```
1  using System;
2  using UnityEngine;
3  using UnityEngine.UI;
4
5  /// <summary>
6  /// GuideHandler is assigned to its respective prefab, allowing for
     transitions between windows and tabs.
7  /// </summary>
8  public class GuideHandler : MonoBehaviour
9  {
10     /// <summary>
11     /// The button color of a selected and unselected tab respectively.
12     /// </summary>
13     [SerializeField]
14     Color selectedTabColor, unselectedTabColor;
15
16     /// <summary>
17     /// The button backgrounds of each window.
18     /// </summary>
19     [SerializeField]
20     Image welcomeWindow, logicGatesWindow, controlsWindow;
21
22     /// <summary>
23     /// The button backgrounds of the tabs within each window.
24     /// </summary>
25     [SerializeField]
26     Image[] welcomeTabs, logicGatesTabs, controlsTabs;
27
28     /// <summary>
29     /// The GameObjects that contains all tabs of each window
         respectively.
30     /// </summary>
31     [SerializeField]
32     GameObject welcomeTabsParent, logicGatesTabsParent,
         controlsTabsParent;
33
34     /// <summary>
35     /// The view areas of the tabs within each window.
36     /// </summary>
37     [SerializeField]
38     GameObject[] welcomeTabsViews, logicGatesTabsViews, controlsTabsViews;
39
40     /// <summary>
41     /// The background button of the current window that is viewable.
42     /// </summary>
43     private Image currentWindow;
44
45     /// <summary>
46     /// Displays the index of the currently opened tab for each window.
```

```
47      /// </summary>
48      private int currentWelcomeTab = 0, currentLogicGatesTab = 0,
          currentControlsTab = 0;
49
50      // Initializes the default window as the welcome window.
51      private void Start() { currentWindow = welcomeWindow; }
52
53      /// <summary>
54      /// Switches to a new tab for the current window in use.
55      /// </summary>
56      /// <param name="newTab">The index of the tab to switch to.</param>
57      public void UpdateTab(int newTab)
58      {
59          GameObject currentView, newView; // The view areas to turn off and
              on respectively.
60          Image currentButton, newButton; // The button backgrounds to color
              unselected and selected respectively.
61
62          // Populates initialized variables based on current window.
63          if (currentWindow == welcomeWindow)
64          {
65              if (newTab == currentWelcomeTab) return;
66
67              currentButton = welcomeTabs[currentWelcomeTab];
68              newButton = welcomeTabs[newTab];
69              currentView = welcomeTabsViews[currentWelcomeTab];
70              newView = welcomeTabsViews[newTab];
71              currentWelcomeTab = newTab;
72          }
73
74          else if (currentWindow == logicGatesWindow)
75          {
76              if (newTab == currentLogicGatesTab) return;
77
78              currentButton = logicGatesTabs[currentLogicGatesTab];
79              newButton = logicGatesTabs[newTab];
80              currentView = logicGatesTabsViews[currentLogicGatesTab];
81              newView = logicGatesTabsViews[newTab];
82              currentLogicGatesTab = newTab;
83          }
84
85          else if (currentWindow == controlsWindow)
86          {
87              if (newTab == currentControlsTab) return;
88
89              currentButton = controlsTabs[currentControlsTab];
90              newButton = controlsTabs[newTab];
91              currentView = controlsTabsViews[currentControlsTab];
92              newView = controlsTabsViews[newTab];
```

```
93              currentControlsTab = newTab;
94          }
95
96          // Incorrent window currently in use
97          else throw new Exception("Invalid current window.");
98
99          // Updates the obtained values
100         currentButton.color = unselectedTabColor;
101         newButton.color = selectedTabColor;
102         currentView.SetActive(false);
103         newView.SetActive(true);
104     }
105
106     /// <summary>
107     /// Switches to a new window.
108     /// </summary>
109     /// <param name="newWindow">The new window to switch to.</param>
110     public void UpdateWindow(Image newWindow)
111     {
112         if (currentWindow == newWindow) return;
113
114         // Updates the button colors.
115         currentWindow.color = unselectedTabColor;
116         newWindow.color = selectedTabColor;
117
118         // Makes the current window invisible.
119         if (currentWindow == welcomeWindow) welcomeTabsParent.SetActive
              (false);
120
121         else if (currentWindow == logicGatesWindow)
              logicGatesTabsParent.SetActive(false);
122
123         else if (currentWindow == controlsWindow)
              controlsTabsParent.SetActive(false);
124
125         else throw new Exception("Invalid current window.");
126
127         // Makes the new window visible.
128         if (newWindow == welcomeWindow) welcomeTabsParent.SetActive(true);
129
130         else if (newWindow == logicGatesWindow)
              logicGatesTabsParent.SetActive(true);
131
132         else if (newWindow == controlsWindow) controlsTabsParent.SetActive
              (true);
133
134         else throw new Exception("Invalid new window.");
135
136         currentWindow = newWindow;
```

```
137     }
138 }
```