345 {
346
347     result = result / 2;
348     result = Mathf.Ceil(result);
349     players[currentPlayer].onTurtle = false;
350     players[currentPlayer].moveCountt = (int)result;
```

```
351         players[currentPlayer].MoveForward(result);
352     }
353
354     else
355     {
356         players[currentPlayer].moveCountt = (int)result;
357         players[currentPlayer].MoveForward(result);
358     }
359
360 }
361
362 public void IfTwoPlayer()
363 {
364     players[currentPlayer].GetComponent<PlayersAniim>().StopAnim();
365 }
366
367 public void TurtlePanel()
368 {
369
370     if (players[0].onTurtle == true)
371     {
372         p1turtle.SetActive(true);
373     }
374
375     else if (players[0].onTurtle == false)
376     {
377         p1turtle.SetActive(false);
378     }
379
380     if (players[1].onTurtle == true)
381     {
382         p2turtle.SetActive(true);
383     }
384
385     else if (players[1].onTurtle == false)
386     {
387         p2turtle.SetActive(false);
388     }
389
390     if (players[2].onTurtle == true)
391     {
392         p3turtle.SetActive(true);
393     }
394
395     else if (players[2].onTurtle == false)
396     {
397         p3turtle.SetActive(false);
398     }
399
400     if (players[3].onTurtle == true)
401     {
402         p4turtle.SetActive(true);
403     }
```



```
404
405     else if (players[3].onTurtle == false)
406     {
407         p4turtle.SetActive(false);
408     }
409 }
410
411 //public void JailPanel()
412 //{
413 //    if (players[0].inJail == true)
414 //    {
415 //        p1jail.SetActive(true);
416 //    }
417 //
418 //    else if (players[0].inJail == false)
419 //    {
420 //        p1jail.SetActive(false);
421 //    }
422 //
423 //    if (players[1].inJail == true)
424 //    {
425 //        p2jail.SetActive(true);
426 //    }
427 //
428 //    else if (players[1].inJail == false)
429 //    {
430 //        p2jail.SetActive(false);
431 //    }
432 //
433 //    if (players[2].inJail == true)
434 //    {
435 //        p3jail.SetActive(true);
436 //    }
437 //
438 //    else if (players[2].inJail == false)
439 //    {
440 //        p3jail.SetActive(false);
441 //    }
442 //
443 //    if (players[3].inJail == true)
444 //    {
445 //        p4jail.SetActive(true);
446 //    }
447 //
448 //    else if (players[3].inJail == false)
449 //    {
450 //        p4jail.SetActive(false);
451 //    }
452 //}
453
454 public void ActiveTwoPanel()
455 {
456
```

```
457
458     panelSound.Play();
459     isTwoPlayerPanel.SetActive(true);
460
461
462     isTwoPlayerPanel.GetComponent<Animator>().Play("panelanim");
463
464 }
465
466
467
468 public void TwoPlayersSameBlock(List<PlayerMovement> players)
469 {
470     ActiveTwoPanel();
471     PlayersAnim();
472
473     for (int i = 0; i < players.Count; i++)
474     {
475         twoPlayers.Add(players[i]);
476     }
477
478     if (twoPlayers[0].isSheildForFight == true)
479     {
480         CardEvent.instance.ComeBackCardGreen(twoPlayers
481             [0].sheildForFightCard);
482
483     }
484
485     else if (twoPlayers[1].isSheildForFight == true)
486     {
487         CardEvent.instance.ComeBackCardGreen(twoPlayers
488             [1].sheildForFightCard);
489
490     }
491
492     else
493     {
494         //NickOnOff.Instance.currentPlMeydanOkuma(twoPlayers
495             [0].playerIndex);
496         //SetCameraTarget(twoPlayers[0].playerIndex);
497         //SelectedPlayerChange(twoPlayers[0].playerIndex);
498     }
499 }
500
501 public TextMeshPro dice1;
502 public TextMeshPro dice2;
503
504 public void TwoPlayersCheck()
505 {
506     if (players[0].isTwoPlayer == true)
507     {
508         twoPlayers.Add(players[0]);
509     }
510 }
```

```
507
508     if (players[1].isTwoPlayer == true)
509     {
510         twoPlayers.Add(players[1]);
511     }
512
513     if (players[2].isTwoPlayer == true)
514     {
515         twoPlayers.Add(players[2]);
516     }
517     if (players[3].isTwoPlayer == true)
518     {
519         twoPlayers.Add(players[3]);
520     }
521 }
522
523
524 public void CancelSiyrilmaci()
525 {
526     panelSound.Play();
527     isTwoPlayerPanel.SetActive(true);
528
529
530     isTwoPlayerPanel.transform.position = new Vector3
531         (isTwoPlayerPanel.transform.position.x, Screen.height,
532         isTwoPlayerPanel.transform.position.z);
533
534
535
536     isTwoPlayerPanel.transform.localScale = Vector3.zero;
537
538     LeanTween.moveY(isTwoPlayerPanel, Screen.height * 0.5f, 0.5f)
539         .setEaseOutBounce()
540         .setOnComplete(() =>
541         {
542             });
543
544     LeanTween.scale(isTwoPlayerPanel, Vector3.one, 0.5f)
545         .setEase(LeanTweenType.easeOutCubic);
546 }
547
548 void SelectedPlayerChange(int playerIndex)
549 {
550     for (int i = 0; i < players.Count; i++)
551     {
552         LeanTween.cancel(players[i].gameObject);
553         players[i].transform.localScale = Vector3.one * 0.5f;
554     }
555     LeanTween.scale(players[playerIndex].gameObject, Vector3.one *
556         0.633135736f, 0.5f).setLoopPingPong();
```

```
557     public bool isMeydanOkuma = false;
558     //public void TwoPlayerSelect(List<PlayerMovement> selectedPlayers)
559     //{
560         //    if (selectedPlayers.Count == 2)
561         //    {
562             //        twoPlayerPanel.SetActive(false);
563             //        isMeydanOkuma = true;
564             //        NickOnOff.Instance.currentPlMeydanOkuma(selectedPlayers
565                 [0].playerIndex);
566             //        SetCameraTarget(selectedPlayers[0].playerIndex);
567             //        SelectedPlayerChange(selectedPlayers[0].playerIndex);
568         //    }
569     //}
570
571     public void PrintDiceValues()
572     {
573         if (twoPlayers[0] == players[0] || twoPlayers[1] == players[0])
574         {
575             diceValuePanel.SetActive(true);
576         }
577
578         if (twoPlayers[0] == players[1] || twoPlayers[1] == players[1])
579         {
580             diceValuePanelTwo.SetActive(true);
581         }
582
583         if (twoPlayers[0] == players[2] || twoPlayers[1] == players[2])
584         {
585             diceValuePanelThree.SetActive(true);
586         }
587
588         if (twoPlayers[0] == players[3] || twoPlayers[1] == players[3])
589         {
590             diceValuePanelFour.SetActive(true);
591         }
592     }
593
594     public int victimOneDice;
595     public int victimTwoDice;
596
597     public void TwoPlayersDices()
598     {
599         if (victimOneDice < victimTwoDice)
600         {
601             GameManager.instance.twoPlayers[0].GoToHospital();
602             twoPlayers.Remove(twoPlayers[1]);
603             twoPlayers.Remove(twoPlayers[0]);
604             Way.instance.playerMovement.Clear();
605         }
606
607         else if (victimOneDice > victimTwoDice)
608         {
```

```
609         GameManager.instance.twoPlayers[1].GoToHospital();
610         twoPlayers.Remove(twoPlayers[1]);
611         twoPlayers.Remove(twoPlayers[0]);
612         Way.instance.playerMovement.Clear();
613     }
614
615     else if (victimOneDice == victimTwoDice)
616     {
617         Diced.Instance.TwoDeneme();
618     }
619
620
621 }
622
623 public void SameBlockDiceValue(int diceValue)
624 {
625     Debug.Log("Zar atıldı: " + diceValue);
626     meydanOkumaDiceValues.Add(diceValue);
627     // NickOnOff.Instance.currentPlMeydanOkuma(twoPlayers
628     [1].playerIndex);
629     //SetCameraTarget(twoPlayers[1].playerIndex);
630     SelectedPlayerChange(twoPlayers[1].playerIndex);
631
632     if (twoPlayers.Count >= 2)
633     {
634         if (twoPlayers[0].playerIndex == 0)
635         {
636             diceValueText.text = meydanOkumaDiceValues[0].ToString
637             ();
638         }
639         else if (twoPlayers[1].playerIndex == 0 &&
640             meydanOkumaDiceValues.Count >= 2)
641         {
642             diceValueText.text = meydanOkumaDiceValues[1].ToString
643             ();
644         }
645         if (twoPlayers[0].playerIndex == 1)
646         {
647             diceValueTextTwo.text = meydanOkumaDiceValues
648             [0].ToString();
649         }
650         else if (twoPlayers[1].playerIndex == 1 &&
651             meydanOkumaDiceValues.Count >= 2)
652         {
653             diceValueTextTwo.text = meydanOkumaDiceValues
654             [1].ToString();
655         }
656         if (twoPlayers[0].playerIndex == 2)
```

```
655     {
656         diceValueTextThree.text = meydanOkumaDiceValues
        [0].ToString();
657     }
658
659     else if (twoPlayers[1].playerIndex == 2 &&
        meydanOkumaDiceValues.Count >= 2)
660     {
661         diceValueTextThree.text = meydanOkumaDiceValues
        [1].ToString();
662     }
663
664     if (twoPlayers[0].playerIndex == 3)
665     {
666         diceValueTextFour.text = meydanOkumaDiceValues
        [0].ToString();
667     }
668
669     else if (twoPlayers[1].playerIndex == 3 &&
        meydanOkumaDiceValues.Count >= 2)
670     {
671         diceValueTextFour.text = meydanOkumaDiceValues
        [1].ToString();
672     }
673 }
674
675 if (meydanOkumaDiceValues.Count >= 2)
676 {
677
678
679     for (int i = 0; i < twoPlayers.Count; i++)
680     {
681         twoPlayers[i].isTwoPlayer = false;
682     }
683     if (meydanOkumaDiceValues[0] == meydanOkumaDiceValues[1])
684     {
685         meydanOkumaDiceValues.RemoveAt(1);
686         StartCoroutine(Diced.Instance.RollDiceforTwoPlayer());
687     }
688     else
689     {
690         if (meydanOkumaDiceValues[0] > meydanOkumaDiceValues
        [1])
691         {
692             twoPlayers[1].GoToHospital();
693             twoPlayers.Remove(twoPlayers[1]);
694             twoPlayers.Remove(twoPlayers[0]);
695         }
696         else if (meydanOkumaDiceValues[0] <
        meydanOkumaDiceValues[1])
697         {
698             twoPlayers[0].GoToHospital();
699             twoPlayers.Remove(twoPlayers[0]);
```

```
700         twoPlayers.Remove(twoPlayers[1]);
701     }
702
703     meydanOkumaDiceValues.Clear();
704     // isMeydanOkuma = false;
705     twoPlayers.Clear();
706     StartNextTurn();
707
708     Invoke("CloseDiceValues", 2f);
709 }
710
711 // StartNextTurn();
712
713
714
715 }
716 }
717 void CloseDiceValues()
718 {
719     diceValueText.text = "";
720     diceValueTextTwo.text = "";
721     diceValueTextThree.text = "";
722     diceValueTextFour.text = "";
723
724     diceValuePanel.SetActive(false);
725     diceValuePanelTwo.SetActive(false);
726     diceValuePanelThree.SetActive(false);
727     diceValuePanelFour.SetActive(false);
728
729 }
730 public List<int> meydanOkumaDiceValues = new List<int>();
731 public void MeydanOkumaDiceValue(int diceValue)
732 {
733     Debug.Log("Zar atıldı: " + diceValue);
734     meydanOkumaDiceValues.Add(diceValue);
735     NickOnOff.Instance.currentPlMeydanOkuma      ↗
736         (CardEvent.instance.selectedPlayers[1].playerIndex);
737     SetCameraTarget(CardEvent.instance.selectedPlayers      ↗
738         [1].playerIndex);
739     SelectedPlayerChange(CardEvent.instance.selectedPlayers      ↗
740         [1].playerIndex);
741
742     if (meydanOkumaDiceValues.Count >= 2)
743     {
744         if (meydanOkumaDiceValues[0] == meydanOkumaDiceValues[1])
745         {
746             meydanOkumaDiceValues.RemoveAt(1);
747             StartCoroutine(Diced.Instance.MeydanOkumaRollDice());
748         }
749         else
750         {
751             if (meydanOkumaDiceValues[0] > meydanOkumaDiceValues      ↗
```

```
[1])
750     {
751         CardEvent.instance.selectedPlayers[1].BckMove
            (meydanOkumaDiceValues[1]);
752     }
753     else if (meydanOkumaDiceValues[0] <
            meydanOkumaDiceValues[1])
754     {
755         CardEvent.instance.selectedPlayers[0].BckMove
            (meydanOkumaDiceValues[0]);
756     }
757
758     // meydanOkumaDiceValues.Clear();
759     // isMeydanOkuma = false;
760     CardEvent.instance.selectedPlayers.Clear();
761 }
762
763 // StartNextTurn();
764 }
765 }
766
767 public void ZarAtButton()
768 {
769     buttonText.SetActive(false);
770     dice.SetActive(true);
771     if (isMeydanOkuma)
772     {
773         StartCoroutine(Diced.Instance.MeydanOkumaRollDice());
774     }
775     else if (twoPlayers.Count >= 2)
776     {
777         StartCoroutine(Diced.Instance.RollDiceforTwoPlayer());
778     }
779     else
780         Diced.Instance.StartWithClick();
781 }
782
783 }
```