```
1  using System;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  /// <summary>
6  /// PreviewStructure contains all serializable values to restore a preview
     scene.
7  /// </summary>
8  [Serializable]
9  public class PreviewStructure
10 {
11     /// <summary>
12     /// The list of circuit identifiers pertaining to each circuit within
         the scene.
13     /// </summary>
14     [SerializeField]
15     List<CircuitIdentifier> circuits = new List<CircuitIdentifier>();
16
17     /// <summary>
18     /// The unique ID assigned to this preview structure.<br/><br/>
19     /// Functionally, this ID is utilized to access the specific folder
         under which the connection and save information of the preview
         structure is.
20     /// </summary>
21     [SerializeField]
22     int id;
23
24     /// <summary>
25     /// The order in which empty inputs and outputs were selected by the
         user.<br/><br/>
26     /// Functionally, a visualized custom circuit will output these inputs
         and outputs in their selected order (bottom to top).
27     /// </summary>
28     [SerializeField]
29     List<int> inputOrders,
30         outputOrders;
31
32     /// <summary>
33     /// Identifying list of connections that exist within the custom
         circuit.
34     /// </summary>
35     [SerializeField]
36     List<InternalConnection> connections;
37
38     /// <summary>
39     /// Name of the custom circuit.
40     /// </summary>
41     [SerializeField]
42     string name;
```

```
43
44      /// <summary>
45      /// The corresponding user-assigned label for each empty input/output.
46      /// </summary>
47      [SerializeField]
48      List<string> inputLabels,
49          outputLabels;
50
51      /// <summary>
52      /// Location of the camera within the editor scene.
53      /// </summary>
54      [SerializeField]
55      Vector3 cameraLocation;
56
57      public PreviewStructure(string name) { this.name = name; }
58
59      /// <summary>
60      /// Internal class utilized to obtain the custom circuit ID via in-
          scene raycasting.
61      /// </summary>
62      public class PreviewStructureReference : MonoBehaviour
63      {
64          private int id;
65
66          public int ID { get { return id; } set { id = value; } }
67      }
68
69      // Getter and setter methods
70      public List<CircuitIdentifier> Circuits { get { return circuits; } set
          { circuits = value; } }
71
72      public int ID { get { return id; } set { id = value; } }
73
74      public List<int> InputOrders { get { return inputOrders; } set
          { inputOrders = value; } }
75
76      public List<int> OutputOrders { get { return outputOrders; } set
          { outputOrders = value; } }
77
78      public List<InternalConnection> Connections { get { return
          connections; } set { connections = value; } }
79
80      public List<string> InputLabels { get { return inputLabels; } set
          { inputLabels = value; } }
81
82      public List<string> OutputLabels { get { return outputLabels; } set
          { outputLabels = value; } }
83
84      public Vector3 CameraLocation { get { return cameraLocation; } set
```

```
        { cameraLocation = value; } }

    // Getter method
    public string Name { get { return name; } }
}
```