**SOC09109 Group Project**

“Real-Time 3D Sound/Music Visualisation”  
**Client:** Gregory Leplatre

USER DOCUMENTATION

**Project team:**

Luis Del Pozo Tenor

Grzegorz Laskarzewski

Emmanuel Miras

Conner Weatherston

Jack Yorston

Ximo Zeng

Table of contents

1. **Introduction**
2. **Software requirements**
3. **Setting up the project**
4. **Available visualisations**
5. **Using the node editor**
6. **Inputting sound**
7. **Closing the project**

**1. Introduction**

This project is designed to manipulate the physical properties of a three-dimensional mesh in real-time using auditory input. This is achieved through a combination of two pieces of software: **Unity** – a popular game engine – and **MaxMSP** – a visual programming language for music and multimedia.

*<side-by-side screenshots of Unity and MaxMSP in use>*

What follows is a simplified explanation of how the project works:

* Auditory input is fed into MaxMSP. This input can be a pre-recorded piece of audio, or real-time audio captured by a microphone connected to the system.
* MaxMSP will detect several of the audio’s properties, such as pitch and amplitude. It then converts these properties into representative values, which in turn will be passed to Unity.
* Unity processes those values via a node-based interface. Values representing the 3D mesh’s visual physical properties are determined by the values representing the auditory properties. The ensuing constant changes to the physical properties are reflected in real-time by the 3D mesh.

We will now go into a deeper explanation of how to set up and operate the project.

**2. Software requirements**

Due to the nature of the project, it has not been built as a standalone executable. It requires both pieces of software used to develop it to also run it.

**Unity -** *unity3d.com/get-unity/download*The latest version of Unity was used during development**.** At time of writing, this is **v5.5.2f1  
  
MaxMSP** - *cycling74.com/downloads  
<ask Luis for version of Max used during development>*

**3. Setting up the project**

**4. Available visualisations**

There are 3 built-in visualisations available for use:

*<screenshot of each mesh>*

*<describe how to substitute meshes>*

**5. Using the node editor**

The node editor is where the connections between the audio and visual properties of the project are defined. There are four types of node:

* + **Audio** – outputs a value based on a property of the input audio (e.g. pitch, amplitude). Takes the output from MaxMSP as input.
  + **Visual** – alters a property of the three-dimensional mesh rendered in Unity (e.g. scale, rotation). Takes the output from an AudioNode as input.
  + **Operator** – a function that alters the value passed into it by adding to/subtracting from it or multiplying/dividing it. Takes the output from an AudioNode as input.
  + **Controller** - <*explanation of Controller node>*

To begin setting up Visual nodes, a Controller node must first be created.

To create a connection between nodes, click the ‘Attach’ button on the node that will provide the input (i.e. Audio or Operator), then click the ‘Attach’ button on the node that will receive the input (i.e. Operator or Visual). A line joining the two nodes together will appear, indicating the connection you just made.

Audio nodes cannot have an input connection.

Deleting a node will delete all connections to it. Right-click the node and press ‘Delete’ on the context menu.

**6. Inputting audio**

**7. Closing the project**

To close the project, simply click the same ‘Run’ button in Unity that was used to start the project.