

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class BuildRoom : MonoBehaviour
6  {
7      Touch touch;
8      GameObject enterButton, exitButton;
9      LayerMask UI;
10
11     [SerializeField] private GameObject room;           //
12     Prefab here
13     #pragma warning disable
14     [SerializeField] private int wood, stone, metal, food, water, dollar; //
15     Removed warning for unassigned vars as they are done through the inspector.
16     #pragma warning enable
17
18     private void Awake()
19     {
20         // We do this since one is always disabled
21         enterButton = GameObject.Find("MenuUI").transform.GetChild
22         (1).gameObject; // Finds the enter button
23         exitButton = GameObject.Find("MenuUI").transform.GetChild
24         (2).gameObject; // Finds the exit button
25         UI = LayerMask.GetMask("UI");
26     }
27     void Update()
28     {
29         if (Input.touchCount > 0)
30         {
31             touch = Input.GetTouch
32             (0); //
33             Grab touch and store it in our touch var.
34
35             if (touch.phase == TouchPhase.Began)
36             {
37                 Vector3 touchPos = Camera.main.ScreenToWorldPoint
38                 (touch.position); // Take the touch var we
39                 stored and convert it into a 2d vector.
40                 Vector2 touchPos2D = new Vector2(touchPos.x, touchPos.y);
41
42                 Ray ray = new Ray(touchPos2D, new Vector3(0, 0,
43                 1)); // Create a ray using our
44                 vector2D made from touch var.
45
46                 RaycastHit2D touchCollider = Physics2D.GetRayIntersection(ray,
47                 Mathf.Infinity, UI); // We use GetRayIntersection here since we
48                 are dealing with multiple colliders that are
49

```

```

        // overlapping and both need to be interactable.
40
41         if (touchCollider.collider.gameObject.tag ==
            "bNode") // Checks whether or not
            we hit a build node.
42     {
43         //Debug.Log("hit Bnode");
44
45         GameObject[] bNodes = GameObject.FindGameObjectsWithTag
            ("bNodeParent"); // We disable bNodeParents here.
46         foreach (GameObject x in bNodes)
47             x.SetActive(false);
48
49
50         Transform bNodeTransform = touchCollider.collider.transform;
51         Instantiate(room, bNodeTransform.position,
            bNodeTransform.rotation); // Timer, room function, etc
            handled on prefab scripts.
52
53         GameObject selectedBNode =
            touchCollider.collider.gameObject; // Disable the
            selected bNode and if possible give the bnode info(gameobject)
            to the prefab so we will have
54         selectedBNode.SetActive
            (false); // easier
            time with the destruction script of a room where we will
            reenale the bnode.
55
56         Resources.subtractResources(wood, stone, metal, food, water,
            dollar); // Take away resources
57
58         gameObject.SetActive
            (false); // Disable
            the parent gameobject of this script(BuildRoom)
59     }
60 }
61 }
62 }
63 private void OnDisable()
64 {
65     enterButton.SetActive(true);
66     exitButton.SetActive(false);
67     gameObject.SetActive
        (false); //
        Ensures script(BuildRoom) is disabled.
68 }
69 }
70
71
72

```