CMPM 163 Notes

Malcolm Riley

Winter 2019 — May 14, 2019

Audio in Unity

- Audio in Unity is accomplished by the following two object components: `Audio Source` and `Audio Listener`.

- Typically, the `Audio Listener` is attached to the main camera, and in the default scene it will be set up this way.

- Importing audio assets works the same way as importing other assets in Unity - simple drag-n-drop.

- Implementing `GameObject` response to audio is accomplished via scripts

- The general idea is to populate an array of floating-point values with data from the audio spectrum. This data is then interpreted to the desired effect.

- The interpretation of an audio waveform is based on the principle of the **Fourier Transform**.

- Consider that an audio wave may be represented by the **amplitude** of a function over the domain of time.

- The **Fourier Transform** interprets the amplitude components of the waveform function at different "frequencies"; the number of discrete frequencies being sampled is the "size" of the transform. These samples are typically over a very short duration, usually on the order of milliseconds

- The **Frequency Resolution** is given by the formula $\frac{R}{S}$ where R is the sample rate in Hz, and S is the "size" of the **Fast Fourier Transform**

- The maximum measurable frequency, called the **Nyquist Limit**, is equal to one half of the **Sample Rate**.

- Fortunately, Unity has built-in methods for acquiring **Fast Fourier Transform** data from an `Audio Source`. On an `AudioSource` component, call the `GetSpectrumData()` method! It will populate a `float` array with the desired data.

– When performing audio analysis, there is always a tradeoff. The larger the **Fast Fourier Transform** size, the higher the frequency resolution, but computational expense increases. Furthermore, resolution scales over the two domains independently: the frequency domain, and the time domain.