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

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class AvailablePos : MonoBehaviour
6  {
7
8      private int step; // counts iterations
9      List<HexData> initalHexes = new List<HexData>(); // collects neighbouring hexes for
   evaluated hex
10
11     internal List<HexData> GetNewInitialHexes() // collects objects whose neighbours need
   to be found
12     {
13         initalHexes.Clear(); // empty the array before filling it again
14         foreach (HexData hex in FieldManager.allHexesArray)
15         {
16             if (hex.isNeighbouringHex & !hex.isIncluded) // eliminates unnecessary hexes
17             {
18                 initalHexes.Add(hex);
19             }
20         }
21         return initalHexes;
22     }
23      public void GetAvailablePositions(HexData startingHex, int stepsLimit, IAdjacentFinder
   AdjFinder)
24     {
25         AdjFinder.GetAdajacentHexesExtended(startingHex); // looks for hexes adjacent to
   starting hex.
26         // runs iterations to find all positions available. steps = number of iterations
27         for (step = 2; step <= stepsLimit; step++)
28         {
29             initalHexes = GetNewInitialHexes(); // collects hexes ready for iteration
30             foreach(HexData hex in initalHexes)
31             {
32                 AdjFinder.GetAdajacentHexesExtended(hex); // defines neighbouring hexes for
   each hex in the collection
33                 hex.isIncluded = true; // defines evaluated hex as available position
34             }
35         }
36     }
37 }
38
```