Assignment 2 Design Document

Adam Rowland

15/05/20

1. Overview

Part 2 of my assignment is a sequencer that supports triangle wave notes played with a linear ADSR envelope. This document covers the implementation details of the sequencer as well as some possible improvements.

2. Linear ADS[illegible] definition

A linear ADSR [illegible] of altering the amplitude over the [illegible] t contains four different sections [illegible] stain and release. In the attack section, the amplitude increases linearly from 0 to a predefined maximum. In the decay section, the amplitude decreases linearly from the maximum amplitude in the attack section to the constant amplitude of the sustain section. In the sustain section, the amplitude remains fixed at a predefined amplitude. In the release section, the amplitude decreases linearly from the amplitude in the sustain section to 0. All other aspects of the note such as frequency and timbre, remain constant. The general shape can be seen in Figure 1.

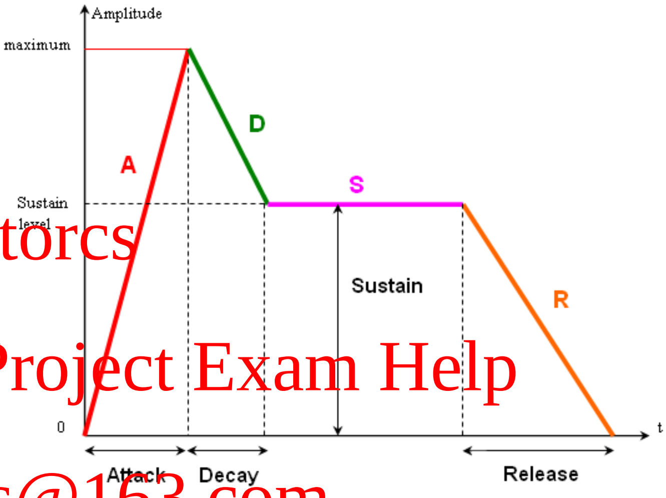

Figure 1: Linear ADSR envelope

3. Implementation

3.1 Memory structure

Notes are encoded in memory as an array of records. Each record contains the frequency in Hz, the maximum amplitude, the total duration of the note, the duration of the attack section, the duration of the decay section, the amplitude of the sustain section and the duration of the release section. All durations are measured in 100th's of a second. Changing these values allows for different pitches and durations, as well as different envelopes as long as they are linear ADSR. Rests are encoded with a 0 half-word and then their duration.

3.2 Note and rest generation

Each note is played using a triangle wave. This waveform has constant incline and decline and consistent peaks over each period. An example can be seen in Figure 2. Triangle waves requires less precise and less complicated computations than other kinds of waves such as sine waves and thus lend themselves well to amplitude modulation over several periods. As a result, the frequencies and durations of the notes may not be exact. Rests are played by writing constant 0 for `frequency*duration/100` samples.

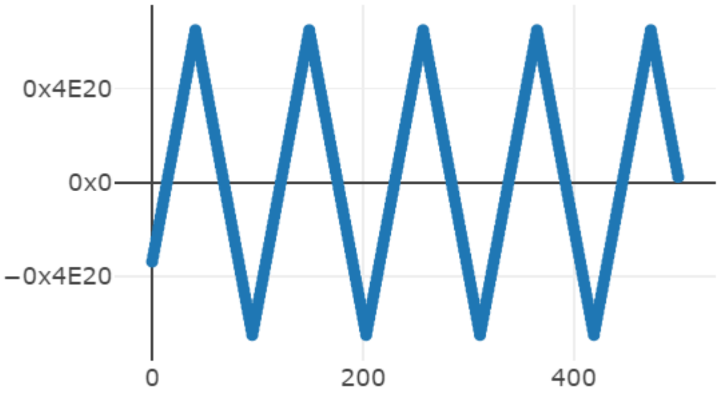

Figure 2: Triangle wave

3.3 ADSR envelope

Figure 3 shows a triangle wave under an ADSR envelope with attack duration 1, decay duration 1, sustain amplitude 0x5ffd and release duration 2. In each segment of the envelope, the number of periods is calculated as frequency*duration/100 and the increase or decrease in amplitude per period is [illegible] amplitude - min_amplitude)/no_of_periods. In the case of the attack and re[illegible] minimum amplitude is 0.

The frequenc[illegible] nstant throughout the envelo[illegible] f the attack, decay and releas[illegible]. The duration of the sustain [illegible] from the total duration of the note and the durations mentioned previously. Maximum amplitudes are given for the attack and release sections, while a constant amplitude is given for the sustain section and the amplitude range is given for the decay section. This is common practice. Amplitude values are calculated before writing each period to simplify the calculations.

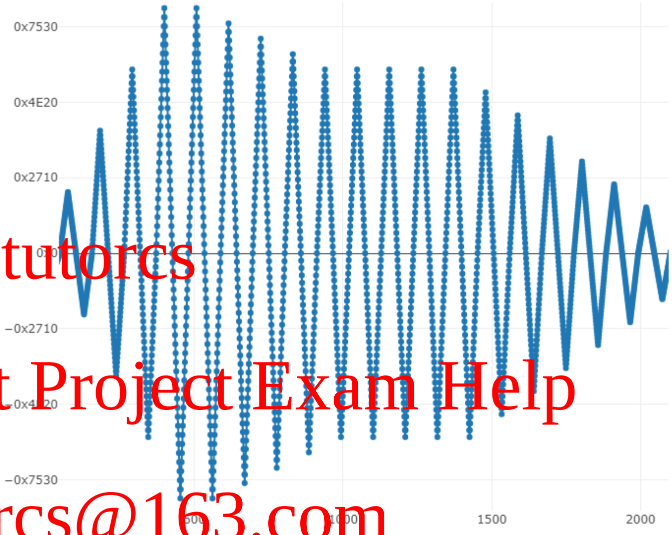

Figure 3: Triangle wave with linear ADSR envelope

4. Improvements

The sequencer currently only supports triangle waves. It could be extended to include saw waves and square waves by altering the wave_period function. More complex waves such as sine waves with lookup tables would require fine-tuned amplitude scaling and may require a major redesign in implementation.

The amplitude envelope could be extended to more generalised envelopes. For example, the increases and decreases in amplitude could be made non-linear or a completely different envelope system could be used.

The lack of precision sometimes results in varied pitch. The pitch can drop slightly especially for low frequency and amplitude values. Pitch can also be slightly off for certain frequency values. Use of the floating point unit may solve some of these issues, although the samples must still be rounded to 16-bit integers.

The records in memory could be made more space efficient. The durations could be encoded in less than a half-word, and rests could also be encoded more efficiently. The storage could be made more compact by either reducing the space needed for each record or alternatively encoding more information in the space given.