Realistic Traffic Controller

Thank you for purchasing and using RTC! This documentation will guide you on how to use API to create new lanes, create new waypoints, and edit them at runtime or inside the editor.

RTC.cs

You can use RTC class to do create new lanes and waypoints at runtime or inside the editor. All public static methods have been explained below;

```
    /// <summary>
    /// Creates a scene manager and returns as a gameobject.
    /// </summary>
    public static GameObject CreateSceneManager() {}
```

```
    /// <summary>
    /// Creates a new empty lane and returns as a RTC_Lane.
    /// </summary>
    public static RTC_Lane CreateNewLane() {}
```

```
    /// <summary>
    /// Creates a new lane with waypoints (with vectors) and returns as a RTC_Lane.
    /// </summary>
    /// <param name="waypointPositions"></param>
    /// <param name="radius"></param>
    /// <param name="speed"></param>
    /// <returns></returns>
    public static RTC_Lane CreateNewLaneWithWaypoints(Vector3[] waypointPositions, float
radius, float speed) {}
```

```
    /// <summary>
    /// Creates a new waypoint with given lane, position, radius, and target speed.
    /// </summary>
    /// <param name="lane"></param>
    /// <param name="position"></param>
    /// <param name="createWithRadius"></param>
    /// <param name="createWithSpeed"></param>
    public static void CreateNewWaypoint(RTC_Lane lane, Vector3 position, float
createWithRadius, float createWithSpeed) {}
```

```
    /// <summary>
    /// Creates a previous waypoint with given lane, position, radius, and target speed.
    /// </summary>
    /// <param name="lane"></param>
    /// <param name="position"></param>
    /// <param name="createWithRadius"></param>
    /// <param name="createWithSpeed"></param>
    public static void CreatePreviousWaypoint(RTC_Lane lane, Vector3 position, float
createWithRadius, float createWithSpeed) {}


    /// <summary>
    /// Creates a next waypoint with given lane, position, radius, and target speed.
    /// </summary>
    /// <param name="lane"></param>
    /// <param name="position"></param>
    /// <param name="createWithRadius"></param>
    /// <param name="createWithSpeed"></param>
    public static void CreateNextWaypoint(RTC_Lane lane, Vector3 position, float
createWithRadius, float createWithSpeed) {}


    /// <summary>
    /// Reorders all waypoints of the lane.
    /// </summary>
    /// <param name="lane"></param>
    public static void ReOrderWaypoints(RTC_Lane lane) {}


    /// <summary>
    /// Closes the circuit by connecting first and last waypoints together.
    /// </summary>
    /// <param name="lane"></param>
    public static void CloseCircuit(RTC_Lane lane) {}


    /// <summary>
    /// Breaks the circuit by disconnecting first and last waypoints from each other.
    /// </summary>
    /// <param name="lane"></param>
    public static void BreakCircuit(RTC_Lane lane) {}


    /// <summary>
    /// Inverse direction.
    /// </summary>
    /// <param name="lane"></param>
    public static void ReverseCircuit(RTC_Lane lane) {}
```

```
    /// <summary>
    /// Aligns all waypoints by X axis.
    /// </summary>
    /// <param name="lane"></param>
    public static void AlignX(RTC_Lane lane) {}


    /// <summary>
    /// Aligns all waypoints by Y axis.
    /// </summary>
    /// <param name="lane"></param>
    public static void AlignY(RTC_Lane lane) {}


    /// <summary>
    /// Aligns all waypoints by Z axis.
    /// </summary>
    /// <param name="lane"></param>
    public static void AlignZ(RTC_Lane lane) {}


    /// <summary>
    /// Inserts a new waypoint at given index.
    /// </summary>
    /// <param name="lane"></param>
    /// <param name="waypoint"></param>
    /// <param name="index"></param>
    public static void InsertWaypoint(RTC_Lane lane, RTC_Waypoint waypoint, int index) {}


    /// <summary>
    /// Finds nearest point on the line and returns as vector 3.
    /// </summary>
    /// <param name="linePnt"></param>
    /// <param name="lineDir"></param>
    /// <param name="pnt"></param>
    /// <returns></returns>
    public static Vector3 NearestPointOnLine(Vector3 linePnt, Vector3 lineDir, Vector3
pnt) {}


    /// <summary>
    /// Gets the container for lanes or creates it.
    /// </summary>
    /// <returns></returns>
    public static Transform LaneContainer() {}


    /// <summary>
    /// Updates all lanes and waypoints.
    /// </summary>
    public static void UpdateEverything() {}
```

Support

You can contact me via email. I'll be responding within 24 hours. Please include your invoice number while sending me an email.

BoneCrackerGames@gmail.com