Operation Guide
'Press 1' to load `Bin/Json/ShapesSceneLayout1.json`
'Press 2' to load `Bin/Json/ShapesSceneLayout1.json`
'Press ESC' to quit the program

ShapesSceneLayout
The shapes that are created within the scene are located in the ` ShapesSceneLayout ` files within the `Bin/Json/` folder.

There are two predefined layout files which can be loaded and reloaded during runtime using the defined keyboard keys 1 & 2.

Json
I used the JSON format to store the shape's information within the scene layout files. Json is a low weight, easy to read format that is widely supported.

SceneLayoutFile -> Shape Entry
Shape Entry Required Keys
Model

- Alias to a model that is defined in the `Bin/Json/Models.json` file
- **NOTE:** Do to define this to a model file path

MovePattern

- PATTERN_UP_DOWN = 1
- PATTERN_LEFT_RIGHT = 2
- PATTERN_CIRCULAR = 3
- PATTERN_BOX = 4

Dimension

- Sets the scale of the mesh

MoveSpeed

- Movement speed of the mesh

Colour

- Diffuse colour of the shape's mesh shader

StartPosition

- Position in which the shape does its pattern movement around

CShape
All shapes are created using a single class `CShape`. The data for the shapes are loaded through the scene layout files.

When loading a different scene layout, previous shapes are recycled and any leftover shapes are deleted.

CPathMover
This class handles the movement of the shapes within the scene. It holds a strong reference to the owner shape. The class moves the shape by changing the position within the shape's world matrix.

Models
I render the shapes by importing models. I had not implemented debug drawing functionality into my existing framework and therefore I used what was available.