Poly|Nav

PolyNav is a path finding system to use with Unity's 4.3 2D system. One of the best benefits of PolyNav2D is it's ease of use and setup. There are three prime components you should be aware of:

PolyNav2D is the main component. You should place *only one* in your scene preferably on a new empty game object.

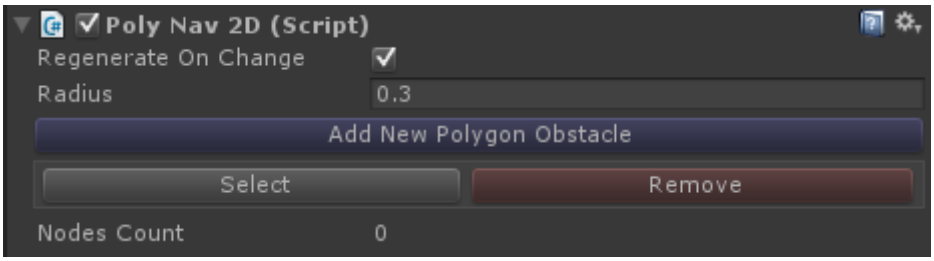
PolyNavAgent is the component that you should attach on any game object that you would like to navigate around the world

PolyNavObstacle is the component that most probably you aren't going to use directly as it will be created through PolyNav2D, although you can place that component on a game object to manually create a navigation obstacle.

Step 1

So the first thing you should do is create an empty game object and attach the PolyNav2D component or go to GameObject / Create Other / PolyNav2D. A PolygonCollider2D will automatically be attached as well. By using the default Unity controls for adjusting such a collider, you can define any shape for the navigation polygon.

(Ctrl + Click to remove a point. Shift + Click&Drag to move a point or to add a new if hovering an edge)



The "Regenerate On Change" toggle, will regenerate the map whenever any obstacle polygon change it's position, rotation or scale, but keep in mind that this option might be very costly on performance.

As a side note, the map is either way regenerated whenever a new Obstacle is added or removed in play time. You can also generate the map manually by calling GenerateMap() on the PolyNav2D like so: PolyNav2D.current.GenerateMap();

The Radius settings is the offset that agents will have from the edges. Typically, you set this radius equal to the radius that the agents will have.

To create an obstacle (or a hole) within that main navigation polygon, click the "Add New Polygon Obstacle" on the Inspector of the PolyNav, select that new obstacle and edit it just like you did the main polygon obstacle, since it also comes with a PolygonCollider2D. You don't have to be exact. You can have half an obstacle inside the walkable area.

The PolyNavObstacle has one option to invertPolygon. If it is added when the gameobject already has an atumated created PolyGonCollider because of a sprite existing, it will be set to true. If you think something goes wrong it will most probably be because the exiting polygon collider is not inverted. So try to set this to true.

Step 2

Select the game object that you want to be able to navigate within the map and attach the PolyNavAgent to it.

Max Speed is of course the maximum speed that the agent will move
Mass is the mass of the agent. The bigger the mass the more inertia it will gave
Stopping Distance is the distance from the goal that the agent will and consider reached
Slowing Distance is the distance from the goal that the agent will begin slowing down
Deceleration Rate is the rate at which the agent will slow down when when reaching the goal
Closer Point OnInvalid will make the agent go to the closer possible position if requested is invalid
Rotate Transform if checked will also rotate the agent
Rotate Speed is the speed that the agent will rotate if Rotate Transform is checked
Send Messages if checked will SendMessages to this game object regarding it's status *(more later)*
Debug Path if checked will draw the path
Avoid Radius is the agent avoidance radius with other agents. Leave it at 0 if you don't need avoidance

There are 2 demo scripts provided to move the agent about for a quick start. **ClickToMove** and **MoveBetween**

ClickToMove will simply move the agent at the clicked position while *MoveBetween* will move the agent between some way points at random selection.

Simply put either on the game object if you want a quick start.

Step 3

Regarding scripting, there are 2 important function on the PolyNavAgent:

bool SetDestination(Vector2 goal, Action<bool> callback = null)
Sets a new goal for the agent. You can optionally provide a callback function which will be called on either arrival with a *True* argument or if the destination is or becomes invalid with a *False* argument. The function itself return whether or not the destination is valid.

An alternative way instead of providing a callback function will be to catch the messages send by the agent if "Send Messages" is checked as described before. Here are the messages sent if 'Send Messages' is checked:

"OnDestinationReached"
"OnDestinationInvalid"
"OnNavigationStarted"
"OnNavigationPointReached" (this is when a path node is reached)

Please check the demo scripts ClickToMove and MoveBetween for example usage

Stop()
Will simply stop the agent from moving

There are also some important properties that you can use:

pathPending:bool	is any path pending right now?
hasPath:bool	does the agent has any active path
nextPoint:Vector2	the next point in the path that the agent is moving towards
remainingDistance:float	the remaining distance along the path for the goal

Well that's it! Remember that you can enable "Gizmos" in play mode on the top right of the "Game" tab, so that you are able to view the polygon map, obstacles and agent paths for debugging.

To use from any language other than C#, you can rename the PolyNav2D/Scripts folder to "Plugins" and move it under the root 'Assets' folder. Or you can SendMessage("SetDestination", goal:Vector2) to the PolyNavAgent and catch the send messages from the agent as described before.

Thanks and I hope you enjoy the simplicity of Poly|Nav

Gavalakis Vaggelis

nuverian@creative-minds.gr

Unity Forum Link