

~\OneDrive - St Paul's Catholic College\Documents\2D Strategy Game -
Liberator\Assets\Scripts\Hexes\HexData.cs

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using TMPro;
4 using UnityEngine;
5 using UnityEngine.UI;
6 public class HexData : MonoBehaviour
7 {
8     public int horizontalCoordinate;
9     public int verticalCoordinate;
10    public bool isSecondLevel = false; // taking into account the higher level terrain hex
11    public ClickOnHex clickOnHex;
12    public Image Landscape; // Landscape image of each prefab in the battlehex script which
    goes for each hex
13    public Distance distanceText; // access to DistanceText Object
14    [SerializeField] protected Image currentState;
15    public bool isStartingHex = false; // define if a hex is a starting position for a
    character
16    public bool isNeighbouringHex = false;
17    public bool isIncluded = false; // helps to define a hex as available position
18
19
20    private void Awake()
21    {
22        clickOnHex = GetComponent<ClickOnHex>(); // gets the hex the player clicked on
23    }
24
25    public virtual void MakeMeAvailable()
26    {
27        currentState.sprite = clickOnHex.fieldManager.availableToMove; // sets the white
    frame to a hex
28        currentState.color = new Color32(255,255,255,255);
29    }
30
31    public virtual void MakeMeTargetToMove()
32    {
33        clickOnHex.isTargetToMove = true;
34        Controller.targetToMove = this; // the keyword this refers to the current instance
    of the class
35        currentState.sprite = clickOnHex.fieldManager.availableAsTarget; // sets the green
    frame to a hex
36    }
37
38    public void DefineMeAsStartingHex() // defines this hex as starting position
39    {
40        distanceText.distanceFromStartingPoint = 0;
41        isStartingHex = true;
42        distanceText.stepsToGo = 1; // to get out from deserts and swamps, so that the
    first step takes only 1 unit of the soldiers velocity
43    }
44
45    public virtual bool AvailableToGround()
46    {
47        return true;
48    }
49
50    public void SetDefaultValue()
51    {
52        isStartingHex = false;
```

```
53 |         isNeighbouringHex = false;
54 |         isIncluded = false;
55 |         distanceText.GetComponent<TextMeshProUGUI>().color = new Color32(255,255,255,0); //
    | hides numbers
56 |         currentState.color = new Color32(255,255,255,0);
57 |         Landscape.color = new Color32(255,255,255,255); // hides optimal path
58 |         distanceText.distanceFromStartingPoint = 20;
59 |         distanceText.stepsToGo = 1;
60 |     }
61 | }
62 |
```