Dynamic Audio for Digital Media
Project Brief

You are asked by a client to develop a desktop-based music/sound-focused, application that fulfills the following **five** specifications:

1) The user should interact with the application using three discrete inputs.

Examples: Three keys from the computer keyboard, Three on-screen buttons accessed through the mouse, Two sliders and one toggle accessed through the mouse.

2) The application should make use of samples stored within Arrays.

Examples: Brief snippets of verbal narrative, SFX such as footsteps, explosions or UI sounds are played back from arrays sometime during the gameplay experience.

3) The application should make use of at least one sound synthesis technique for the generation of music.

Examples: FM, AM, Additive, Subtractive synthesis used for the generation of dynamic music responding to the user's actions.

4) The application should make use of at least one signal processing technique

Examples: Delays, comb filtering, filtering. These could be used to process either the samples or the synthesis.

5) There must be some sort of gameplay experience or narrative. This doesn't necessarily mean that the application should be a game in the traditional sense, but the user should be able to navigate through different "states" of the app based on the interaction with the application.

Application Examples:

1) A platform game where music, sound and movement are all interconnected (such as Braid)

2) An audio engine where you have to move between different presets based on visual cues (such as the Inception App)

3) A voice-based story where you learn more about it by clicking buttons and interacting through an interface. (such as Papa Sangre)

4) A music-based game where you follow musical cues to perform a piece of music (such as Steve Reich's Clapping Music App)

Clarification:

You should be using **Unity** for the interface, **Pure Data Vanilla** for the audio engine, and the **libPd4Unity integration package from Blackboard**.

Make sure that you can also test all the audio elements of the application using **a dummy interface comprising of sliders, buttons and toggles within Pure Data**. This would replicate actions happening within the game, without having to use the Unity interface.

For any questions or further clarifications:

c.michalakos@abertay.ac.uk