

## **PROJECT**

### **Instructions & Guidelines**

This project is part of your assessment for the following learning outcomes:

1. Practical skills in developing effective, efficient and user-friendly system
2. Appreciation of possible applications of multimedia systems

You should be able to complete this project by revising the practical exercises related to audio, images and video. The project will be based on MATLAB and will require the knowledge and skills that you have gained within this module.

This project accounts for 10% of your final assessment and is due by 23:55 PM, Friday of Week 9 i.e. 7<sup>th</sup> September 2018. Please follow the following guidelines regarding the code/files:

1. Try to write neat code. For instance, use self-describing variable names, insert comments wherever appropriate
2. Try your best to make the code user-interactive rather than hard coding everything.
3. Refer to the MATLAB online library to understand various functions needed to carry out the project, however do NOT simply copy-paste. It is easy to detect copied code.

### **OBJECTIVE:**

You are write a simple MATLAB application. The application could be command-driven or could have a simple user interface. Extra marks will be awarded for designing a simple user interface. The application should have the following sections, and choice of sections to be run can be prompted from the user:

#### **Section 1: Audio**

Once the user is in this section, the application should prompt the user to pick a wav file from a given list of wav files (you can include a few wav files within your project folder). Once a file has been chosen, the user should be prompted to pick one of the following operations to be performed on the wave file(s):

1. Playback the audio file
2. Increase the volume by 50%
3. Decrease the volume by 50%
4. Concatenate two audio files and play them back
5. Prompt the user asking for the value of an amplitude  $A$  and frequency,  $f$ . Use those values to generate a 3 sine wave audio signals with values  $(A, f)$ ,  $(A/2, 2f)$  and  $(A/3, 3f)$ . Create a tone by combining them (they can be combined in various ways).

#### **Section 2: Images**

Once the user is in this section, the application should prompt the user to pick an image file from a given list of images (you can include a few image files within your project folder). Once a file

# Multimedia

has been chosen, the user should be prompted to pick one of the following operations to be performed on the image file(s):

1. Display the red, green and blue channels of the image individually.
2. Convert the image into grayscale and increase the overall brightness of the image by a user-defined value. Prompt the user for a value (usually between 0-255, think about what to do if the final intensity value exceeds 255) and then use that value to increase the overall brightness by that amount.
3. Cut out a rectangular section from the chosen image, save it and display it as a new image. Set the red channel to 0 and display the new image
4. Apply the log transformation as a contrast enhancement technique on a chosen image (do not use MATLAB's built-in `imadjust` function). Pick an appropriate value for the constant  $c$  (refer to lecture slides).
5. Apply Gamma correction to a chosen image. Make sure you pick an appropriate value of  $c$  and  $\gamma$  (refer to lecture slides).

## Section 3: Video

In this section, start off by recording a video of yourself. Import this video into MATLAB. Implement features that displays information about the video clip such as:

- The duration of the video clip
- The pixel dimension of every frame
- The dimensionality of the array per frame
- Number of bits per pixel
- Number of frames per second
- The number of frames within the video clip

Implement a feature that allows the video clip to be played back. Within this feature, include a code snippet that starts playback from a specified time. The starting time could either be user-specified or hard-coded within the code.

**-END-**