```
 1 using System.Collections;
 2 using System.Collections.Generic;
 3 using UnityEngine;
 4 using UnityEngine.UI;
 5 using UnityEngine.SceneManagement;
 6
 7
 8 public class UIScript : MonoBehaviour
 9 {
10     public Image SpeedRing;
11     public Text SpeedText;
12     public Text GearText;
13     public Text LapNumberText;
14     public Text TotalLapsText;
15     public Text LapTimeMinutesText;
16     public Text LapTimeSecondsText;
17     public Text RaceTimeMinutesText;
18     public Text RaceTimeSecondsText;
19     public Text BestLapTimeMinutes;
20     public Text BestLapTimeSeconds;
21     public Text CheckPointTime;
22     public Text WrongWayT;
23     public Text TotalCarsText;
24     public Text PlayersPosition;
25     public GameObject CheckPointDisplay;
26     public GameObject NewLapRecord;
27     public GameObject WrongWayText;
28     private float DisplaySpeed;
29     private int TotalLaps = 3;
30     private int TotalCars = 1;
31     public bool RaceTrack = true;
32     public GameObject F1Opponent1;
33     public GameObject F1Opponent2;
34     public GameObject F1Opponent3;
35     public GameObject F1Opponent4;
```

```
36      public GameObject F1Opponent5;
37      public GameObject F1Opponent6;
38      public GameObject F1Opponent7;
39      public GameObject QuitMenu;
40
41
42
43      // Start is called before the first frame update
44      void Start()
45      {
46          TotalLaps = UniversalSave.LapCounts;
47          TotalCars = UniversalSave.OpponentsCount + 1;
48          SpeedRing.fillAmount = 0;
49          SpeedText.text = "0";
50          GearText.text = "1";
51          LapNumberText.text = "0";
52          TotalLapsText.text = "/" + TotalLaps.ToString();
53          CheckPointDisplay.SetActive(false);
54          NewLapRecord.SetActive(false);
55          WrongWayText.SetActive(false);
56          SaveScript.MaxLaps = TotalLaps;
57          TotalCarsText.text = "/" + TotalCars.ToString();
58          PlayersPosition.text = "1";
59          if (RaceTrack == true)
60          {
61              SetCarVisibility();
62              QuitMenu.SetActive(false);
63          }
64      }
65
66      void SetCarVisibility()
67      {
68          if (TotalCars == 1)
69          {
70              F1Opponent1.SetActive(false);
```

```
71              F1Opponent2.SetActive(false);
72              F1Opponent3.SetActive(false);
73              F1Opponent4.SetActive(false);
74              F1Opponent5.SetActive(false);
75              F1Opponent6.SetActive(false);
76              F1Opponent7.SetActive(false);
77          }
78          if (TotalCars == 2)
79          {
80              F1Opponent1.SetActive(true);
81              F1Opponent2.SetActive(false);
82              F1Opponent3.SetActive(false);
83              F1Opponent4.SetActive(false);
84              F1Opponent5.SetActive(false);
85              F1Opponent6.SetActive(false);
86              F1Opponent7.SetActive(false);
87          }
88          if (TotalCars == 3)
89          {
90              F1Opponent1.SetActive(true);
91              F1Opponent2.SetActive(true);
92              F1Opponent3.SetActive(false);
93              F1Opponent4.SetActive(false);
94              F1Opponent5.SetActive(false);
95              F1Opponent6.SetActive(false);
96              F1Opponent7.SetActive(false);
97          }
98          if (TotalCars == 4)
99          {
100             F1Opponent1.SetActive(true);
101             F1Opponent2.SetActive(true);
102             F1Opponent3.SetActive(true);
103             F1Opponent4.SetActive(false);
104             F1Opponent5.SetActive(false);
105             F1Opponent6.SetActive(false);
```

```
106             F1Opponent7.SetActive(false);
107         }
108         if (TotalCars == 5)
109         {
110             F1Opponent1.SetActive(true);
111             F1Opponent2.SetActive(true);
112             F1Opponent3.SetActive(true);
113             F1Opponent4.SetActive(true);
114             F1Opponent5.SetActive(false);
115             F1Opponent6.SetActive(false);
116             F1Opponent7.SetActive(false);
117         }
118         if (TotalCars == 6)
119         {
120             F1Opponent1.SetActive(true);
121             F1Opponent2.SetActive(true);
122             F1Opponent3.SetActive(true);
123             F1Opponent4.SetActive(true);
124             F1Opponent5.SetActive(true);
125             F1Opponent6.SetActive(false);
126             F1Opponent7.SetActive(false);
127         }
128         if (TotalCars == 7)
129         {
130             F1Opponent1.SetActive(true);
131             F1Opponent2.SetActive(true);
132             F1Opponent3.SetActive(true);
133             F1Opponent4.SetActive(true);
134             F1Opponent5.SetActive(true);
135             F1Opponent6.SetActive(true);
136             F1Opponent7.SetActive(false);
137         }
138         if (TotalCars == 8)
139         {
140             F1Opponent1.SetActive(true);
```

```
141             F1Opponent2.SetActive(true);
142             F1Opponent3.SetActive(true);
143             F1Opponent4.SetActive(true);
144             F1Opponent5.SetActive(true);
145             F1Opponent6.SetActive(true);
146             F1Opponent7.SetActive(true);
147         }
148     }
149 
150 
151     // Update is called once per frame
152     void Update()
153     {
154         // Speedometer
155         DisplaySpeed = SaveScript.Speed / SaveScript.TopSpeed;
156         SpeedRing.fillAmount = DisplaySpeed;
157         SpeedText.text = (Mathf.Round(SaveScript.Speed).ToString());
158         GearText.text = (SaveScript.Gear + 1).ToString();
159 
160         //LapNumber
161         LapNumberText.text = SaveScript.LapNumber.ToString();
162 
163         //LapTime
164         if(SaveScript.LapTimeMinutes <= 9)
165         {
166             LapTimeMinutesText.text = "0" + (Mathf.Round(SaveScript.LapTimeMinutes).ToString()) + ":";
167         }
168         else if (SaveScript.LapTimeMinutes >= 10)
169         {
170             LapTimeMinutesText.text = (Mathf.Round(SaveScript.LapTimeMinutes).ToString()) + ":";
171         }
172         if (SaveScript.LapTimeSeconds <= 9)
173         {
174             LapTimeSecondsText.text = "0" + (Mathf.Round(SaveScript.LapTimeSeconds).ToString());
175         }
```

```
176         else if (SaveScript.LapTimeSeconds >= 10)
177         {
178             LapTimeSecondsText.text = (Mathf.Round(SaveScript.LapTimeSeconds).ToString());
179         }
180 
181         //RaceTime
182         if (SaveScript.RaceTimeMinutes <= 9)
183         {
184             RaceTimeMinutesText.text = "0" + (Mathf.Round(SaveScript.RaceTimeMinutes).ToString()) + ":";
185         }
186         else if (SaveScript.RaceTimeMinutes >= 10)
187         {
188             RaceTimeMinutesText.text = (Mathf.Round(SaveScript.RaceTimeMinutes).ToString()) + ":";
189         }
190         if (SaveScript.RaceTimeSeconds <= 9)
191         {
192             RaceTimeSecondsText.text = "0" + (Mathf.Round(SaveScript.RaceTimeSeconds).ToString());
193         }
194         else if (SaveScript.RaceTimeSeconds >= 10)
195         {
196             RaceTimeSecondsText.text = (Mathf.Round(SaveScript.RaceTimeSeconds).ToString());
197         }
198 
199         //Working out best Lap Time
200         if (SaveScript.LapChange == true)
201         {
202             if (SaveScript.LastLapM == SaveScript.BestLapTimeM)
203             {
204                 if (SaveScript.LastLapS < SaveScript.BestLapTimeS)
205                 {
206                     SaveScript.BestLapTimeS = SaveScript.LastLapS;
207                     SaveScript.NewRecord = true;
208                 }
209             }
210             if (SaveScript.LastLapM < SaveScript.BestLapTimeM)
```

```
211             {
212                 SaveScript.BestLapTimeM = SaveScript.LastLapM;
213                 SaveScript.BestLapTimeS = SaveScript.LastLapS;
214                 SaveScript.NewRecord = true;
215             }
216         }
217 
218 
219         //Display Best Lap Time
220         if (SaveScript.BestLapTimeM <= 9)
221         {
222             BestLapTimeMinutes.text = "0" + (Mathf.Round(SaveScript.BestLapTimeM).ToString()) + ":";
223         }
224         else if (SaveScript.BestLapTimeM >= 10)
225         {
226             BestLapTimeMinutes.text = (Mathf.Round(SaveScript.BestLapTimeM).ToString()) + ":";
227         }
228         if (SaveScript.BestLapTimeS <= 9)
229         {
230             BestLapTimeSeconds.text = "0" + (Mathf.Round(SaveScript.BestLapTimeS).ToString());
231         }
232         else if (SaveScript.BestLapTimeS >= 10)
233         {
234             BestLapTimeSeconds.text = (Mathf.Round(SaveScript.BestLapTimeS).ToString());
235         }
236 
237         if(SaveScript.NewRecord == true)
238         {
239             NewLapRecord.SetActive(true);
240             StartCoroutine(LapRecordOff());
241         }
242 
243 
244         //CheckPoint working out for CheckPoint 1
245         if(SaveScript.CheckPointPass1 == true)
```

```
246         {
247             SaveScript.CheckPointPass1 = false;
248             Debug.Log("CheckPoint1 Passed");
249             if (SaveScript.LapNumber > 1)
250             {
251                 CheckPointDisplay.SetActive(true);
252 
253                 if (SaveScript.ThisCheckPoint1 > SaveScript.LastCheckPoint1)
254                 {
255                     CheckPointTime.color = Color.red;
256                     CheckPointTime.text = "-" + (SaveScript.ThisCheckPoint1 - SaveScript.LastCheckPoint1).ToString();
257                     StartCoroutine(CheckPointOff());
258                 }
259 
260                 if (SaveScript.ThisCheckPoint1 < SaveScript.LastCheckPoint1)
261                 {
262                     CheckPointTime.color = Color.green;
263                     CheckPointTime.text = "+" + (SaveScript.LastCheckPoint1 - SaveScript.ThisCheckPoint1).ToString();
264                     StartCoroutine(CheckPointOff());
265                 }
266             }
267         }
268 
269         //CheckPoint working out for CheckPoint 2
270         if (SaveScript.CheckPointPass2 == true)
271         {
272             SaveScript.CheckPointPass2 = false;
273             Debug.Log("CheckPoint2 Passed");
274             if (SaveScript.LapNumber > 1)
275             {
276                 CheckPointDisplay.SetActive(true);
277 
278                 if (SaveScript.ThisCheckPoint2 > SaveScript.LastCheckPoint2)
279                 {
280                     CheckPointTime.color = Color.red;
```

```
281                     CheckPointTime.text = "-" + (SaveScript.ThisCheckPoint2 - SaveScript.LastCheckPoint2).ToString();
282                     StartCoroutine(CheckPointOff());
283                 }
284 
285                 if (SaveScript.ThisCheckPoint2 < SaveScript.LastCheckPoint2)
286                 {
287                     CheckPointTime.color = Color.green;
288                     CheckPointTime.text = "+" + (SaveScript.LastCheckPoint2 - SaveScript.ThisCheckPoint2).ToString();
289                     StartCoroutine(CheckPointOff());
290                 }
291             }
292         }
293 
294         //Wrong way message
295         if(SaveScript.WrongWay == true)
296         {
297             WrongWayText.SetActive(true);
298         }
299         if (SaveScript.WrongWay == false)
300         {
301             WrongWayText.SetActive(false);
302         }
303 
304         //Wrong Way Reset Text
305         if(SaveScript.WWTextReset == false)
306         {
307             WrongWayT.text = "WRONG WAY!";
308         }
309         if (SaveScript.WWTextReset == true)
310         {
311             WrongWayT.text = " ";
312         }
313 
314         //Display Position
315 
```

```
316         PlayersPosition.text = SaveScript.PlayerPosition.ToString();
317
318
319     //Switching on the quit menu
320     if(RaceTrack == true)
321         {
322             if(Input.GetKeyDown(KeyCode.Escape))
323             {
324                 QuitMenu.SetActive(true);
325             }
326         }
327
328     }
329
330     IEnumerator CheckPointOff()
331     {
332         yield return new WaitForSeconds(2);
333         CheckPointDisplay.SetActive(false);
334     }
335
336     IEnumerator LapRecordOff()
337     {
338         yield return new WaitForSeconds(2);
339         SaveScript.NewRecord = false;
340         NewLapRecord.SetActive(false);
341     }
342
343     public void QuitRace()
344     {
345         SceneManager.LoadScene(1);
346     }
347
348     public void CloseQuit()
349     {
350         QuitMenu.SetActive(false);
```

```
351     }
352
353     public void RaceReturnToMenu()
354     {
355         SceneManager.LoadScene(1);
356     }
357
358 }
359
```