RegionRoaming

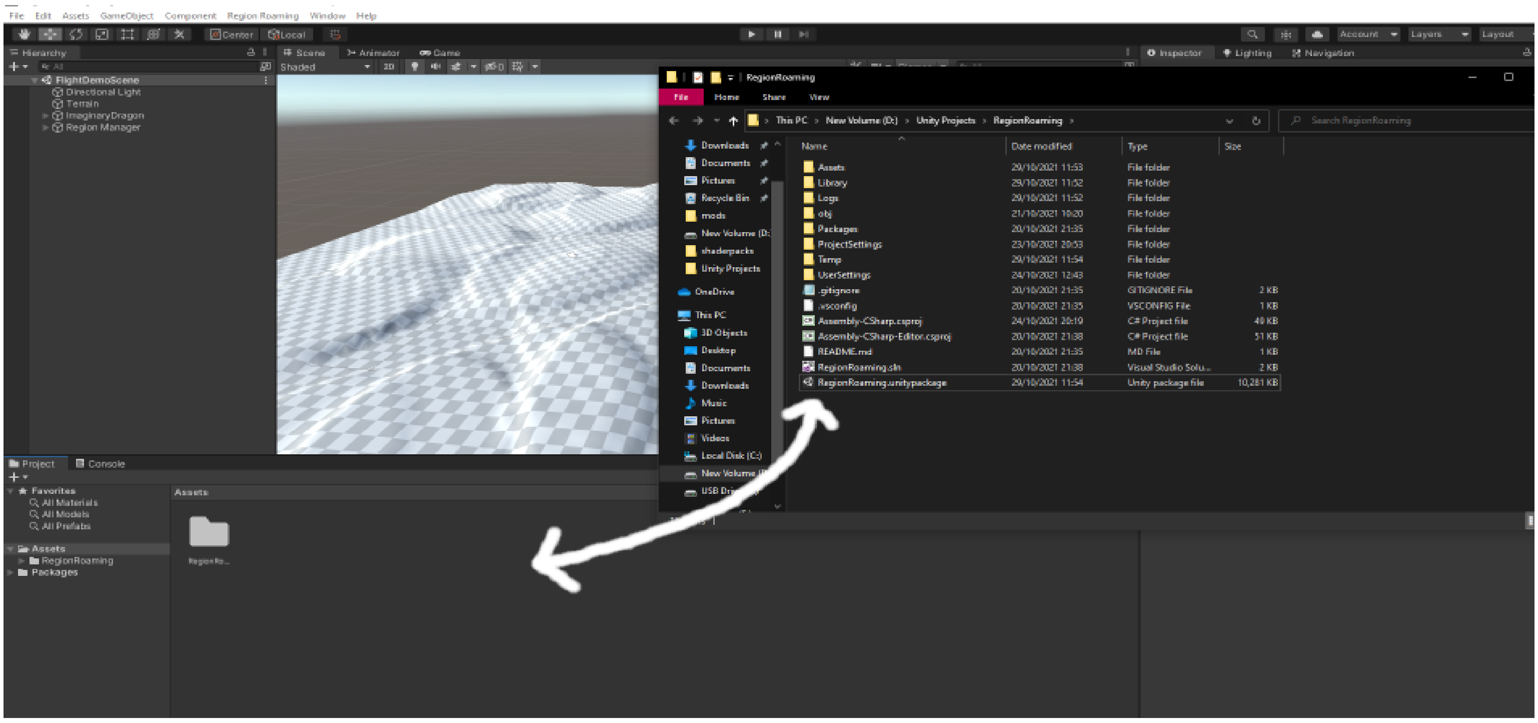
Introduction

Region Roaming is a vertical slice of an AI tool I proposed for my college course. While the tools shouldn't have any problematic bugs or glitches, I can't say for sure I will keep this version updated. I plan to create the full tool in the future.

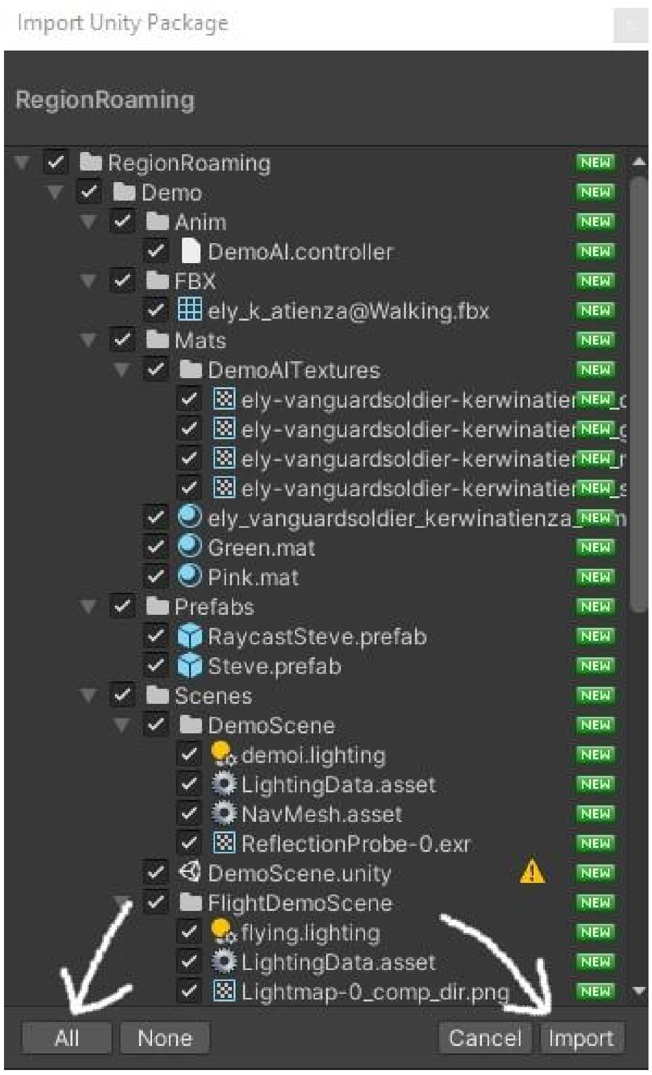
This tool does not create AI behaviour or movement, it simply provides mathematics or logical processes for the next location for the AI. The final product aims to add more life and reason to AI behaviour than simply providing them with mechanics alongside features.

Installation

Find the Unity Package file named Region Roaming and drag that into your project's asset folder. You should see a similar import box to when you install a tool of the assets store, make sure everything is ticked and click import.



Name	Date modified	Type	Size
Assets	29/10/2021 11:53	File folder	
Library	29/10/2021 11:52	File folder	
Logs	29/10/2021 11:52	File folder	
obj	21/10/2021 10:20	File folder	
Packages	20/10/2021 21:35	File folder	
ProjectSettings	23/10/2021 20:53	File folder	
Temp	29/10/2021 11:54	File folder	
UserSettings	24/10/2021 12:43	File folder	
.gitignore	20/10/2021 21:35	GITIGNORE File	2 KB
.vsconfig	20/10/2021 21:35	VSCONFIG File	1 KB
Assembly-CSharp.csproj	24/10/2021 20:19	C# Project file	49 KB
Assembly-CSharp-Editor.csproj	20/10/2021 21:38	C# Project file	51 KB
README.md	20/10/2021 21:35	MD File	1 KB
RegionRoaming.sln	20/10/2021 21:38	Visual Studio Solu...	2 KB
RegionRoaming.unitypackage	29/10/2021 11:54	Unity package file	10,281 KB



```
RegionRoaming
  RegionRoaming
    Demo
      Anim
        DemoAI.controller
      FBX
        ely_k_atienza@Walking.fbx
      Mats
        DemoAITextures
          ely-vanguardsoldier-kerwinatier
          ely-vanguardsoldier-kerwinatier
          ely-vanguardsoldier-kerwinatier
          ely-vanguardsoldier-kerwinatier
        ely_vanguardsoldier_kerwinatienza_
        Green.mat
        Pink.mat
      Prefabs
        RaycastSteve.prefab
        Steve.prefab
      Scenes
        DemoScene
          demoi.lighting
          LightingData.asset
          NavMesh.asset
          ReflectionProbe-0.exr
        DemoScene.unity
        FlightDemoScene
          flying.lighting
          LightingData.asset
          Lightmap-0_comp_dir.png
```

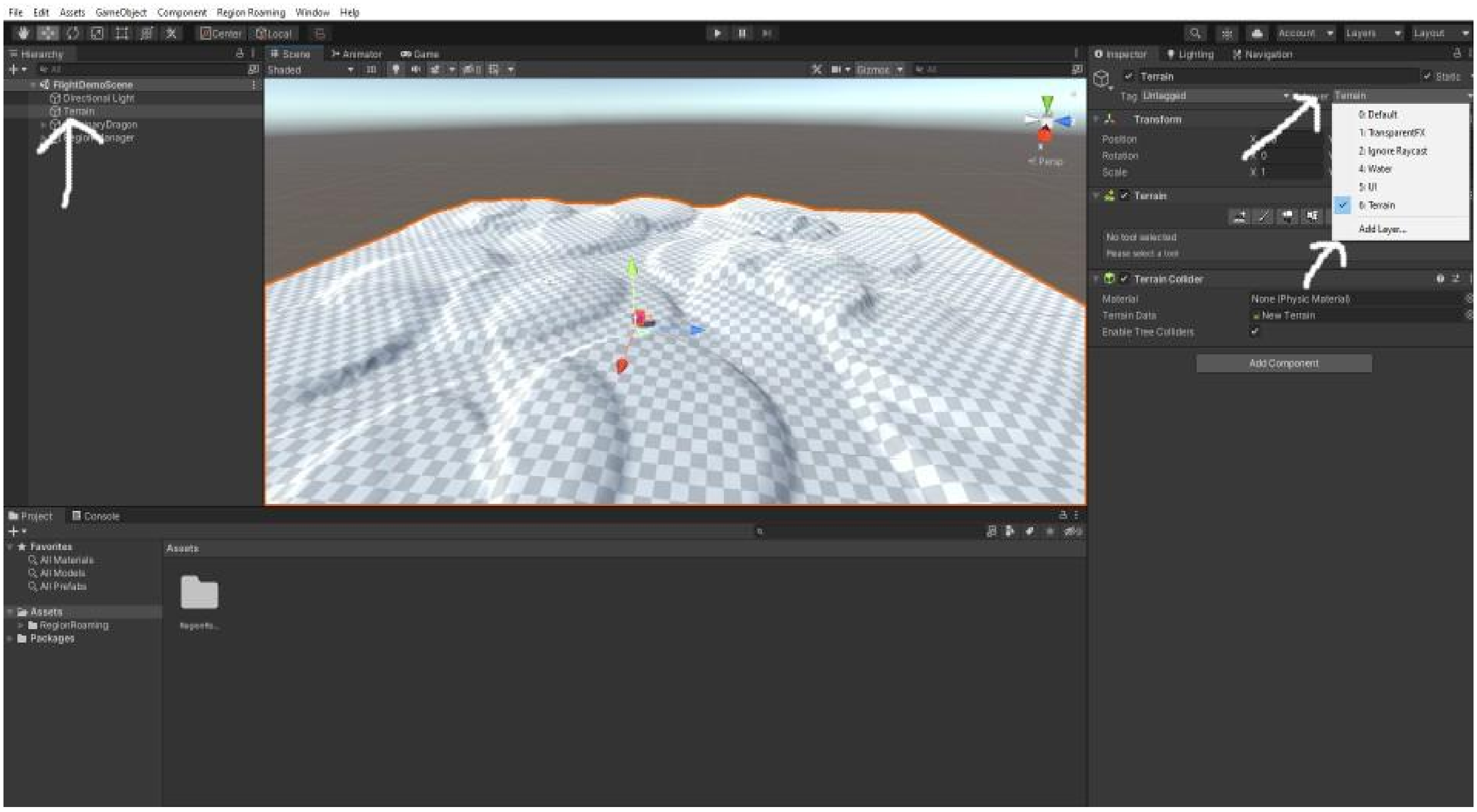
It's done, now you can make use of the tool. Follow the Guide section below for information on setting up and making use of the tool.

Guide

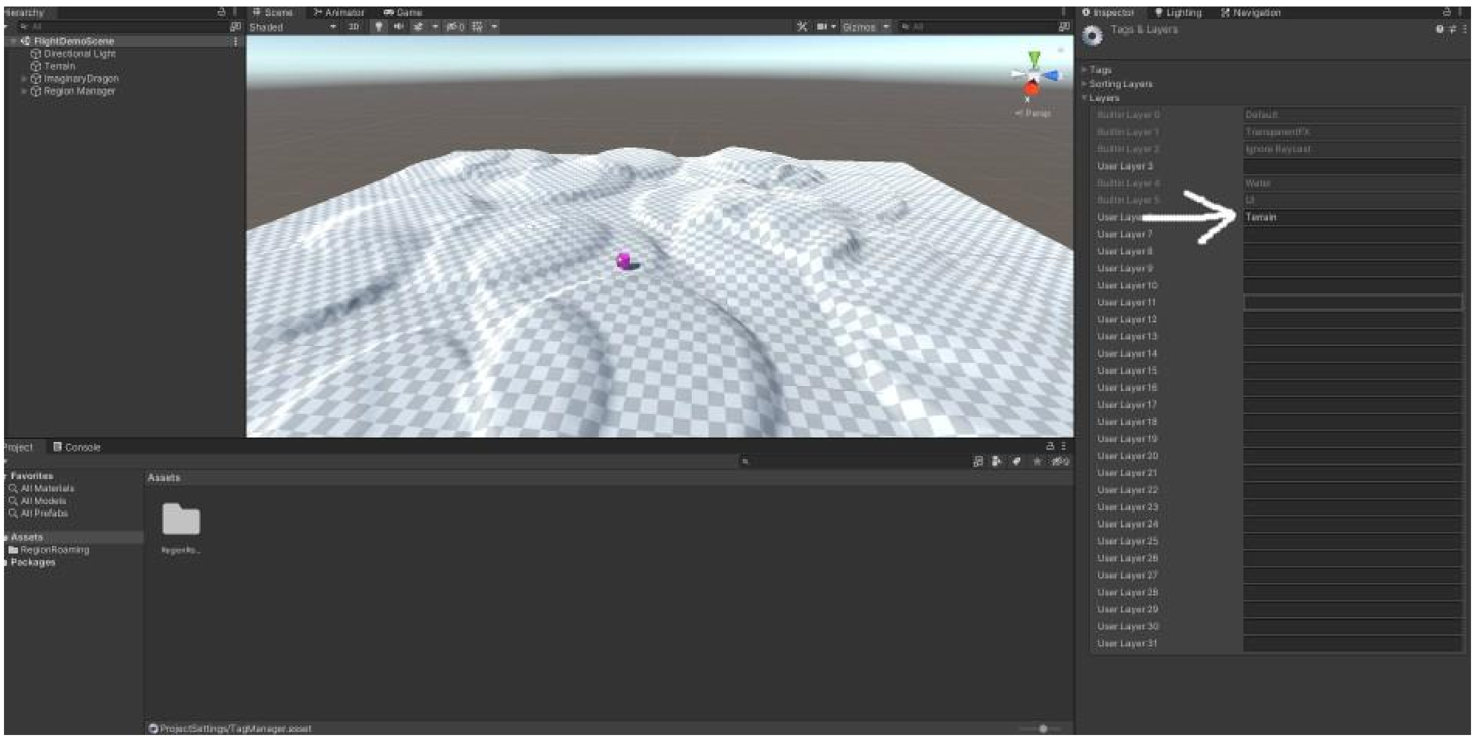
Set Up

There isn't much set up to the tool, I designed it to be as user friendly as possible. However, due to the features and ability to test them there is a single step to do.

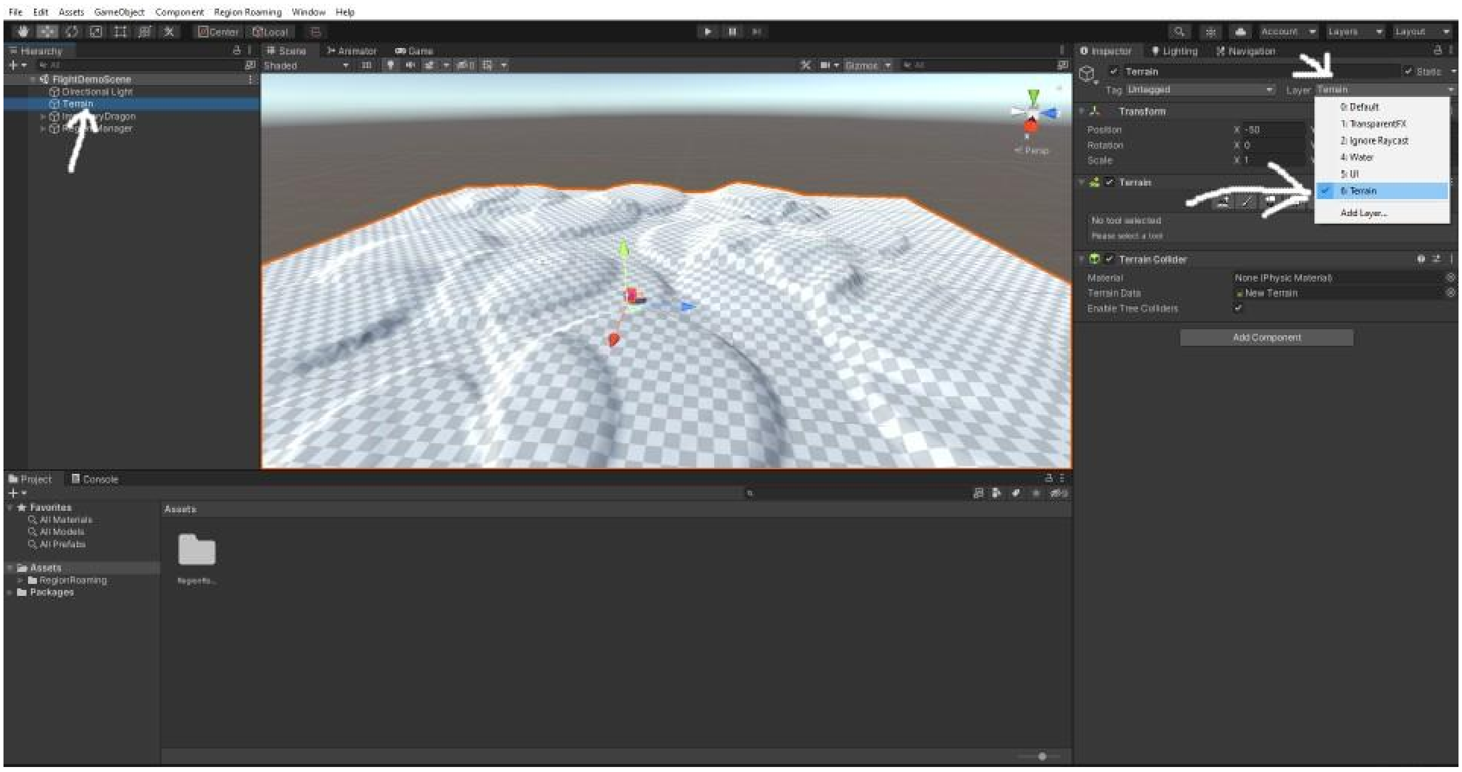
Navigate to the Layers tab by clicking on an object in the scene, clicking layers within the inspector and then Add New Layer.



Then click on any layer, number doesn't matter, and type in Terrain or terrain

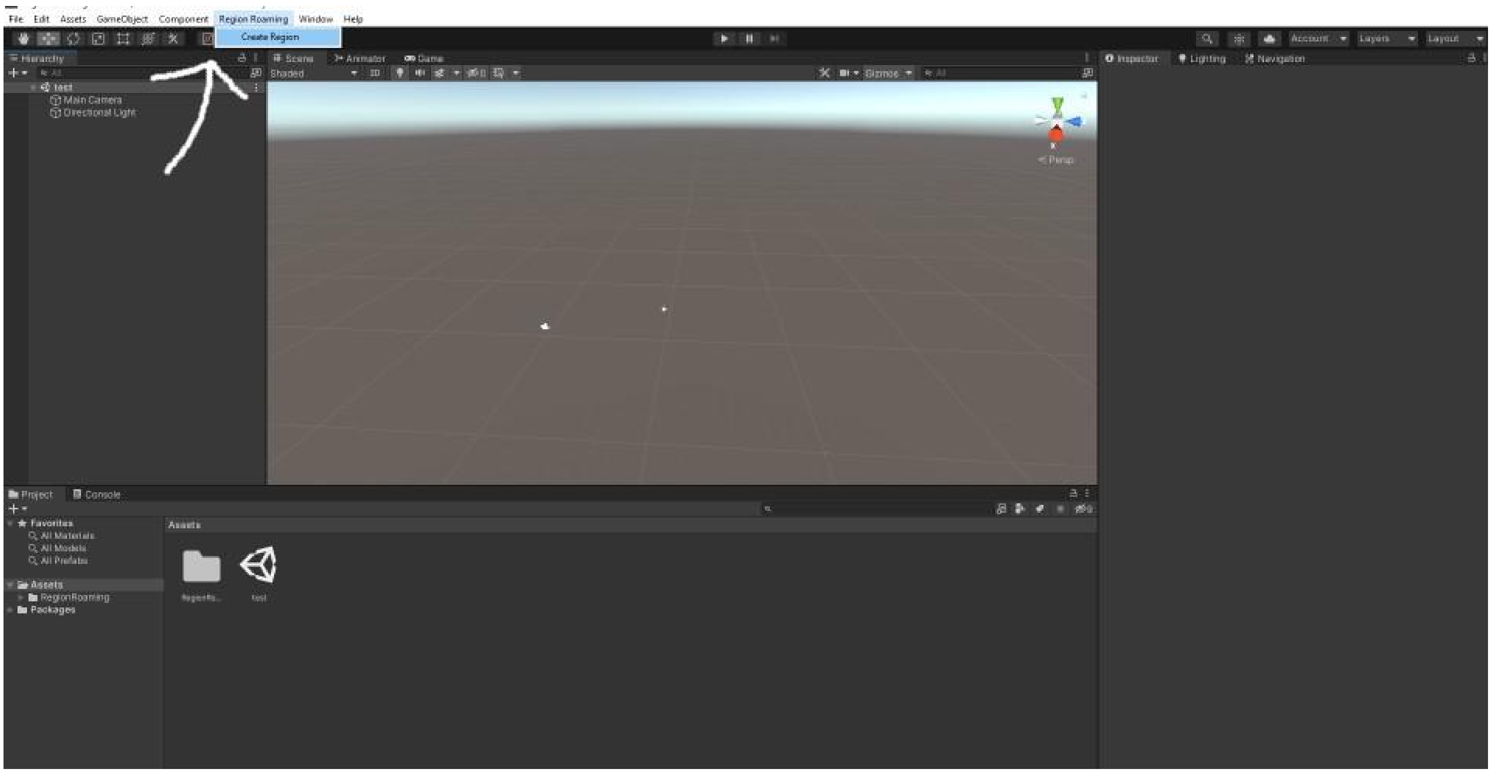


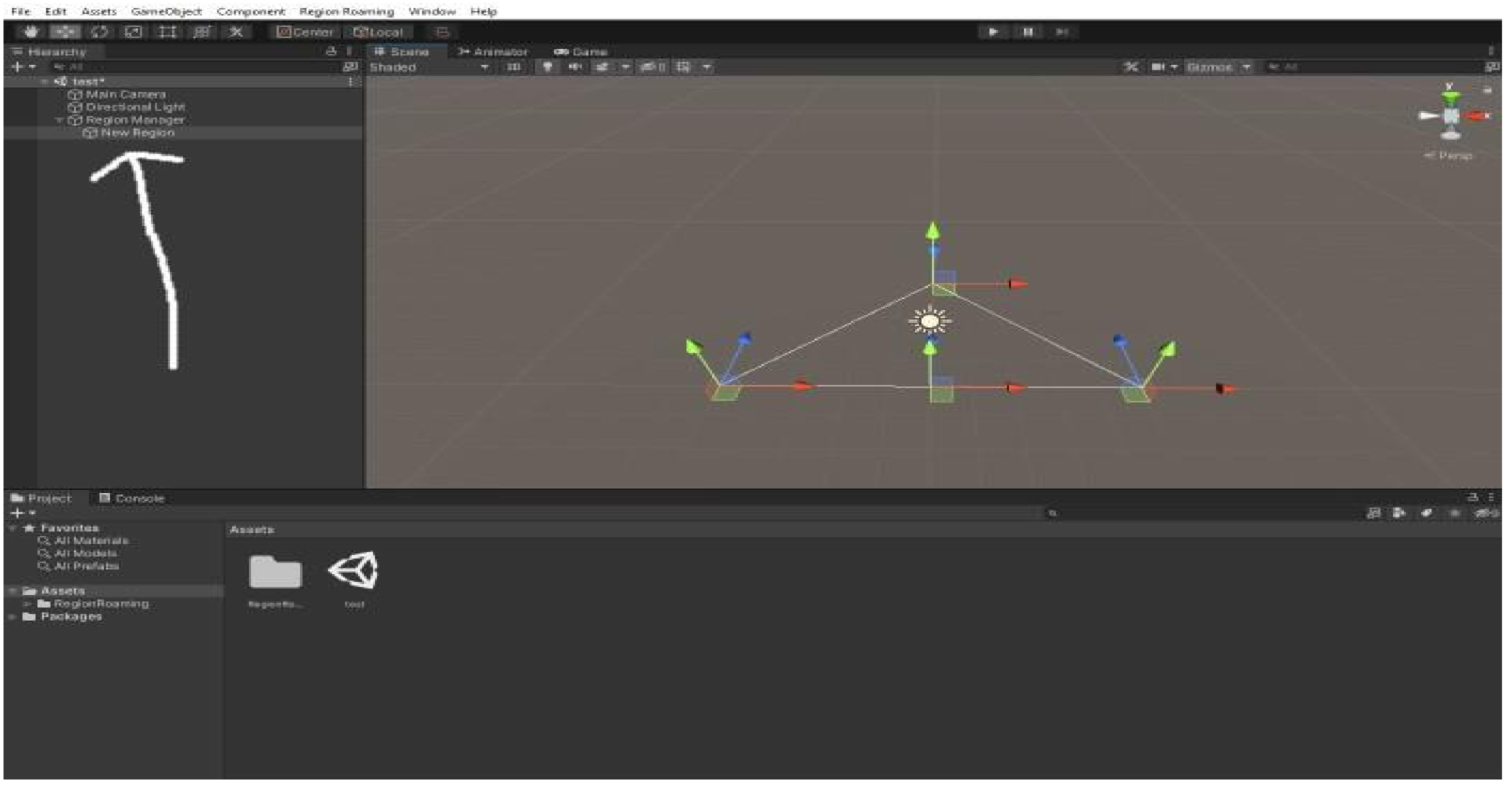
That's all the set up, Now to make use of the Raycast function and Flight function you will need to make the terrain or ground the AI can walk/fly over a terrain layer by once again selecting the object, clicking terrains and clicking the layer named terrain.



Region Creation and Customisation

To create a new region, navigate to the top menu bar and look for Region Roaming. Click on that button and it will open up a dropdown menu. Click on the Create New Region.

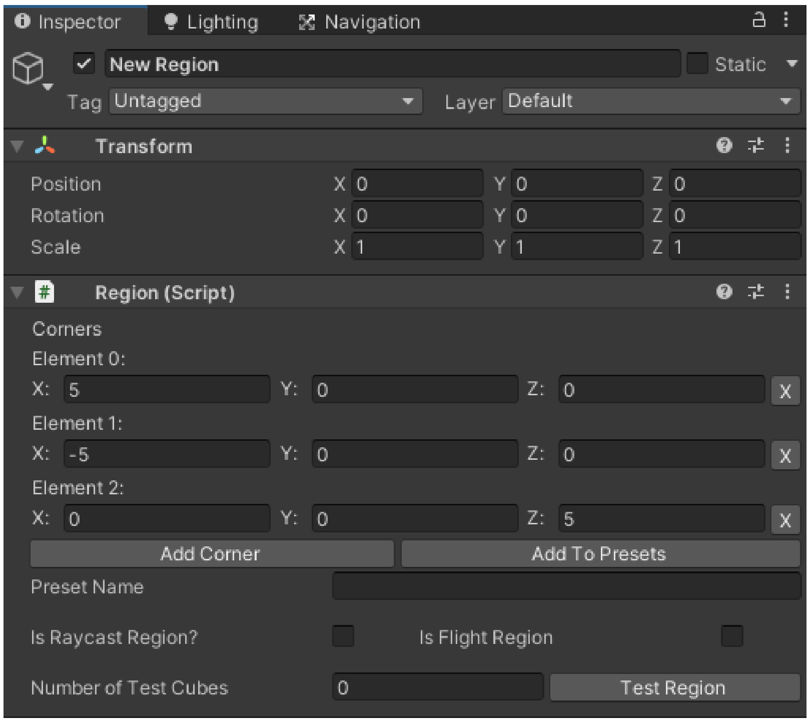




You should now see in your hierarchy that a New Region has been created alongside a Region Manager. The New Region is the region object used to define a space, the Region Manager holds presets and is a useful asset for keeping all your regions in one place.
You should keep all the regions within the Region Manager and keep the Region Manager in the scene. Bugs with the preset function will occur and possibly more, also your project will get quite messy full of regions if its removed.

Region Features

This is the inspector of the Region, I will give a quick rundown of the region inspector:



New Region
Static
Tag Untagged
Layer Default

Transform

Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Region (Script)

Corners

Element 0:
X: 5 Y: 0 Z: 0

Element 1:
X: -5 Y: 0 Z: 0

Element 2:
X: 0 Y: 0 Z: 5

Preset Name

Is Raycast Region?

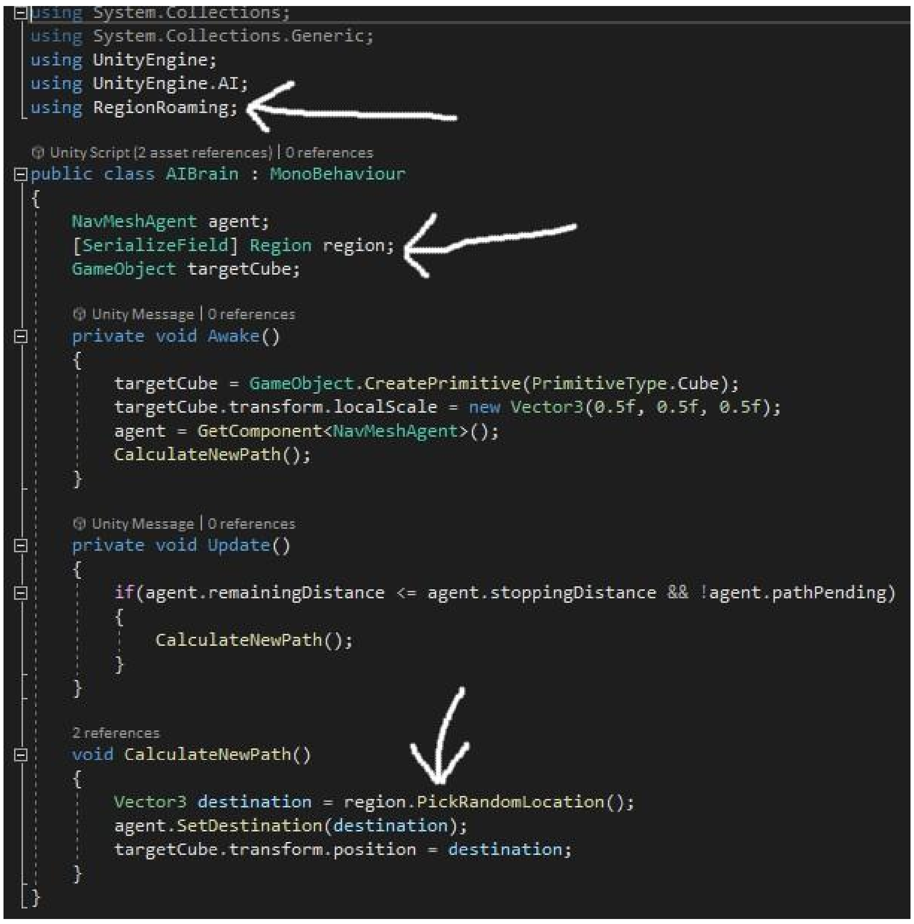
Is Flight Region

Number of Test Cubes 0

Elements - Elements 0 through 2 in this case is simply the number and reference to each corner. The XYZ below the element name is a breakdown of their position vector3 meaning if you change The X in the inspector, or scene view using handles, the element will update its position to the new value.
Add Corner Button - The Add Corner button will add a new corner/element to the region allowing for more dramatic shapes.
Add To Presets Button - Add to Presets button will add the current amount of corners and their location to a preset. This will allow you to load that preset for other regions so you don't need to painstakingly keep recreating it.
Preset Name - Preset name is the name of the preset you wish to save, input the name BEFORE you press Add to Presets.
Is Raycast Region - Is raycast is a testing feature bool, if the region is going to use the raycast function instead of the normal function, click this button. It will make the Test Region use the Raycast function instead of the normal to give accurate testing results.
Is Flight Region - The same as the Is Raycast above but for the flight function, click this for accurate testing if the region is going to be used for flight AI.
Number of Test Cubes - Number of Test Cubes is the amount of cubes you wish to spawn for a test run, this does create game objects so don't go overboard or crashes could occur.
Test Region Button - Runs a test of the region, spawning in the number of test cube input and updating their position to a random position within the selected region using the Raycast function or flight function if the bool is selected.

Using Regions

To make use of a region for an AI you need to do 3 things:
Firstly, use the Region Roaming namespace.
Secondly, Add a reference to a Region that is either public or a SerializedField.
Lastly, add your AI code using one of the following functions to get the AIs destination vector3 position.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.AI;
using RegionRoaming;

Unity Script (2 asset references) | 0 references
public class AIBrain : MonoBehaviour
{
    NavMeshAgent agent;
    [SerializeField] Region region;
    GameObject targetCube;

    Unity Message | 0 references
    private void Awake()
    {
        targetCube = GameObject.CreatePrimitive(PrimitiveType.Cube);
        targetCube.transform.localScale = new Vector3(0.5f, 0.5f, 0.5f);
        agent = GetComponent<NavMeshAgent>();
        CalculateNewPath();
    }

    Unity Message | 0 references
    private void Update()
    {
        if(agent.remainingDistance <= agent.stoppingDistance && !agent.pathPending)
        {
            CalculateNewPath();
        }
    }

    2 references
    void CalculateNewPath()
    {
        Vector3 destination = region.PickRandomLocation();
        agent.SetDestination(destination);
        targetCube.transform.position = destination;
    }
}
```

PickRandomLocation - This is the function to use with the Unity agent and navmesh system. It outputs a Vector3 where the Y is always 0, however the Unity navmesh system is able to take this point and get the relative point of the navmesh itself.
PickRandomRaycastLocation - If you are using custom AI logic use this function, this provides a random position similar to the PickRandomLocation function but with the added step of casting a ray from a high height down to the terrain to get the correct Y position. Requires the Terrain Layer to be set up.
PickRandomFlightLocation - similar to the PickRandomRaycastLocation but with the added functionality of getting random points in the air. If the generated Vector3 is not high enough off the terrain you get a position on the terrain, if the generated vector3 is high enough off the terrain it will output that vector3 at a flight position. minFlyingHeight is the minimum distance the vector needs to be from the terrain for

the vector to be in the air, maxFlyingHeight is the maximum the Y in vector will be allowed to go unrelated to terrain height at that position.

Advanced Usage

The tools also provide a small mathematical class and Triangle.NET library to make use of both of these. You need the script to use the name spaces Region Roaming Mathematics and TriangleNet.

These are the pure mathematical functions used for the tool but might be useful in other applications however reframe from changing these functions as they will affect the tool. These are all the functions and what they do:

Triangulate - Takes in a list of Vector3s, the points of the region, and returns a ICollection of Triangles, a data type from Triangle.NET. A second function exists taking in an IEnumerable of Vector2s, it creates a poly out of these Vector2s and triangulates that poly returning the ICollection of Triangles.

ToVertex - Takes in a Vector2 and converts it to a vertex, a data type of Triangle.NET.

ToVector2 - Takes in a Vertex and converts it to a Vector2.

TriArea - Takes in a triangle and using vector cross products calculates the triangles area.

ToVector2List - Takes in a triangle and converts each point into a vertex adding it to a list and returning the created list of triangle vertices.

Notes

This tool requires a terrain or floor, to create a terrain Right Click the hierarchy, 3D and then Terrain.

Make sure the region is laid out correctly, meaning the lines shown are not crossing.

For a more detailed look into the Package, go to the Wiki - https://github.com/TSGameDev/RegionRoaming/wiki/1---Introduction

Bug Report

This tool is a vertical slice and for a college course so I do not plan to keep this updated but I do plan to create a full version of this tool in the future.

If you wish to let me know of a bug for that future version then please send a message to my twitter https://twitter.com/TSGame_Dev.