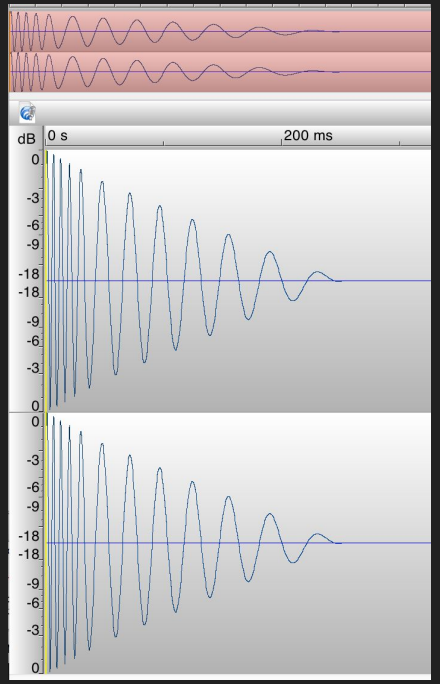
Sub Kick Generator

EP-353 Final Project, by Aaron Wolff

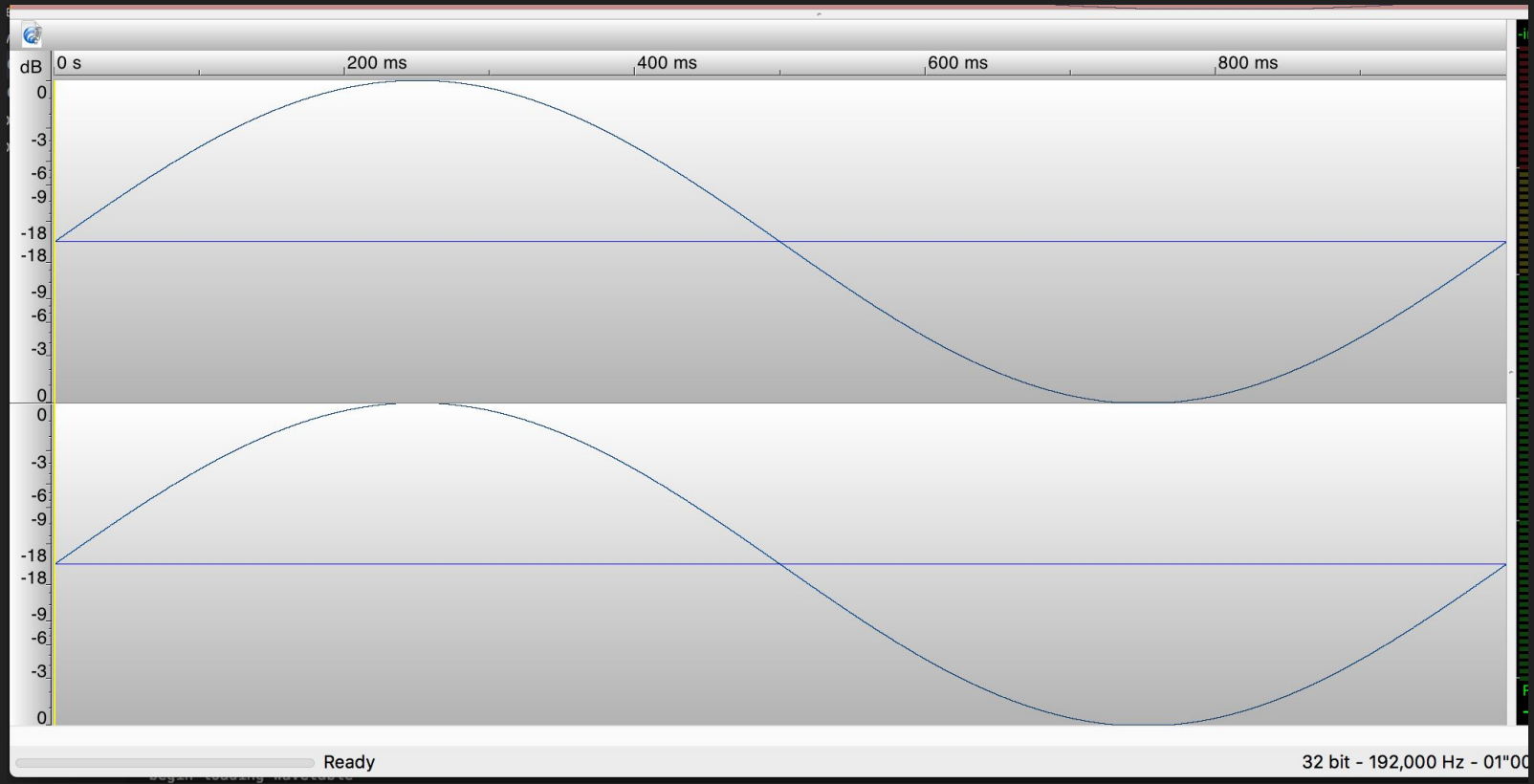
Making kicks with numbers is very reliable

```
instr 1
kamp linseg 1, 0.04, 0.8, 0.2, 0
kcps linseg 200, 0.04, 60, 0.7, 60, 0.2, 20
a1 oscil kamp, kcps
out a1
endin
```



```
size_t kickgen(float* buffer, int initfreq, int attacktime, int primfreq, char yndescend, int resttime, int fadetime){
    float current_amp = 0;

    Envelope* AmpEnvelope;
    Envelope* FreqEnvelope;
    //trying to initialize the lengths of both envelopes (getCurrentAmp and getCurrentPitch)
    //I made get current pitch so that I could have two separate envelopes for pitch and amplitude
    initEnvelopeFreq(FreqEnvelope, (float)(attacktime+resttime+fadetime), initfreq, attacktime, primfreq, yndescend, resttime, fadetime, 192000.05);
    initEnvelopeAmp(AmpEnvelope, (float)(attacktime+resttime+fadetime), initfreq, attacktime, primfreq, yndescend, resttime, fadetime, 192000.05);

    Wavetable* wavetable;
    loadwavetable(wavetable);

    int i=0;
    for(; i<AmpEnvelope->totalFrames; i++){
        setFrequencyHz(wavetable, getCurrentFreq(FreqEnvelope));
        current_amp = getCurrentAmp(AmpEnvelope);
        buffer[i] = current_amp * next(wavetable);
    }

    return (size_t) AmpEnvelope->totalFrames;
}
```

Using a Wavetable



Wavetable.h and Wavetable.c

Thank you Akito

```c
#pragma once

#include <stdlib.h>
#include <stdbool.h>
#include <math.h>
#include "Audio.h"

#define kDefaultFrequency 440.0
#define kTableSize (1<<7) //128

typedef struct Wavetable {
  float *table;
  unsigned long size;
  float curIndex;
  float delta;
  bool isPlaying;
  float amplitude;
} Wavetable;

// Function Prototypes
void createWavetable(Wavetable *wavetable);
void initWavetable(Wavetable *wavetable, unsigned long length);
void writeToWavetable(Wavetable *wavetable, float* buffer, unsigned long length);
float next(Wavetable *wavetable);
void setFrequency(Wavetable *wavetable, char *midi);
void setAmplitude(Wavetable *wavetable, float amplitude);
void setFrequencyHz(Wavetable *wavetable, float freq_Hz);
void loadwavetable(Wavetable *wavetable);
```

Envelope.h, with PitchEnvelope.c and AmpEnvelope.c

```
  envelope->points[0].x = 0;
  envelope->points[0].y = 1;
  envelope->points[1].x = ((float)attacktime/1000)*192000;
  envelope->points[1].y = 0.8;
  envelope->points[2].x = ((float)resttime/1000)*192000;
  envelope->points[2].y = primfreq;
  envelope->points[3].x = ((float)fadetime/1000)*192000;
  envelope->points[3].y = 0;

  //Initialize points in the envelope
  for(int i=0; i<kNumPoints; i++){
    envelope->points[i].x = xUnit * i;
    //Change y points to have different envelope shapes
    envelope->points[i].y = kMaxAmp - yUnit * i;
  }
  envelope->points[kNumPoints-1].x = envelope->totalFrames;
  envelope->points[kNumPoints-1].y = 0.0f;
}

void resetAmp(Envelope *envelope){
  envelope->curFrame = 0;
  envelope->curIndex = 0;
  envelope->gate = false;
}

float getCurrentAmp(Envelope *envelope){
  static Point a, b;
  static float m, y; //slope and amp

  //Check to make sure the gate is on
  if(!envelope->gate) return 0;
```

```
  envelope->points[0].x = 0;
  envelope->points[0].y = initfreq;
  envelope->points[1].x = (attacktime/1000)*192000;
  envelope->points[1].y = primfreq;
  envelope->points[2].x = resttime;
  envelope->points[2].y = primfreq;
  envelope->points[3].x = fadetime;
  if(yndescend == 'y'){
    envelope->points[3].y = 20;
  }else if(yndescend == 'n'){
    envelope->points[3].y = primfreq;
  }



  //Initialize points in the envelope
  for(int i=0; i<kNumPoints; i++){
    envelope->points[i].x = xUnit * i;
    //Change y points to have different envelope shapes
    envelope->points[i].y = kMaxAmp - yUnit * i;
  }
  envelope->points[kNumPoints-1].x = envelope->totalFrames;
  envelope->points[kNumPoints-1].y = 0.0f;
}

void resetFreq(Envelope *envelope){
  envelope->curFrame = 0;
  envelope->curIndex = 0;
  envelope->gate = false;
}
```

```
W E L C O M E    T O    T H E    F A T   K I C C   G E N E R A T O R

Starting Frequency (Hz): 200

Attack Time (ms): 40

Resting Frequency (Hz): 60

Descend? (y/n): y

Ok, How long to rest before going down? (ms): 150

How long should it take to go down? (ms): 200
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
W E L C O M E     T O     T H E     F A T     K I C C     G E N E R A T O R
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Starting Frequency (Hz): 200

Attack Time (ms): 40

Resting Frequency (Hz): 60

Descend? (y/n): y

Ok, How long to rest before going down? (ms): 150

How long should it take to go down? (ms): 200

init wavetable
a
begin loading wavetable
b
d
Segmentation fault: 11
Aarons-MacBook-Pro:Aaron Wolff EP353 Final copy aaron$
```