Ziad Mohamed ElGendy

*55-3395*

*T16*

***Signals project milestone 1: Pianist***

This project was implemented in Python using the PyCharm IDE and additional libraries installed using pip, and the following libraries were used: numpy, matplotlib, time, and sounddevice.

The linspace() and zeroes() functions from the numpy library were used to generate array lists to simulate x and y axis for a continuous function with a sample rate of 4096 points per second.

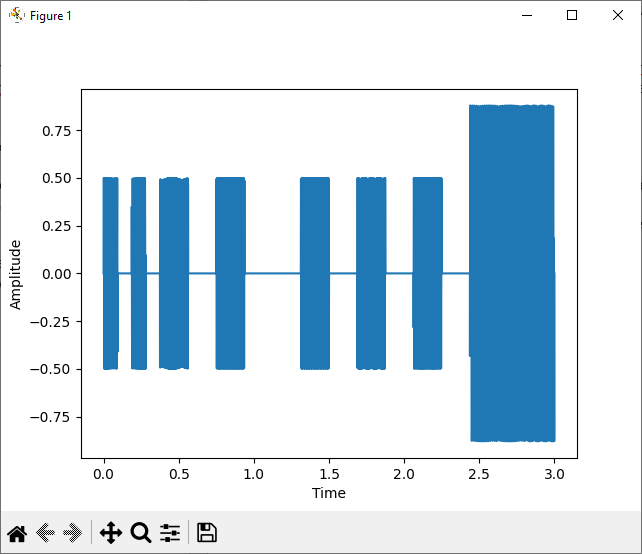
The sin() function from the numpy library was then used multiple times to generate various musical notes, each multiplied by a subtracted step functions to dictate their start and end point.

Matplotlib plots the t and x arrays with the plot(t, x) function using variable t as the x-axis and variable x as the y-axis. The show() function was used as PyCharm lacks a plotting window unlike Spyder.

Sounddevice then plays back the generated signal x as a sound wave. time.sleep() is used to prevent the program from ending before the playback occurs.

Constants for musical notes and musical timings are used to make the code more intuitive.

Certain functions were imported directly for cleaner code.



***Signals project milestone 2: Noise Cancellation***

For the second part of the project, the scipy library was imported for its Fast Fourier Transform function.

2 random frequencies were generated using the np.random.randint() function and added to the original signal. The fft() function was used to plot the signal onto the frequency domain before and after adding the noise.

The noise was detected by iterating through the FFT of x and looking for values that cross a certain threshold, after which the value is added to a ‘Do not re-use’ to prevent repetition due to rounding inaccuracies, and compared to the next value to ensure the value is at its peak.

After finding the noise frequencies they are then converted back to the time domain and subtracted from the original signal.

