

SETTTING UP TRAFFIC LIGHT CONTROL SYSTEM.

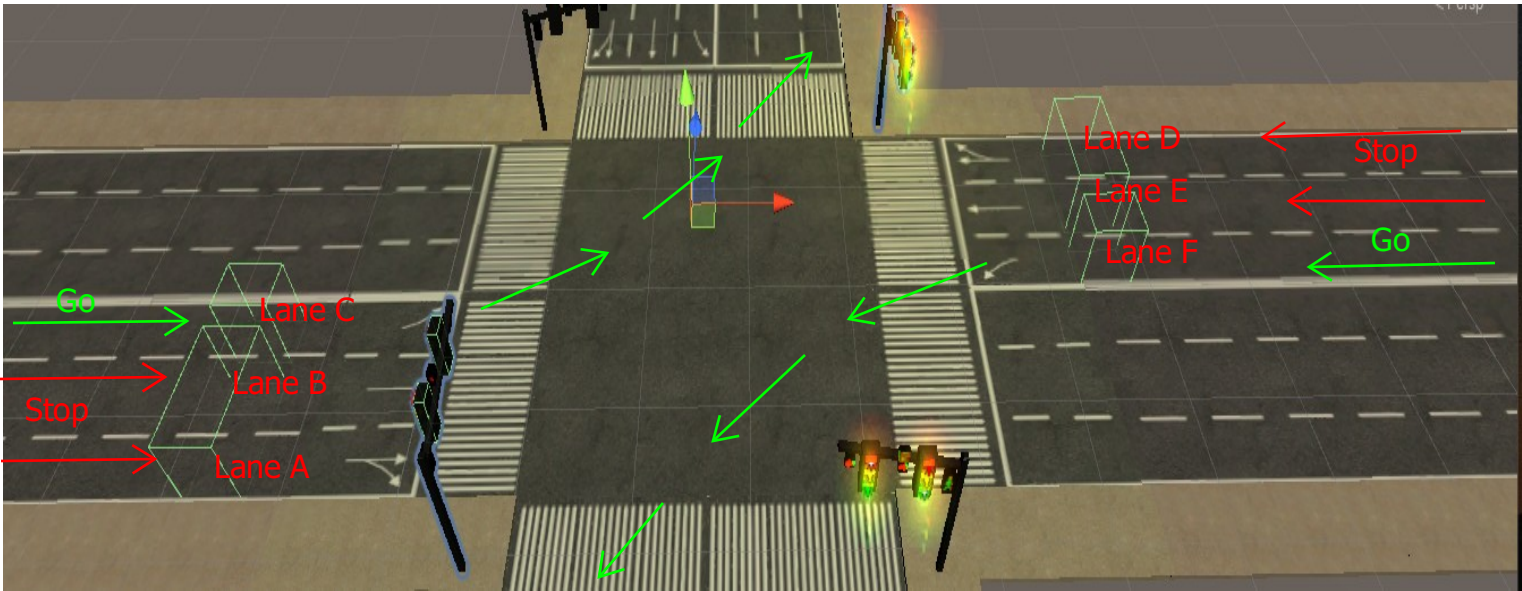
First of all take note of all the components needed to set up the traffic light control system which are listed below

- 1. ESTrafficLightCtrl.
- 2. ESTrafficControlPoint.
- 3. ESTrafficManager.

now lets discuss briefly on what each of those components do.

- ESTrafficLightCtrl : this component holds the red, yellow and green light gameobjects, its main function is to disable and enable the light gameobjects based on sensors and timer so its would be wise to child your lightflares,materials or any component that would be disabled or enabled with the light. for example when its time to go the green light gameobject would be enabled meaning all children objects would be enabled and same goes for the other way around.
- ESTrafficManager : this component contains a list of (ESTrafficLightCtrl). it controls the ESTrafficLightCtrl for example it has the ability to stop and move every ESTrafficLightCtrl component attached to it based on the command from the main controlpoint.
- ESTrafficControlPoint: this component holds all ESTrafficManager component attached to each road intersection so this means that every intersection that has two or more trafficlighs most have a control point. this component will send the stop and go command to each ESTrafficManager this technic plays a major role when dealing with complex roads intersections, lets say we have a junction were two roads meet with three lanes each, like the image below. Fig 1.0

Fig1.0



take note of lane naming in the diagram above

- 1, ← : red arrow means "stop"
- 2, ← : green arrow means "go"

Every lane has a collider which serves as a sensor and ESTrafficManager component attached to it. the control point will send stop and go command to each ESTrafficManager based on the setup

setup as follows:

- Lane A, B, E and D use same ESTrafficManager

- Lane C and F use same ESTrafficManager
(all together we got two ESTrafficManager)

Once in runtime ESTrafficControl sends Stop to the first ESTrafficManager which is Lane A, B, E and D and sends go command to the second which is Lane C and F. this simply means vehicles on Lane A, B, E and D have to wait for a while before the control point switches the command ;).

Now that we know what the three components do, next is how to actually set this up from scratch just follow the steps accurately.

Setting up Traffic Light system

Step 1 : lets say you got a road like the one in the image below



Fig 1.1

- Create a new GameObject call it traffic_controlpoint and place that it in the center of the intersection just to keep things organized.

- Add "ESTrafficControlPoint" component to the traffic_controlpoint gameobject.

-next you need a traffic light gameobject like the one the image below you find it the prefab folder



Fig1.3

-next drag this object place on the side of the road at the intersection,so it fits in like the above image .

-next take note of the lane naming in fig1.0 you will need it.

-select the traffic light gameobject on the right



Fig1.4

-make sure you child all the light gameobjects on the right to the selected traffic light gameobject , same thing goes to the left traffic light object.

-once you are ok with the child process, next is to add sensors to the traffic light (sensors are the built in colliders in unity). sensors gives feedback to incoming Ai vehicle about the current status of the traffic light.

-first we would add sensors to the traffic light on the right, but first observe that vehicles moving straight and right on opposite roads will have a less chance of collision if the both lanes are told to go at same time. explained in the below image.

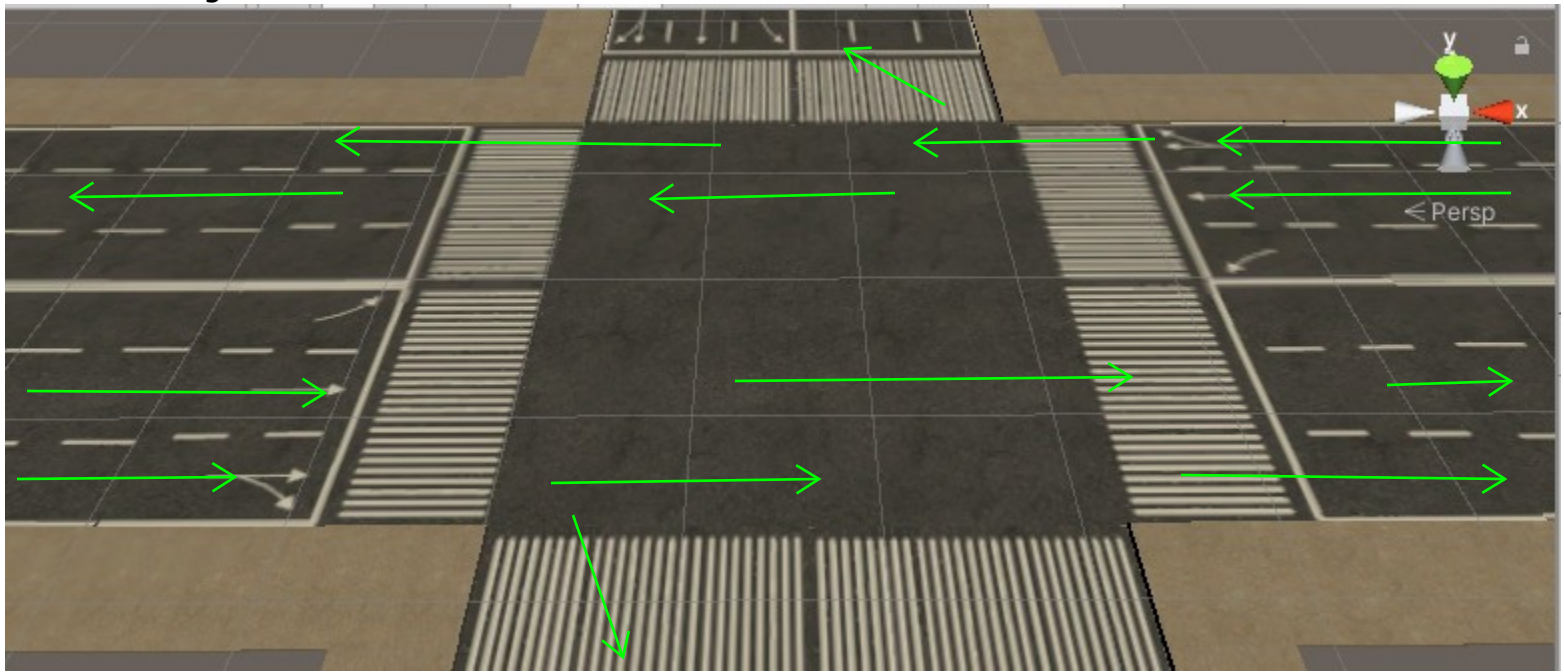


Fig1.5

-now select the traffic light on the right and add a box collider, set the collider position & scale similar to the one in image below.

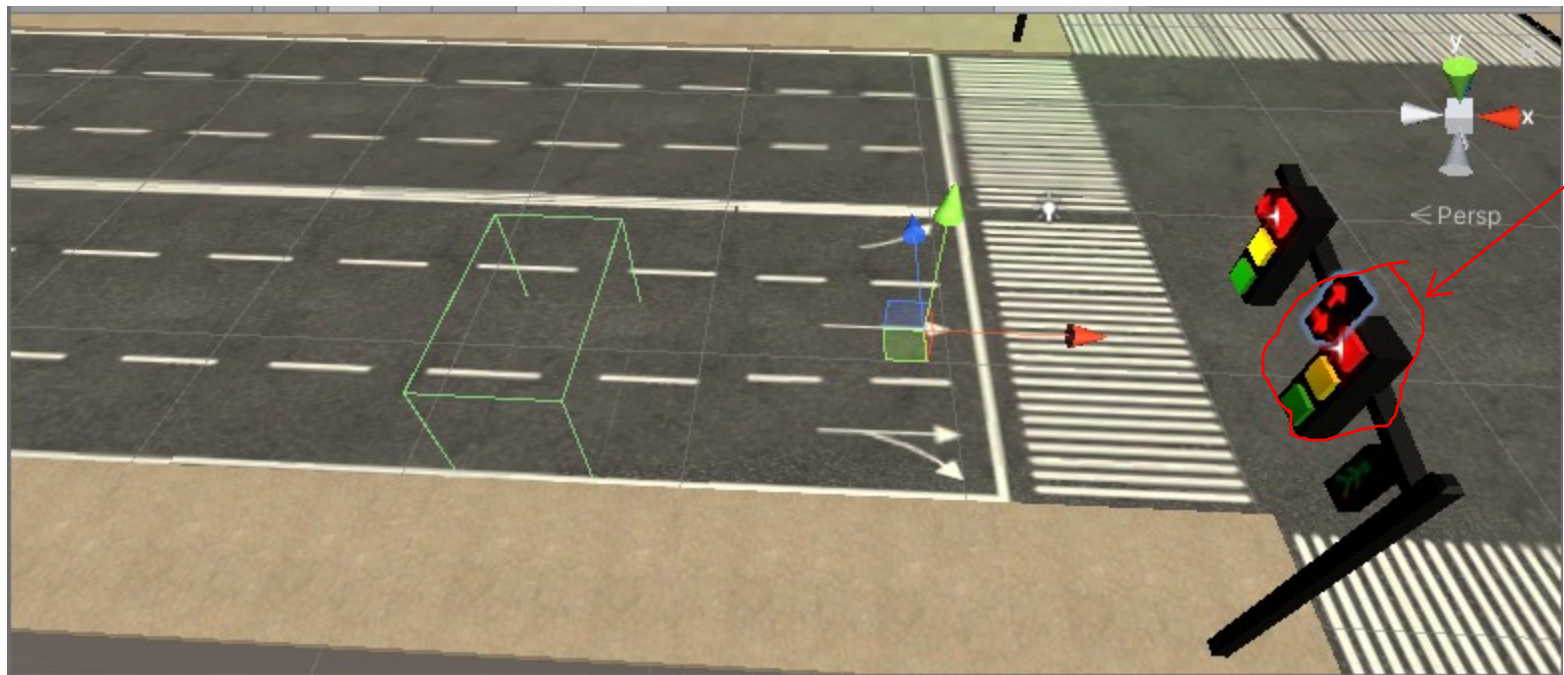


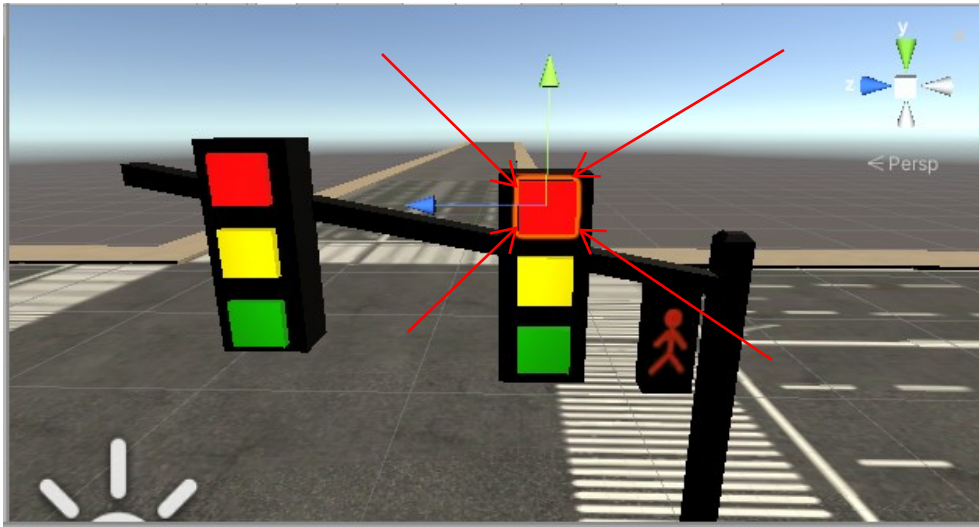
Fig1.6

-next change its tag name to "TrafficLight".

-next add ESTrafficLightCtrl component.

-now we add flares and light objects to the traffic light, this is not compulsory but adding these things will make your game more realistic.

-lets start with the red light object on the right select this object and create a new point light gameobject, then child the point light object to the red light gameobject and reset position.

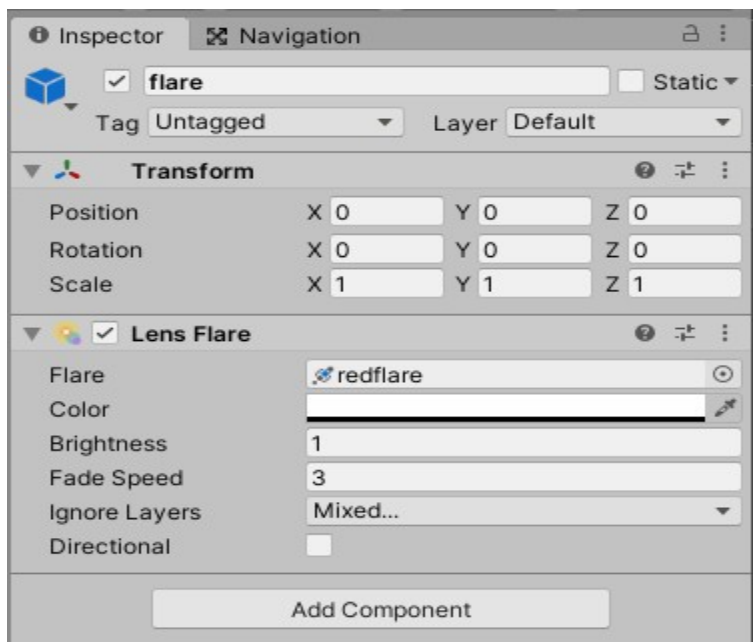


-next make the point light color red then resize to best fit.

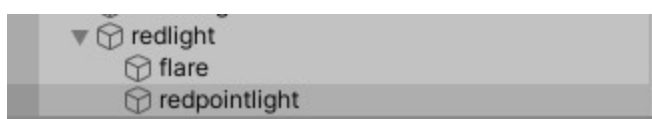
-next we add flares

-create a new empty gameobject call this flare and add the "Lens Flare" component

-drag and drop the redflare lens object in the asset folder into the flare slot. redlensflare is located at (ETS/assets/flare).



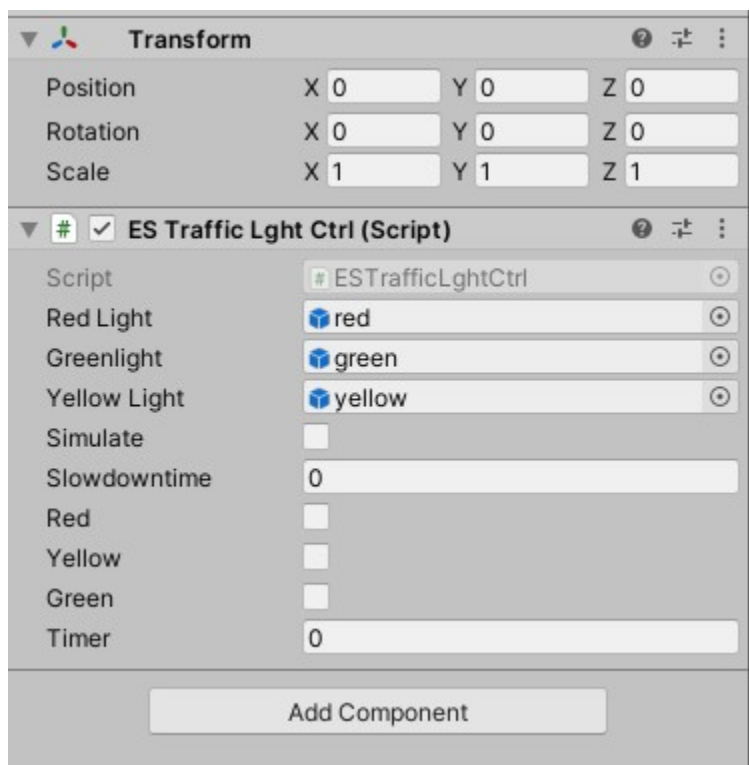
-next child the flare object to the red light gameobject and reset its position



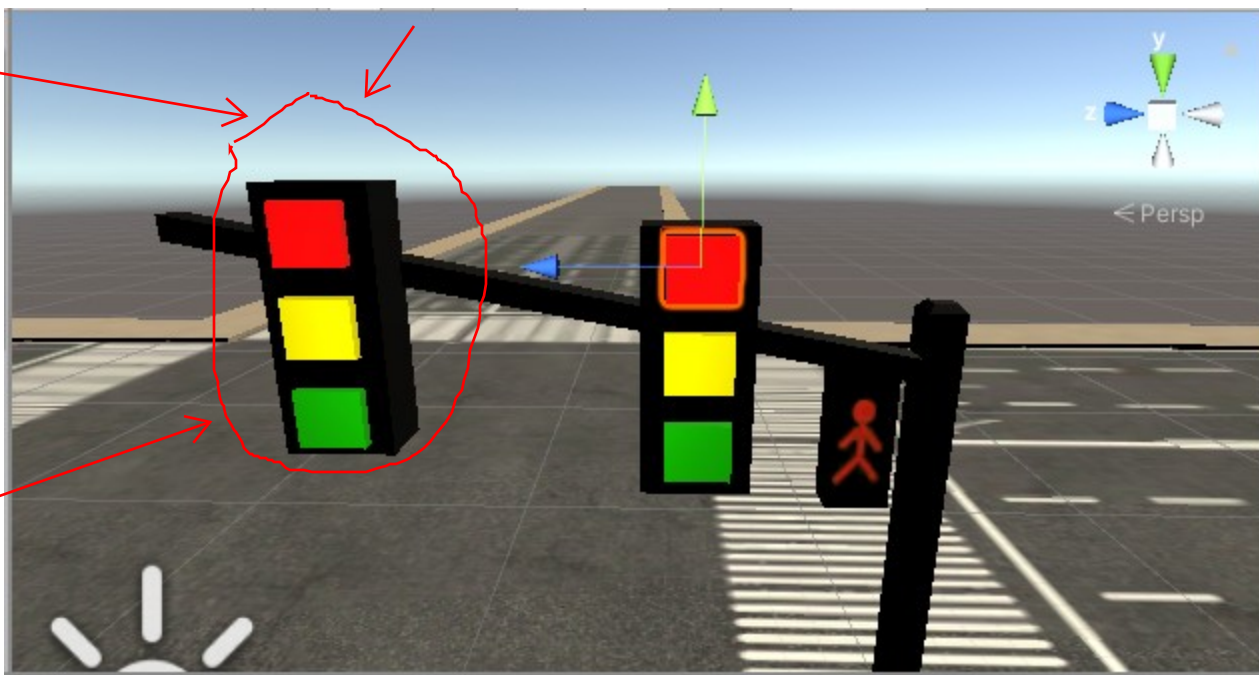
-next do the same for the yellow and green light objects.

-now we have all light gameobject on the right configured.

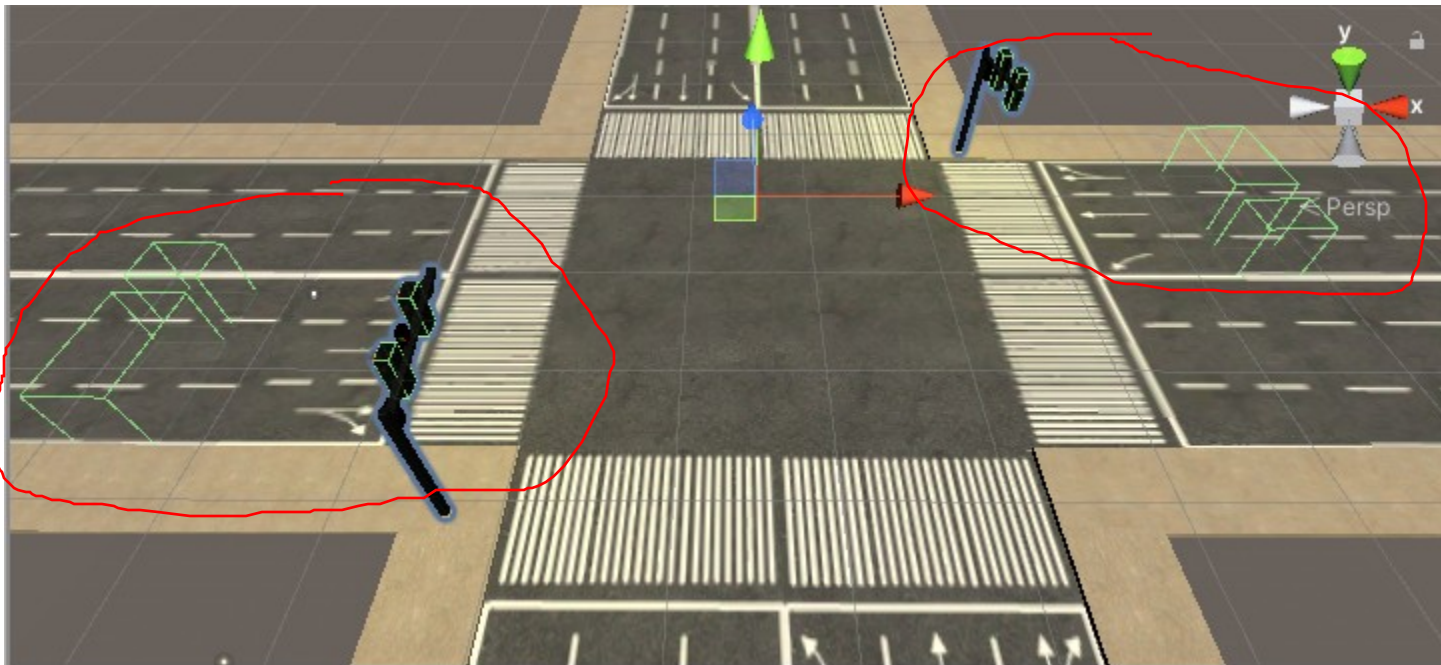
-next drag and drop each light object to the light slots in ESTrafficLightCtrl component. in Fig1.6



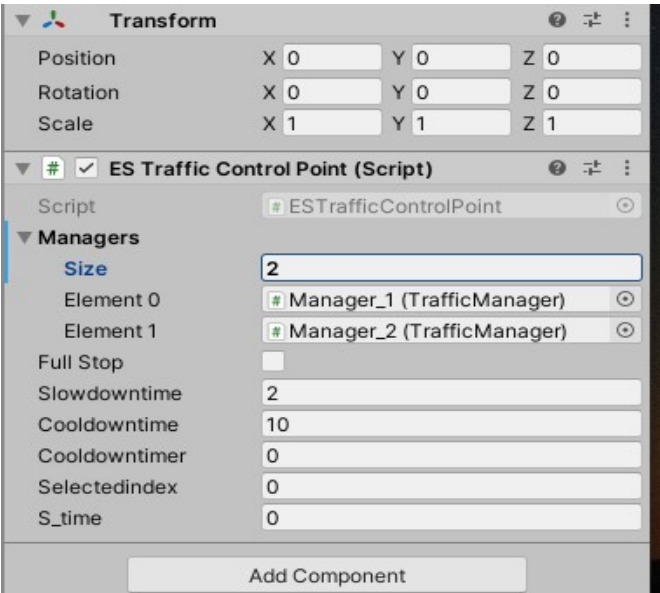
- Once you done you see all light objects in the light slot in the above image, leave all values at default.
- now you have sucessfully configure the traffic light for the right and straight lane.
- next move to the left traffic and follow on same process used for the right.



- Once you are done with the left traffic light.
- duplicate the traffic light parent and place the duplicate on the opposite road it should like the image below.



- now we are going to assign the traffic light gameobject to the main traffic control point
- now go back to fig1.0 and keep note of the lane naming.
- now select the traffic_controlpoint object and create two empty gameobject name the first "Manager_1" and the second Manager_2 and ESTrafficManager component to each managers.
- Manager_1 will control traffic lights for the lane going straight and right(Lane A,B,D andE). while Manager_2 will control traffic light on the lane thats going left only(Lane f and c).
- now drag and drop the traffic light controlling (LaneA, B and Lane D,E) into list of ESTrafficLight on the Manager_1 .
- Also drag and drop the traffic light controlling (lane F and C) to Manager_2.
- Finally select the traffic_controlpoint object and drag and drop the managers into the list of manager slot



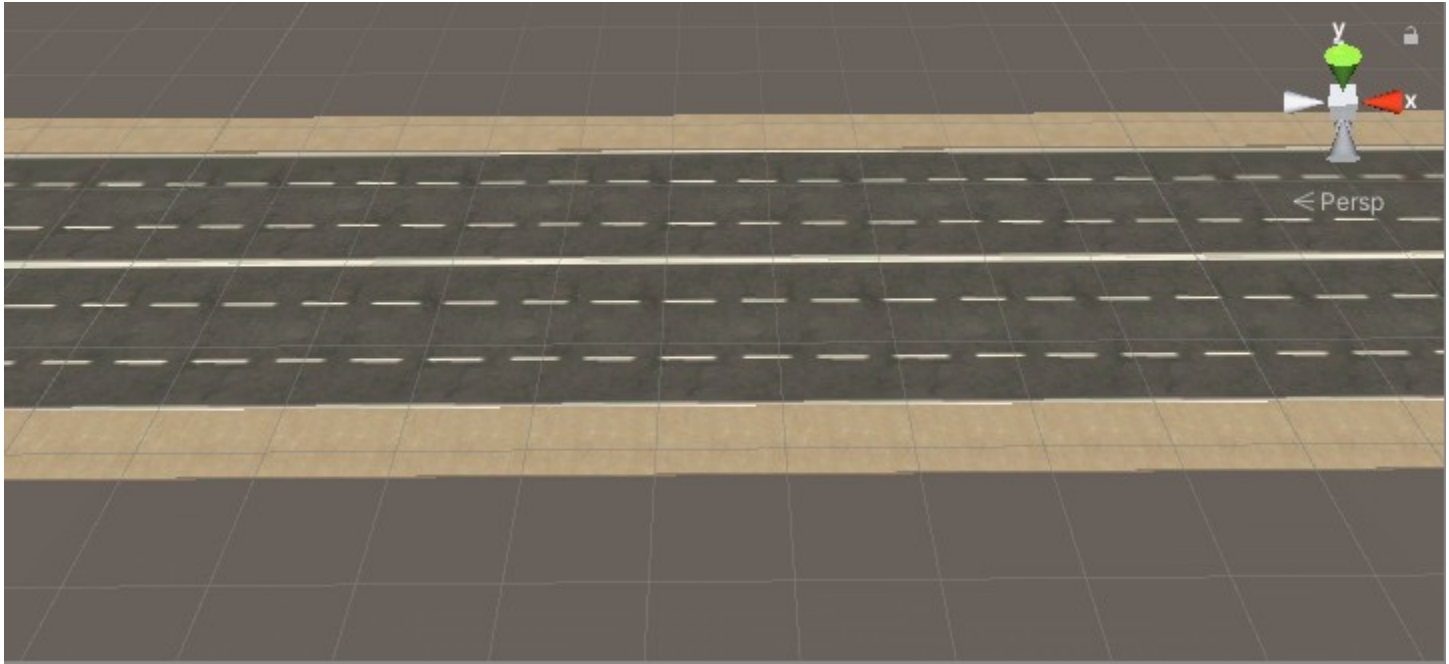
- next set values similar to the image above :)

SETTING UP THE PEDESTRIAN SYSTEM

settings up the pedestrain is almost process with setting up path follow for the ai vehicle, the both systems uses node based tracking system.

To setup accurately follow steps smoothly.

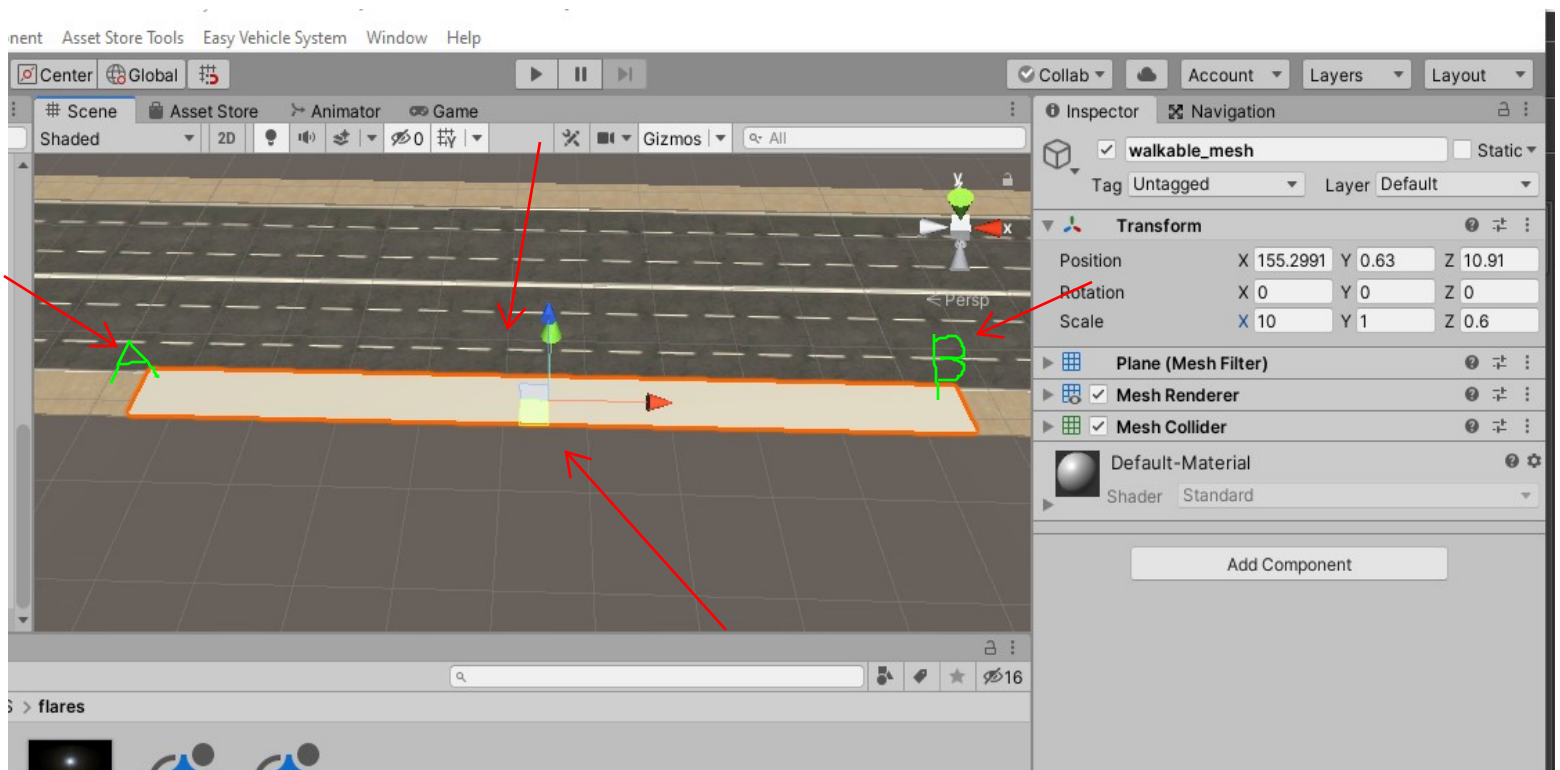
-lets say you got a road with pavement like the image below



-we gonna set a walk path on the side walk of the road.

-first we create a walkable mesh which a flat surface were the humans can move freely to do this we create a new plane gameobject and name it walkable_mesh.

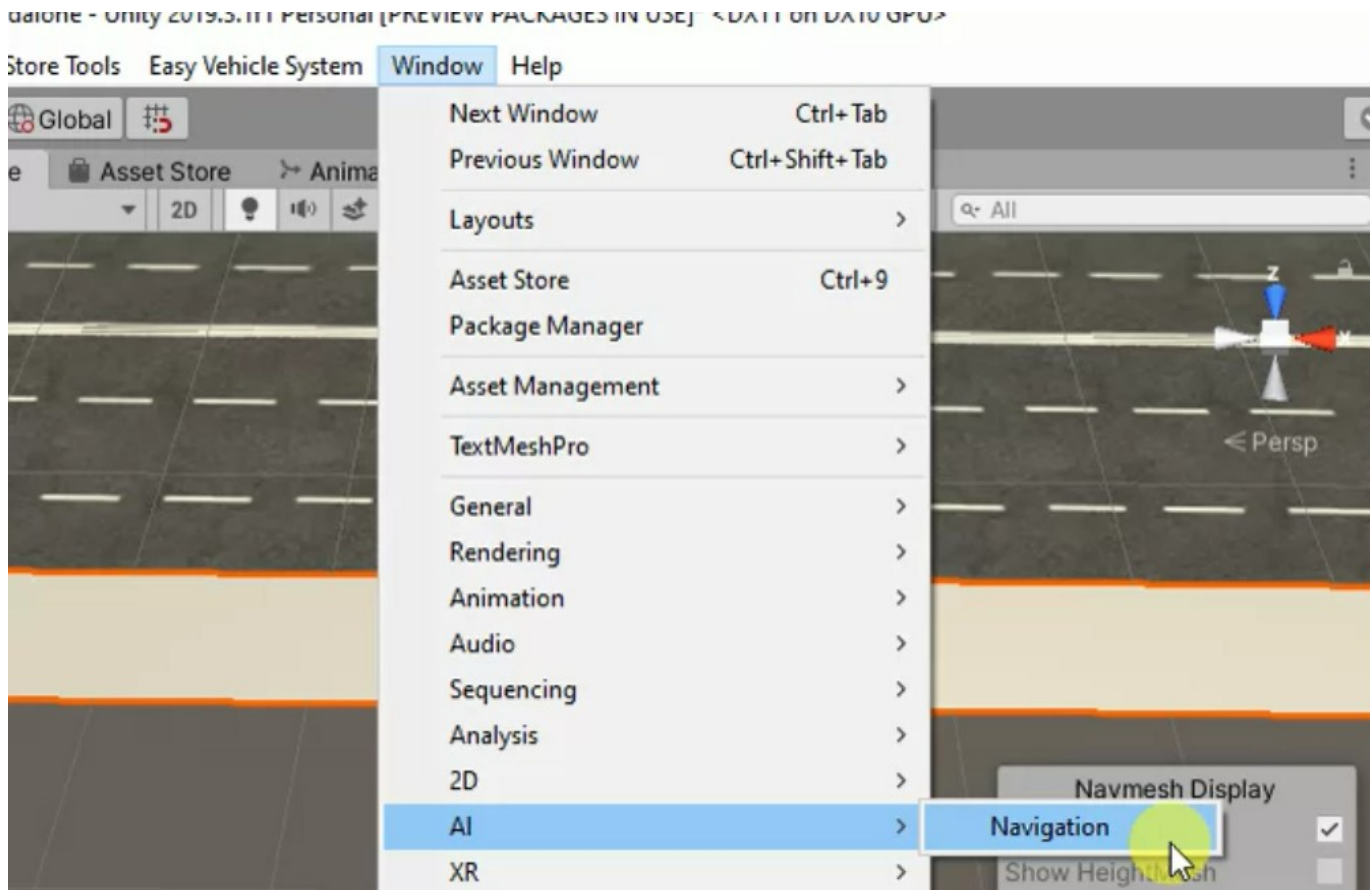
- resize it so fits same size with the road side walk and make a little above the ground level,like the image below



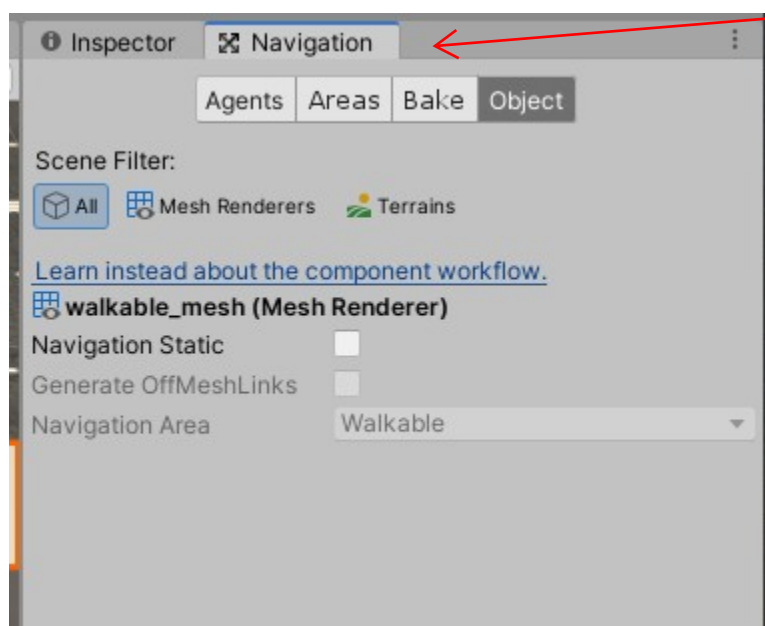
what the above image means is that we specified a simple walk area for humans . pedestrians will freely move from Point A to Point B.

-what we have is create a plane mesh we havent actually a walkable that UTC pedestrians will see. to make this mesh active we gonna use the "NavMesh system"

-next we need to call the navigation window to do that goto (window/AI/Navigation).



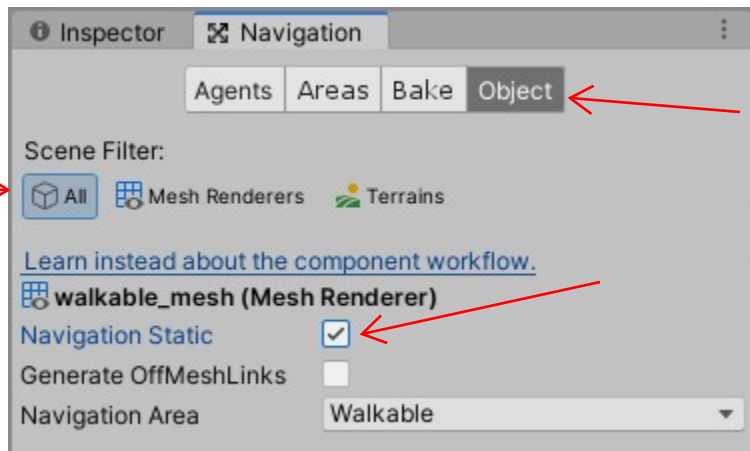
-next you should see this beside the inspector window ,like image below.



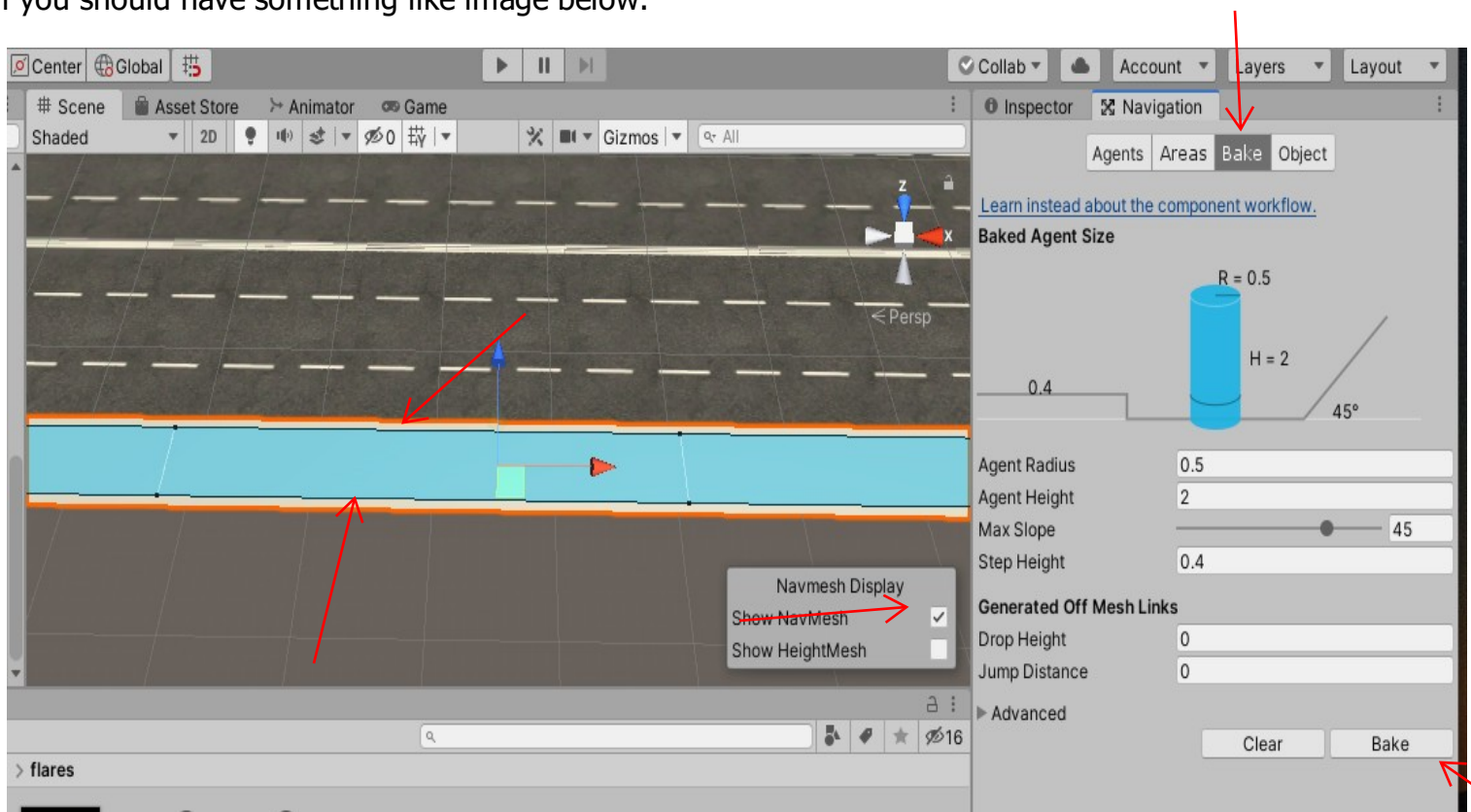
- next we prepare our plane mesh for baking, to this select your plane mesh i.e walkable_mesh, now move to navigation window

-next go to object tab and select "All" option and check the Navigation Static box to true.

-next make sure Navigation Area is set to walkable.

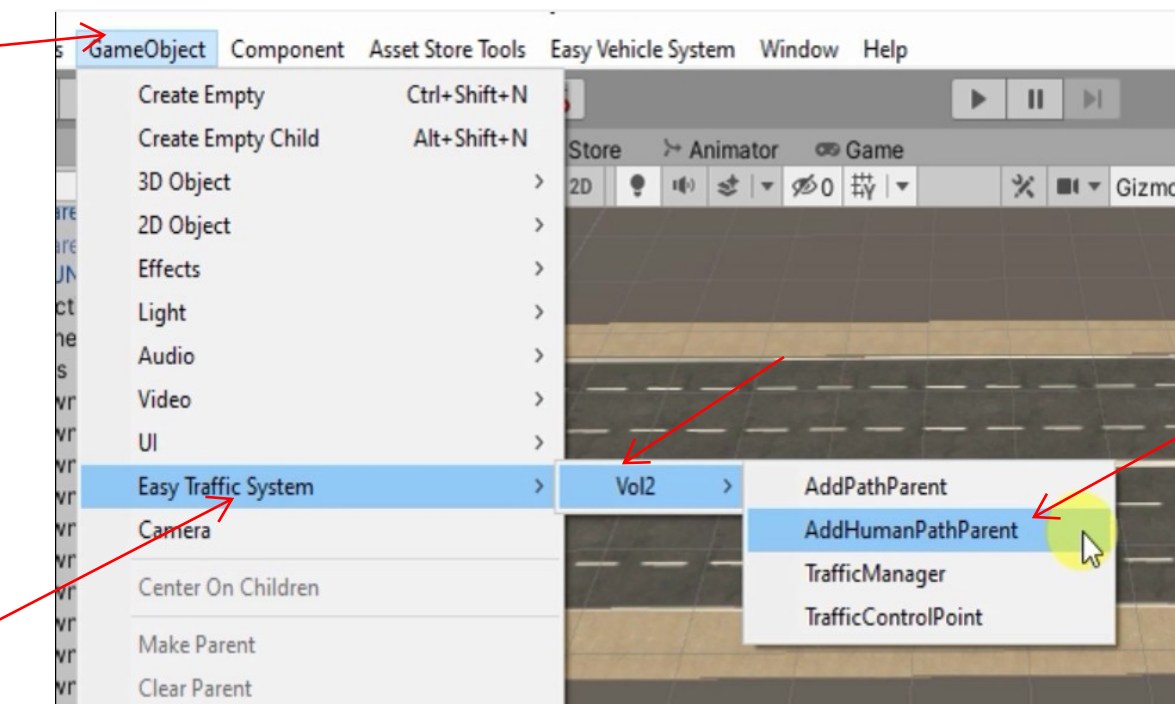


-next move to Bake Tab , keep all values at default unless you know what all these values are used you can tweak to get different AI behaviour. Once in bake tab with your plane mesh selected click on bake if all went well you should have something like image below.

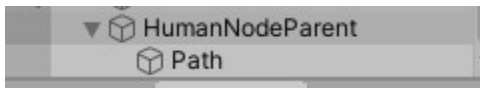


-now we got a complete walkable mesh.

-next we gonna specify waypoints using nodes to this goto (GameObject/EasyTrafficSystem/Vol2/AddHumanPathParent).



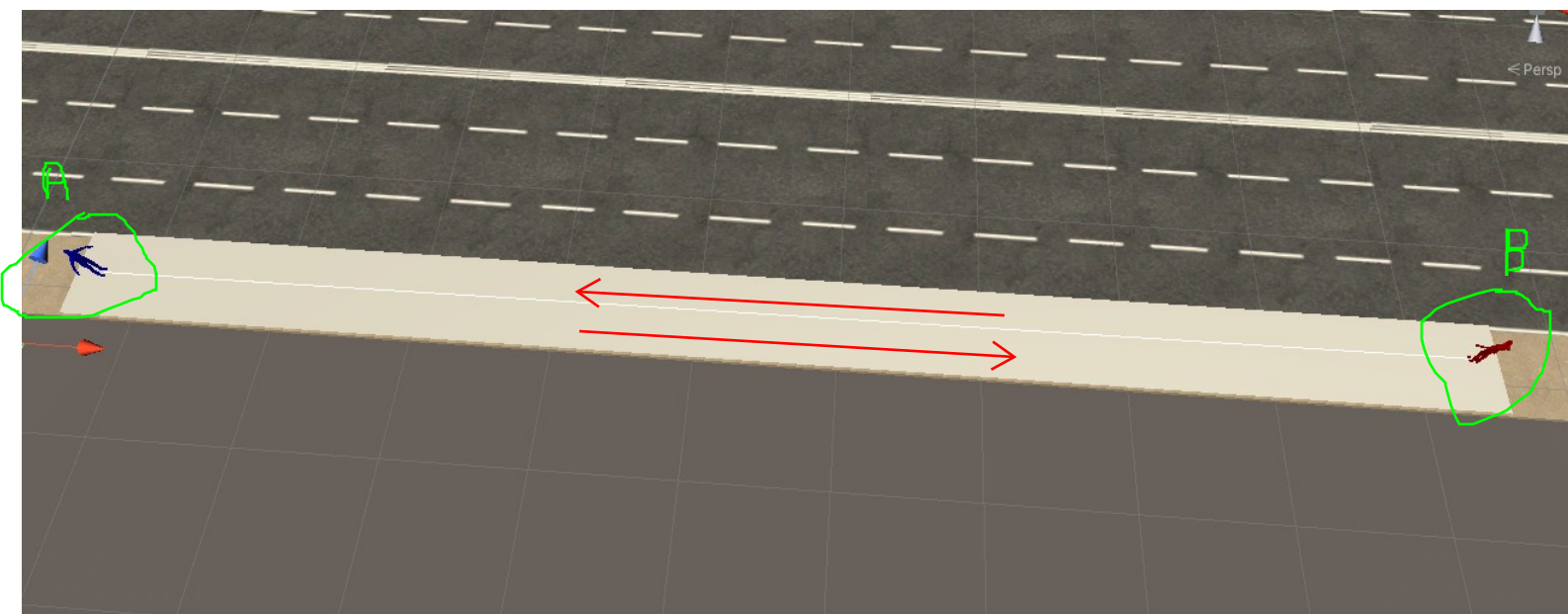
-next this should automatically add a HumanNodeParent gameobject to the hierarchy



-next select the child of HumanNodeParent which is the Path gameobject and hover the mouse icon to point A on the plane mesh in scene view.

-next click on Key "A" to add the first node which will also be gateway to spawn humans

- next with path selected move the mouse to point B on the plane mesh and click on Key "A" to add the second node which will be the last node , we two nodes only because the plane mesh is not long and not complex. it should the image below



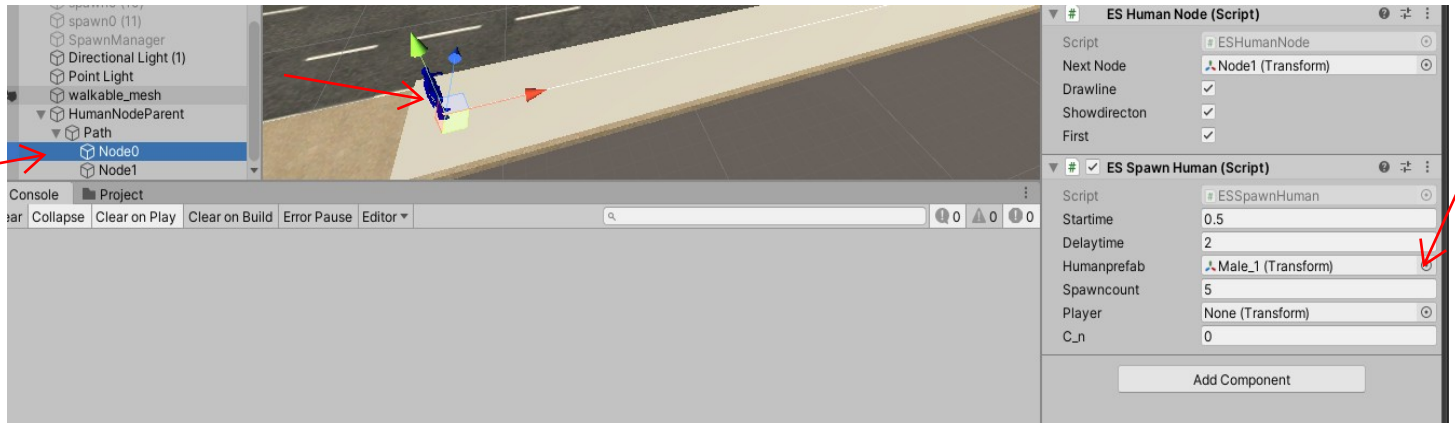
-now that we have configured the path follow sucessfully

-next we add a spawn point to spawn humans into the game

-select Node0 under the path gameobject , always make sure that the spawn point is attached to node, to make this more realistic you can place Node0 in places were player cant acess in the game world

-next add ESSpawnHuman component

-next drag and drop the Male_1 prefab from the prefab folder into the humanprefab slot in ESSpawnHuman component.



-next you increase the delaytime and also amount humans allowed in the game , drag and drop the player gameobject into the player transform slot if you want the spawnpoint to be disabled when the player is close,so the player dont have to see humans coming into the game cos its not realistic :).

-next test run

- :()

