TRACKGEN

1. Introduction

TrackGen can procedurally generate pseudo-endless tracks, commonly used in *Endless Runner* games. Each track is generated by combining a series of blocks fitted to each other, where each block suffers a deformation in its front face, so that the next block to be generated fits its back face on the front face of the previous block, and thus repeating the process indefinitely to build the track. To maintain consistency over time, the asset uses a coherent noise generation module based on *PerlinNoise* algorithm for each rotation axis, in which you can control several parameters to achieve the desired result.

These are some of the main features of the asset:

- Generates pseudo-endless tracks
- Tracks can be generated in runtime
- Deformation control via *Perlin Noise* algorithm

TrackGen currently only supports PC's running 64-bit version of Windows.

2. Demo scenes

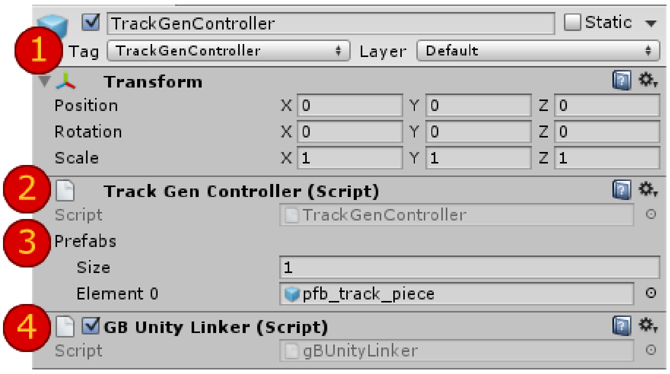
The asset has three sample scenes that can be found at:

- *Assets\TrackGen \DemoScenes\PreGeneratedTrack*
- *Assets\TrackGen \DemoScenes\RealTimeTrack*
- *Assets\TrackGen\DemoScenes\CarDemo*

PreGeneratedTrack shows an example of how to generate an entire track at once before starting the game. *RealTimeTrack* shows an example of how to generate a track in runtime. *CarDemo* shows a simple example application based on the runtime track generation example.

3. Configuration interface

The object *TrackGenController* is responsible for linking Unity to the procedural generation system of *TrackGen*, and its structure can be seen Figure 1.



☑ TrackGenController □ Static

Tag TrackGenController Layer Default

Transform

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

Track Gen Controller (Script)

Script TrackGenController

Prefabs

Size 1

Element 0 pfb_track_piece

☑ GB Unity Linker (Script)

Script gBUnityLinker

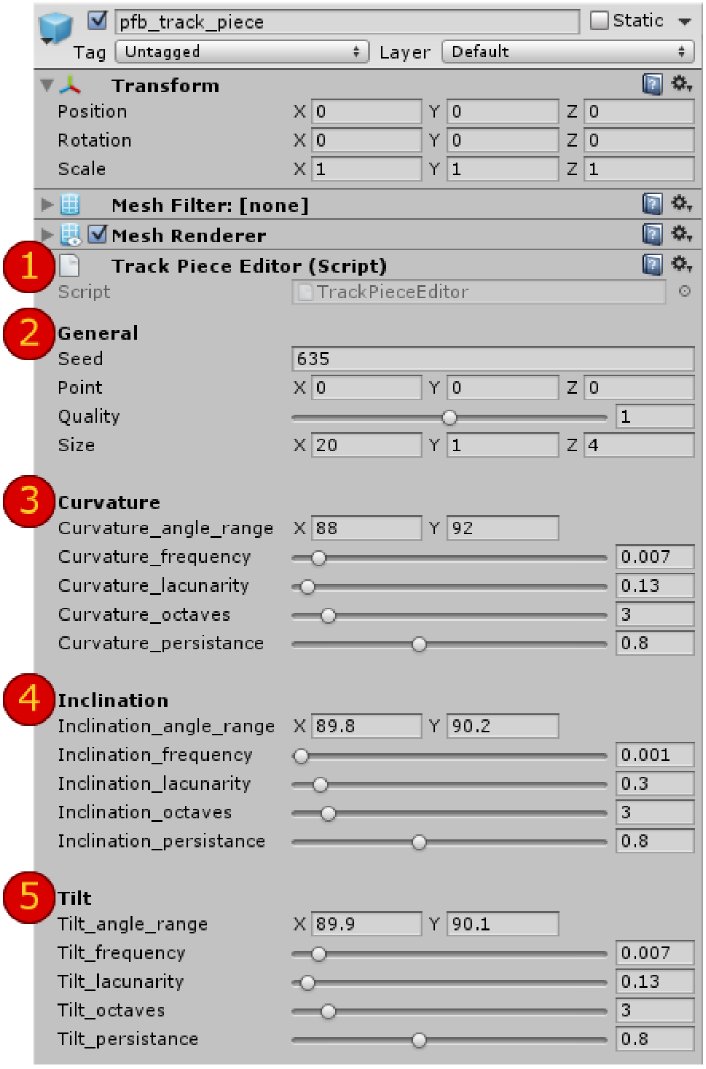
Figure 1. *TrackGenController* interface.

1) Tag *TrackGenController*, should not be changed since it is used internally to locate the object.
2) Script *TrackGenController* contains the prefabs of the objects that compose the generated track.
3) Prefabs corresponding to objects used in the track generation. You can customize them by changing their materials or adding other components to them, however you must keep their names unchanged and should not remove the other existing components.
4) Script *gBUnityLinker* is used to connect Unity to *TrackGen* internal system, so should not be removed.

The block configuration interface can be found in *pfb_track_piece* prefab, as shown in Figure 2. The angles of *Curvature*, *Inclination* and *Tilt* control the range of the deformation. The other parameters are related to the control of coherent noise generation, as follows:

- *Frequency*: Controls the frequency of the noise. The higher the frequency, the greater the variation in the angle of the track.
- *Lacunarity*: Defines the frequency multiplier between successive octaves.
- *Octaves*: Controls the level of detail of the noise generated by the algorithm.
- *Persistance*: Defines the roughness of noise.

It is difficult to describe the impact of each parameter in the generation since many of them are intrinsically connected with each other, so it is easier to change them and observe if the result is in accordance with the desired output.



☑ pfb_track_piece ☐ 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

Mesh Filter: [none]

☑ **Mesh Renderer**

Track Piece Editor (Script)

Script TrackPieceEditor

General

Seed	635					
Point	X	0	Y	0	Z	0
Quality	1					
Size	X	20	Y	1	Z	4

Curvature

Curvature_angle_range	X	88	Y	92
Curvature_frequency	0.007			
Curvature_lacunarity	0.13			
Curvature_octaves	3			
Curvature_persistance	0.8			

Inclination

Inclination_angle_range	X	89.8	Y	90.2
Inclination_frequency	0.001			
Inclination_lacunarity	0.3			
Inclination_octaves	3			
Inclination_persistance	0.8			

Tilt

Tilt_angle_range	X	89.9	Y	90.1
Tilt_frequency	0.007			
Tilt_lacunarity	0.13			
Tilt_octaves	3			
Tilt_persistance	0.8			

Figure 2. Track block configuration interface

1) The *TrackPieceEditor* script allows you to configure the generation parameters of each block.
2) General settings. The parameter *seed* usually is defined by code, as can be seen in the example scenes, as well as the parameter *point*. The parameter *quality* defines the quality of the noise generated by the coherent noise generation system. The parameter *size* defines the dimensions of the blocks.
3) *Curvature* controls the rotation of the blocks in the y-axis, creating curves in the track.
4) *Inclination* controls the rotation of the blocks in the x axis to provide ascents and descents.
5) *Tilt* controls the rotation of the block in the z-axis, generating tilts in the track.

4. What is in the package?

Below you can check out a brief description of the files contained in the package.

- *Assets\Editor* and *Assets\StandardAssets*
    - Only the car scripts and prefab provided by Unity, used in the example scene.
- *Assets\Plugins*
    - It contains the plugins of pseudo-random numbers generation and coherent noise generation used for the generation of the track elements.
- *Assets\TrackGen*
    - *\DemoScenes*
        - The demo scenes.
    - *\Dll*
        - Encapsulates the procedural generation system logic. Where the magic happens.
    - *\Resources*
        - *\Camera*
            - Contains only a camera used in the examples.
        - *\Controller*
            - Contains the *TrackGenController*, responsible for managing the access and settings of the generation system.
        - *\Materials*
            - Here are the default materials used in the track blocks.
        - *\Prefabs*
            - Prefabs of the track elements.
        - *\Textures*
            - Contains some example textures used in the blocks.

5. Warnings

Only a few warnings which you should be aware.

- You may need to add the tag *TrackGenController* manually, otherwise when you run your game, after building it, the object *TrackGenController* will not be found and the game will not work properly. The same behavior does not occur in the Editor.

6. Contact info

If you have any other questions, you can contact me via email *mav.jed@gmail.com*.