Cardiff School of Computer Science and Informatics

Coursework Assessment Pro-forma

Module Code: **CM3106**
Module Title: **Multimedia**
Lecturer: **Prof. David Marshall**
Assessment Title: **MATLAB Interactive Granular Convolution-based Synthesiser**
Assessment Number: 1
Date Set: **Monday 18th October 2021**
Submission Date and Time: **Friday December 2, 2021** at 9:30am
Return Date: Within 3 weeks of submission date

This assignment is worth **50 %** of the total marks available for this module. If coursework is submitted late (and where there are no extenuating circumstances):

1. If the assessment is submitted no later than 24 hours after the deadline, the mark for the assessment will be capped at the minimum pass mark;
2. If the assessment is submitted more than 24 hours after the deadline, a mark of 0 will be given for the assessment.

Your submission **must** include the official Coursework Submission Cover sheet, which can be found here:

https://docs.cs.cf.ac.uk/downloads/coursework/Coversheet.pdf

Submission Instructions

You should submit the following files.

Description		Type	Name
Cover sheet	**Compulsory**	One PDF (.pdf) file	[student number].pdf
Q1	**Compulsory**	One PDF (.pdf) or Word file (.doc or .docx)	Q1_[student number].pdf
	Compulsory	One or more program source file and supporting data	No restriction for individual files, but please upload all files zipped into a single zip file.

Any code submitted will be run **on a Macintosh PC (MATLAB 2020a)** and must be submitted as stipulated in the instructions above. Your solution should therefore be compliant with **MATLAB 2020a (running on any platform).**

Staff reserve the right to invite students to a meeting to discuss coursework submissions

Assignment

The exercise involves developing a **MATLAB Interactive Granular Convolution-based Synthesiser**.

You must submit a typeset report in PDF format — a short (no more that 3,000 words) written description conveying all the appropriate information to demonstrate and explain your programming philosophy is all that is required. You should also submit all MATLAB code as text files or Live Scripts and **include all supporting files** (GUI config files, media, etc.) contained in a single **zip** file for submission. For complete details see the following sheets.

Learning Outcomes Assessed

1. Demonstrate an awareness of the factors involved in multimedia data processing and analysis through detailed study of potential problems of processing multimedia data and of implications of multimedia formats and types.

2. Show familiarity, including practical investigation, with a range of multimedia formats for text, graphics, animation, audio, images and video by detailed study of common multimedia formats.

3. Possess an understanding of the underlying concepts, representation and processing of multimedia data with relevance to the applications in digital audio, imagery and video, including digital audio processing and synthesis, audio/image/video compression and multimedia data retrieval.

Criteria for assessment

Credit will be awarded against the following criteria.

Specifically, the marks will be apportioned as follows:

- ***Part 1 40 marks*** are available for the clear demonstration and validation of a working functional system including the satisfaction of all the coursework requirements:
  This should include an overview of your solution/system, appropriate satisfaction of all the basic criteria for the exercise (specified below).
  This needs to be clearly demonstrated in the written report:
  You should clearly state what coursework requirements have been met and provide evidence by pointing to suitable code fragments/routines and supplying screenshots to show aspects of the working solution.
- ***Part 2 30 marks*** are available for a clear report write up. The write up should clearly state your design and implementation strategy.
- ***Part 3 30 marks*** are available for your design and incorporation of other features and media beyond the basic coursework specification.
  These should also be clearly evidenced in the written report.

A mark breakdown in terms of the percentage marks is shown below.

	1st	2.1	2.2	3rd	Fail
Part 1 (40 marks)	>= 28	>= 24	>= 20	>= 16	< 16
Part 2 (30 marks)	>= 21	>= 18	>= 15	>= 12	< 12
Part 3 (30 marks)	>= 21	>= 18	>= 15	>= 12	< 12

- **Fail** means the submission does not adequately address the stated requirement for the Part.
- **3rd** means the submission minimally addresses the stated requirement for the Part; for example, many elements are not working as specified, some elements may be working.
- **2.2** means the submission partially addresses the stated requirement for the Part; for example, some elements of the solution may be partially working or the solution has not be fully addressed.
- **2.1** means the submission fully addresses the stated basic requirements for the Part, but has weaknesses in terms of the *weakness indicators* below.
- **1st** means the submission fully addresses the stated requirement for the Part, as well as meeting the *excellence indicators* below.

Factor	**Weakness indicator**	**Excellence indicator**
Approach	Does not adopt a professional or defensible approach to the solution.	Adopts appropriate methods with full justification for the choices made. **Clear evidence that they work.**
Argument	Unstructured and shows little or no evidence of research and the logical justification of the use of appropriate methods or conclusions in how they work.	Approach and methodology Clear justified with a demonstrable working outcome.
Insight and understanding	Little or no insight and understanding.	Has developed considerable insight and understanding. **Good reflection and understanding of what has been accomplished evident.**

Feedback and suggestion for future learning

Feedback on your coursework will address the above criteria.

The Individual project work will be returned via Learning Central.

Individual feedback will be given via Learning Central, at the same time as project work is returned on Learning Central (i.e. within 3 weeks of submission)

Further general feedback will be provided on Learning Central in the Coursework Feedback section in Learning Materials, at the same time as individual feedback.

Submission of Coursework

- **Hand in Date**: Submission of all material (Short Report and MATLAB code) by 9.30AM on Friday 2nd December, 2021 (Week 10, Autumn Semester) via Learning Central.
- This work is the *only* coursework component for the assessment of this module: This piece of work is worth **50%** of the marks for the whole module.

Submission Details

The exercise involves developing an *Interactive Granular Convolution-based Synthesiser* in MATLAB. You must submit a typeset report in **PDF format** plus **all MATLAB code**. Your coursework submission to *Learning Central* **must** include:

- A cover page that details the following:
  - Student number, student name, module code, module title, coursework title, lecturer, hours spent on this exercise, and special provision (if applicable).
- A report of **a maximum of 3000 words** (approximately 3–4 pages of text), with *additional* diagrams, screenshots *etc.*, *which are encouraged*, typeset in **PDF format**, detailing the following:
  - An overview of your program design and implementation.
  - A basic algorithmic description of the main elements of your solution and *how they satisfy the basic requirement* listed below.
  - You should also highlight —*any novel features* or *those above the basic requirements*.
- In **addition** to the short report, a copy of all code together with all the media and other assets required to run the code (such as sound files) should be provided.
  - Include all source code as native .m text files.
  - Ensure your name and student ID is associated with each file.

  - It is your responsibly to ensure that your submitted code can be executed properly on a standalone machine running MATLAB.
    * Make sure that running the program does not require any additional setup (such as configuring paths).
    * Ensure all MATLAB GUI set and other files are supplied.
  - All code files **including all support media assets** should be submitted as single zip file collection.

- **Ensure** that your student number and name prominently appear in **each** file that makes up your submission.

Assessed Coursework

You should develop an **Interactive Granular Convolution-based Synthesiser** with *graphical filtering* capabilities in MATLAB.

The inspiration for the work is a piece of audio software called *Iris* by Izotope Inc **and** a piece of hardware called a Granular Convolver

The Granular Convolver device essentially allows you to record sounds and slice them into small fragments referred to as *grains* — see lecture notes on "Granular Synthesis". Once a sound has been recorded, you can convolve the grains (in any configuration you choose) with any new input you care to send into the device — the original device uses live input, **however**, you ***only need*** to apply it to (other) **pre-recorded** input sound. Convolving these two signals creates completely new and interesting sounds. See this youtube video for more details.

Iris is an innovative sampling Fourier transform-based spectral re-synthesiser. In Iris, you can input up to 3 waveforms to dissect and process them in many ways. Using *Iris*'s *spectrogram display* and easy drawing/selection tools to spotlight the most interesting spectral characteristics you can blend and layer your modified samples with some filters that are unrealisable by standard filter designs. The sounds can then be subsequently processed with other audio effects.

Useful web links:

- Izotope Iris main web page:
  https://www.izotope.com/en/products/create-and-design/iris.html
- Izotope Iris Help:
  http://help.izotope.com/docs/izotope-iris-help.pdf
- Izotope Iris Video Tutorials: Izotope Video Guides/Tutorials

- A *10 day trial* **demo** version of Iris is freely available:
  Iris download page here

Please note that you are **only** required to create a program that *emulates* the basic spectrogram editing and playback functionality of *Iris* **and** implements some form of granular convolver with some additional audio processing described in more detail below. *A full emulation of Iris is not expected!*

Coursework Requirements

The following basic requirements should be met in order to gain average to good marks:

- You should implement a method of reading in an input audio file, compute its underlying time-frequency distribution (*i.e.* **short-term Fourier transform**) and provide an interactive means of filtering this audio via the editing of its displayed spectrogram.
- The resulting edited audio then needs to be played back: Playback need only be **monophonic**, *i.e.* it need only be able to play one note of audio at a time. Playback should be able to control:
  - the **pitch** of the audio — the audio pitch should be able to be played back at a given **new pitch** but at the same tempo.
  - the **tempo** of the audio — the audio tempo should be able to be played back at a given **new tempo** but preserving the same pitch.
- Implement a basic form of **Granular Convolution**:
  - If the short-term Fourier transform is computed on a very small sample window then aspects of granular synthesis can be implemented.
  - A basic description of short-term granular synthesis is discussed in the lectures. However, you are expected to read around and develop a more in-depth and thorough understanding of granular synthesis as part of this coursework exercise.

    - The lecture and tutorial notes reference some basic Granular Synthesis code (which you could build upon and *enhance*) and also demo some Granular synthesisers (publicly and commercially available)
    - Another example of a commercial Granular Synthesiser is: http://www.newsonicarts.com/html/granite.php
    - The lecture notes also reference some basic applications of convolution that you can build upon.
    - Applying one single grain as a convolver is the **basic requirement**, although the potential to include more grains, in a multitude of possible configurations, is encouraged as an **additional feature** (see below).

- **In addition**, you should provide some additional audio processing:
    - You should implement some form of *volume shaping* or *envelope shaping* to further control or modulate the basic sounds produced.
    - You should also provide **TWO** additional audio effects that are applied to the newly synthesised waveforms to provide a wider sound palette. The obvious example here would be some form of equalisation, chorus/phaser/flanger or reverb, although other forms of processing could be provided — you may utilise example audio effect code supplied in lectures and labs for this task.

- **The playback of audio DOES NOT have to operate in real time** – you can read the data in, generate/process the audio and then play the output, much like the demo MATLAB programs in the lectures/tutorials.

- Your basic system need not have an advanced GUI (nothing as elaborate as *Iris* is expected!). You could satisfy all the requirements with a minimal GUI to display the audio/spectrogram and allow basic interactive editing.
    - You may use any GUI development environment to create your basic GUI.
    - MATLAB's App Designer is recommended.
    - MATLAB's older GUIDE environment may also be used.

- Your synthesis/effects *pipeline* for generating sounds to output can be **fixed**, i.e. the order of processing elements can be a series of processes that are hard wired in the MATLAB code: You may filter before or after convolving, envelope shaping before or after either these.

In order to gain higher marks you need provide **TWO** novel extensions or additional features. There are endless possibilities here and you are **encouraged** to come up your **own** ideas for **extensions**. Here are a few suggestions:

- Advanced granular convolution:
  - Applying multiple grains as convolvers in series or, even, parallel.
  - Looping through a short audio sample selecting (in series or randomly) grain convolvers.
  - Selection of a set number of grains as in the original granular convolver hardware.
- Like *Iris*, you could provide multi-layer sample playback using more than one audio source.
- You could provide advanced playback functionality to allow for looping of sections of audio, reversing sections of audio *etc.*
- You could implement some additional digital audio effects to the output synthesised notes — You **may** *re-use* any audio code supplied in lectures or labs to satisfy this sub-task.
- Provide a user-friendly editor for the audio and/or to enter musical data.
  - GUI elements to control the synthesis, filtering/modulation and sound output may be provided.
- Provide support for polyphonic output.
- Provide MIDI support for data input.
- Provide additional methods of digital synthesis. Furthermore you could offer a hybrid synthesis approach where more than one mode of synthesis is used at one time to produce musical notes.

- A *modular* synthesis/effects approach may be developed where the sound generation and subsequent processing pipeline (order of synthesis/effects) is not static and can be configured at runtime.

Some Useful Links

You may find these useful in helping you develop your solution to the above assessed coursework:

- Understanding Granular Synthesis (YouTube Tutorial)
- http://www.jyu.fi/musica/miditoolbox/ — MATLAB MidiToolbox.
  - Reads/Writes Midi files, converts midi between note number, musical notes/pitches and frequencies.
  - See also http://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/exercises_BSC/ for local copy of the MidiToolbox.
- http://labrosa.ee.columbia.edu — MATLAB Audio Processing Examples
- http://www.phy.mtu.edu/ suits/notefreqs.html — musical pitches frequency relationship.
- http://en.wikipedia.org/wiki/List_of_meantone_intervals:

  List of pitch/tone intervals as ratios. (Useful for the phase vocoder — see Tutorial 2 notes).

A quick web search should provide plenty of free audio samples for you to use to develop and test your system. Here are a few links to get you started:

- http://www.loopasonic.com/
- https://www.looperman.com/loops
- http://www.phatdrumloops.com

Prof. David Marshall. October 2021