- PathiGen -

-Infinite Path Generator-

Created by RipCord Development
info@ripcorddev.com

PathiGen is a system that lets you quickly and easily generate a path of either defined or infinite length. It uses a script combined with an array of prefabs to generate the path.

MODELS

The PathiGen package includes a number of model sets that can be used to create many different styles of paths. You can use these pieces to quickly get started with PathiGen or you can swap them out for your own pieces.

- **Prefab Names** - The prefabs that get loaded into PathiGen do not need to be named in any specific way. The ones included in this package are named that way for easy identification.
- **Prefab Parts** - Each path prefab is made up of multiple parts. The set up of some of these parts is very important, others do not matter. This will be explained using one of the halfPipe pieces as an example.
  - **halfPipeStraight** - An empty game object that is used as a parent for the rest of the path piece.
    - *This object can be named whatever you want.*
  - **COLLIDER roadStraight** - A mesh collider that has been custom-made to fit the road mesh. (Optional)
    - *The prefabs included in pathiGen all include mesh colliders. These are not needed for pathiGen to function and can be replaced with regular Unity colliders if performance becomes an issue.*
  - **MESH roadStraight** - The actual road mesh.
  - **POINT roadEnd** - An empty game object. This transform position and rotation of this object will be used to place the next path piece. The pivot point of the next path piece generated will be oriented to this point on the previous path piece. Placement is important when setting up your own path pieces.
    - *This object must be named "POINT roadEnd" in order for the pathiGen script to detect it.*

SCRIPTS

PathiGen consists of one main script. This script must be placed on an object in your scene (recommend an empty game object). This script will contain all the path pieces as well as the settings that control their generation and placement.

pathiGen Settings

- **Path Style** - This option lets you choose how the path pieces will be generated. You can select one of 3 options:
  - **useArrayOrder** - This option will generate each path piece in the order it appears in the array. When the last element in the array has been generated, pathiGen will cycle back to the beginning of the array and continue until the path piece limit has been reached.
  - **usePathPieceFrequency** - This option will randomly select one element from the array each time a path piece is generated. Each path piece can be given a frequency value to determine if the piece is more or less likely to be picked. A frequency value of 0, means the piece will never be selected.
  - **useRandomFromArray** - This option will randomly select one element from the array each time a path piece is generated. There is no control over which piece will be selected next.
- **Path Pieces** - This array holds all of the prefabs for path pieces that can be generated.
- **Path Piece Frequency** - This array lists all of the frequency values for each path piece in the pathPieces array. Each value lines up to the same element number in the previous array. (ex. pathPieces element 2 = pathPieceFrequency element 2). The size of this array must be equal to the size of the pathPiece array for predictable results.

- **Path Piece Limit** - This value controls the limit of how many pieces can be generated before the script stops generating new pieces. A value of 0 will all the script to generate pieces infinitely.
    - **Tip***: In order to keep things running smoothly, you will probably want to limit the number of pieces generated. A good way to control this while still having a path generate infinitely is to introduce a counter that increments or decrements each time a piece is created or destroyed. This will limit the number of pieces on screen at any given time, but will also all the script to infinitely generate pieces.*
- **New Path Time** - This value controls the delay between pieces being generated. A value of 0 means a piece is being generated every frame.