CODE NINJAS®

Bronze Belt Ninja Guide

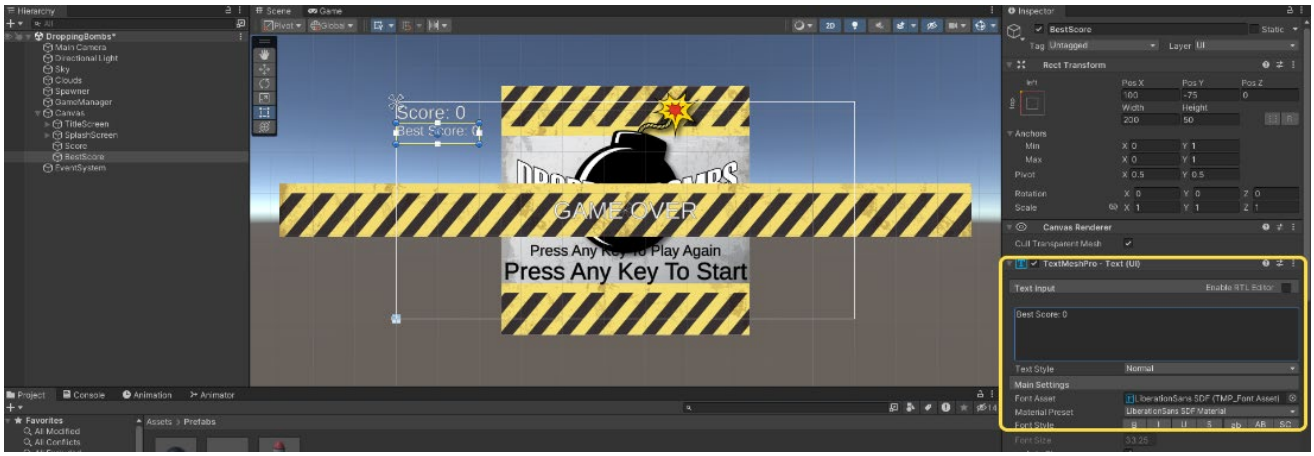
Activity 11: Dropping Bombs Part 5

Activity 11: Dropping Bombs Part 5

What's the point of having a score if you have nothing to compare it to? In this activity, you will complete the Dropping Bombs game by adding a Best Score feature.

1 Before making additional changes to your project, let's make a backup of the entire game. In the **Projects** panel, make sure that the **Assets** folder is selected. Then click on the **Assets** tab at the top of the screen and select **Export Package**. Make sure that all the assets are selected before clicking on the **Export** button. Give the package a name like ***JS-DroppingBombsPart4*.**

2 The first thing we need to do is have something in the interface to display the best score. Select the **Canvas** in the **Hierarchy** panel and find the **Score** Text object. Select it and make a copy of it by clicking **Ctrl+D**. Name the new object ***BestScore*** and position it so that it is below the Score object. Change the text to ***Best Score: 0***.



3 Open the **GameManager** script in the script manager. We need to set up some variables to track the best score. First, create a **private** integer for the score itself named ***bestScore***. Set it to **0**.

You'll need a **public** variable of the **TMP_Text** type for the new text object that was just created. Let's call it ***bestScoreText***.

Finally, in order to know if the high score has been beaten, create a **private bool** called ***beatBestScore***.

```
19      [Header("Score")]
20      public TMP_Text scoreText;
21      public int pointsWorth = 1;
22      private int score;
23
24      private int bestScore = 0;
25      public TMP_Text bestScoreText;
26      private bool beatBestScore;
27
28
29      private bool smokeCleared = true;
30
            Unity Message | 0 references
31      private void Awake()
32      {
```

4 When the game is started, you want to hide both the **Score** and **BestScore** objects. Add a line to the **Awake** function to set **bestScoreText.enabled** to **false**.

We are storing the value of **bestScore** as an integer in the **PlayerPrefs**. **PlayerPrefs** is data that gets saved on the local machine with the game and is not lost when the game is stopped.

Set **bestScore** to **PlayerPrefs.GetInt("BestScore")** and set the **bestScoreText.text** to ***"Best Score: "*+ bestScore.ToString()** (note the extra space after the colon and before closing the quotes. **ToString** converts the integer into text).

```
31      Unity Message | 0 references
        private void Awake()
32      {
33          spawner = GameObject.Find("Spawner").GetComponent<Spawner>();
34          screenBounds = Camera.main.ScreenToWorldPoint(new Vector3(Screen.width, Screen.height, Camera.main.transform.position.z));
35          scoreText.enabled = false;
36
37          bestScoreText.enabled = false;
38      }
39
40      // Start is called before the first frame update
        Unity Message | 0 references
41      void Start()
42      {
43          spawner.active = false;
44          title.SetActive(true);
45          splash.SetActive(false);
46
47          bestScore = PlayerPrefs.GetInt("BestScore");
48          bestScoreText.text = "Best Score: " + bestScore.ToString();
49      }
50
```

5 In the **ResetGame** function, you'll do two things. To start a new game, reset the **beatBestScore** boolean to false.

Also set **bestScoreText.enabled** to **true** so that you can see it.

```
        1 reference
106     void ResetGame()
107     {
108         spawner.active = true;
109         title.SetActive(false);
110
111         splash.SetActive(false);
112
113         scoreText.enabled = true;
114         score = 0;
115
116         beatBestScore = false;
117         bestScoreText.enabled = true;
118
119         player = Instantiate(playerPrefab,new Vector3(0,0,0), playerPrefab.transform.rotation);
120         gameStarted = true;
121     }
122 }
123
```

6 When the **Rocket** is **destroyed**, **OnPlayerKilled** is called. This is a good place to see if the player has beat the best score. First, you'll want to get the value of the current score from the **scoreSystem** object. This is assigned to the variable called **score**.

Next, compare **score** to the value of **bestScore** as stored in **PlayerPrefs**. If score is greater, then change **bestScore** to equal score and save the new value of **bestScore** to **PlayerPrefs**.

Then we set **beatBestScore** to **true** and update the text of the **bestScoreText** object.

Save your script.

```
1 reference
90  void OnPlayerKilled()
91  {
92      spawner.active = false;
93      gameStarted = false;
94
95      splash.SetActive(true);
96
97      Invoke("SplashScreen", 2);
98
99      if(score > bestScore)
100     {
101         bestScore = score;
102         PlayerPrefs.SetInt("BestScore", bestScore);
103         beatBestScore = true;
104         bestScoreText.text = "Best Score: " + bestScore.ToString();
105     }
106 }
107
0 references
```

7 Switch back to Unity. Select the **GameManager** object and drag the **BestScore** text object into the slot for **Best Score Text** in the script component.

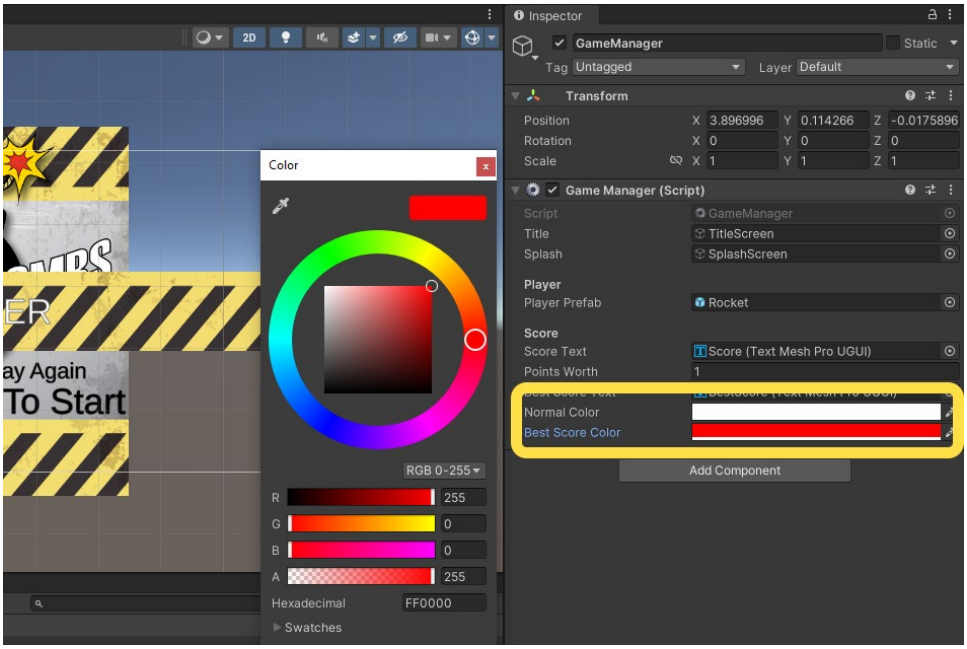
8 At this point you can test the game and see the Best Score working, but we are going to add one last detail. When we beat the high score, we are going to change the text color.

To start, let's go back to our **GameManager** script and create two public Color variables.

```
23
24      private int bestScore = 0;
25      public TMP_Text bestScoreText;
26      private bool beatBestScore;
27
28      public Color normalColor;
29      public Color bestScoreColor;
30
31      private bool smokeCleared = true;
32
        Unity Message | 0 references
33      private void Awake()
34      {
35          spawner = GameObject.Find("Spawner").GetComponent<Spawner>();
36          screenBounds = Camera.main.ScreenToWorldPoint(new Vector3(Screen.width, Screen.height, Camera.m
```

9 Back in Unity, we can see the two **Color** variables that we can click on to change the color. For our **normalColor**, we probably want white or black, but for the **bestScoreColor**, you can pick whatever color you want. I'll use red.

If you can't see the color, try changing the **Alpha (A)** value of the **Color** which defaults to **0**.



10 Finally, we have two lines to add. Back in the **OnPlayerKilled** function, we need to set the **bestScoreText** color to the **bestScoreColor**.
And the same in the **ResetGame** function, but we are resetting it back to normal.

```
91
                1 reference
92              void OnPlayerKilled()
93              {
94                  spawner.active = false;
95                  gameStarted = false;
96
97                  splash.SetActive(true);
98
99                  Invoke("SplashScreen", 2);
100
101                 if(score > bestScore)
102                 {
103                     bestScoreText.color = bestScoreColor;
104
105                     bestScore = score;
106                     PlayerPrefs.SetInt("BestScore", bestScore);
107                     beatBestScore = true;
108                     bestScoreText.text = "Best Score: " + bestScore.ToString();
109                 }
110             }
111
                1 reference
112             void ResetGame()
113             {
114                 bestScoreText.color = normalColor;
115
116                 spawner.active = true;
117                 title.SetActive(false);
118
119                 splash.SetActive(false);
120
121                 scoreText.enabled = true;
122                 score = 0;
123
124                 scoreText.text = "Score: " + score.ToString();
125
```

11 Save your project and play your game. Notice how the best score from the last time you played is still there? And now when you beat the best score, the color of the text changes to red.

At this point, Dropping Bombs is complete! If there is anything else you wish to change about your UI, colors, or Spawner, now is the time to do it!