**Due July 5, 2020**

Write the following function:

fn generatePath pathArray

This function accepts an array of strings where each string represents a different path piece. Think of this like a toy train set where there are straight and curve pieces that can be attached to each other in different configurations. This function will go through the pathArray, create, place and orient each piece such that the pieces are properly connected to each other.

The strings stored in the path array indicate what piece to generate next:

* "left": left turn (45 degrees)
* "right": right turn (45 degrees)
* "straight": straight path piece

This path will be placed in your final scene as a path between the putting greens

For each path piece you must generate the path (can be a single thing box as the basis of each piece)

The length of your straight piece should allow you to form a figure "8" path by using 6 left, 2 straight, 6 right, 2 straight.

**Example:**

suppose we have:

thePath = #("left","left","left","left","left", "left", "left", "left")

generatePath thePath

The above would generate a circular path composed of 8 "left" path pieces.

thePath = #("left","left","left","left","left", "left", "straight", "straight", "right", "right", "right", "right", "right", "right", "straight", "straight")

generatePath thePath

The above would generate a figure 8 path shape

**Basic Path System (GAM536 - 10 marks, DPS936- 8 marks):**

* This is the minimum requirement for completion of the assignment
* The script generates all relevant pieces (left, right, straight) correctly, ensuring the following:
  + The pieces link up seamlessly
  + The array can be any size, with any number of pieces in any order, and your program must still work

**Optional components (GAM536 - up to 10 marks, DPS936- up to 12 marks):**

While the following components are not mandatory, you will need to do some of the following components to get a better mark. Note that you do not need to implement **all** of the options below. As some components are easier, you can choose what to implement to cover the remainder of your assignment mark.

For these components unless otherwise stated, all meshes must be programmatically generated.

**Decorations - 2 marks**

Along the side of the path, place a decorative element such as a lamp post or a bench. The decoration must be placed along the path. The exact nature of the decorative element is entirely up to you, but it needs to be generated in code. It does not need to be a complicated object. patches of grass however, are not allowed.

**UI - 2 marks**

Provide a UI. The UI must support the generation of all the extra elements. For example, if you have decorations, your UI should have an option of the density of the decoration or whether to generate decorations at all. It must allow you to specify the path pieces. This can be done as simply as a single string, that your program parses before calling the generatePath function

**Edge - 2 marks**

Create a raised edge for the path pieces. The edge must exist on both sides. It must fit the path snuggly (ie you can't just use one straight box to edge the curve pieces, it must follow the curve snuggly and bend in the same way)

**Board Walk - 4 marks**

Instead of using a plane or single thin box to form a path piece, create a boardwalk pathway. That is the floor of the path is made up of boards of wood witha small gap between them (represented as small boxes of the same size). the individual boxes themselves should not be stretched in any way. Thus the in the curve sections, the gap will look a bit bigger on outer edge than the inner edge.

**Bridge - 4 marks**

Add a **bridge** piece type to the path system. Alter the generatePath function so that when the string "bridge" is encountered, a bridge piece will be created. The bridge piece should create a tunnel in its midsection, allowing another a straight path pieces to be placed through the tunnel. This implies that the bridge will start level with the rest of the path, rise to however high it needs to go in the midsection, then return to level by the end of the piece. Take this into account when deciding/experimenting with piece proportions. Furthermore the bridge should look supported. There is some artistic license you can take with how to do support as long as the bridge is supported.

**Submission**

Place your code into a single file called a2.ms

A simple tester main will be provided shortly for you to test your path generation function

**Lates**

10% for first week 10% per day there after