Documentation

1. **Waypoint.cs**

Waypoint must exist within a loop to operate waypoint is the basic component of the system it connects to other waypoint through loop or branches

**Variables:**

private int \_id :unique id based on the position"

public WaypointLoop parent :the loop that contains the waypoint"

string json: stores the data of the waypoint in a json format"

public Waypoint Next: the next waypoint in the path, null if the last waypoint"

public Waypoint previous: the previous waypoint in the path, null if the first waypoint"

public Transform HandleA: the handle that control the previous link\nused in creating bezier bath"

public Transform HandleB: the handle that control the next link\nused in creating bezier bath"

public bool LockHandles : True: handles work together as one line result in continous path at the point\nfalse: handles work separately result in a break in the continouity of the path"

private bool DrawLink: controls if the link to the next waypoint is visible or not"

private float localYoffset: temporary saved data for 2D loop"

private float HandleAoffset: temporary saved data for 2D loop"

private float HandleBoffset: temporary saved data for 2D loop"

public List<Bezier.PathPoint> inBetweenPoints: gets the intermediate path points between the the point and the next waypoint"

public Bezier.PathPoint[][] InBetweenBranches: gets the intermediate path points between the the point and the Branches"

public List<Waypoint> Branches: list of waypoints to connect to"

public List<Waypoint> ReverseBranch: list of waypoints that is connected to this point"

public bool enterance, exit: specify if the waypoint is an entrance of a branch of an exit

public Vector3 normalDir: vector directed to the normal to the waypoint, normat to the direction to the nect point"

public float distanceBetweenPoints = 1f :distane between intermediate path points between the point and the next point"

public event EventHandler onStateChanged: event is run when point change position or change direction or handles

this event is run whenever

public event EventHandler onDeleted: event is run when point is deleted"

public bool drawinbetween = true: draw in between points and normal

True: draw gizmos

False: Don’t Draw

public int GizmoMode: change gizmo mode

0: no Gizmo

1: only points

2: only lines

3: with normal

**Methods:**

getInbetween:

public List<Bezier.PathPoint> getInbetween(Waypoint Next)

return the inbetween point using the next waypoint

<param name="Next"></param>

<returns>list of bezier points, null if the Next is not connected to the current point</returns>

The inbetween points are basically the intermediate points when moving from point A and point B

public List<Bezier.PathPoint> getInbetween(int id)

return the inbetween points using index of the waypoint

<param name="id">the index of the branch, -1 if you want next</param>

<returns>list of bezier points, null if the Next is not connected to the current point</returns>

public static List<Bezier.PathPoint> getInbetween(Waypoint A,Waypoint B)

return the bezier path points between any two waypoints regardless the system

<param name="A">start point</param>

<param name="B">end point</param>

<returns>return list of points</returns>

OnStateChanged:

public void OnStateChanged()

recalculate point parameters if state changed.

you can call it whenever you want

it recalculate the normal and the inbetween points and the Branches.

public void OnDeleted()

clean up after deleting waypoint