**Iteration Plan Document**

TeensyAudio Wavetable Synthesis

Iteration #1

**Abstract**

The purpose of this project is to provide a C++ library and accompanying Python utility scripts allowing realistic instrumentation audio to be synthesized on the Teensy 3.2 Arduino Digital Analog Converter (DAC). This library will be exposed to developers, and will allow pitch shifting, looping, tremolo, and vibrato effects to be imposed on a raw byte buffer of recorded samples.

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Description | Date | Modifier |
| 100 | Initial Version | 11/18/2016 | Ryan Mellmer |

**Project Team Roles - Iteration #1**

Ryan Mellmer - Team Lead

Connor Delaplane - Infrastructure Manager

Aida Keifer - Scheduling Manager

Nicholas Craig - Requirements Manager

Josh Bucklin - Design/Architecture Lead (Teensy Library)

Jonathan Jensen - Developer

Xuan Tang - Developer

Note: All members also acting as developers.

**1.0 Introduction**

The purpose of this document is to describe the schedule and iteration artifacts for Iteration 1 of this project.

**1.1 Purpose**

The completion of this iteration completes the following milestones for the project:

* Base API
  + Implement basic modular interface. This will allow the main library functions to be called on demand either by the Teensy code or by another C++ program.
* Wavetable synthesis on Teensy
  + Translation of synthesis prototype script (interpolation functionality only) written in Iteration 1 to C++ code which can run on the Teensy microprocessor in real time.
* Script to extract samples from a SoundFont file
  + A script, written in Python, which will take a SoundFont file and produce a C++-formatted byte array of instrument samples for easy use in wavetable synthesis on the Teensy.

**1.2 Context**

This iteration begins the project by producing a script that allows for data extraction from SoundFont Files. Additionally, a very simple version of the wavetable synthesis class will be implemented for the Teensy during this implementation.

**2.0 Plan**

At the end of this iteration, we will have met these requirements:

**SoundFont Decoding**

* Decode raw audio byte data from SF2 file, producing data for attack, sustain, and decay sections for 1 instrument note.

**Wavetable Synthesis Library for Teensy**

* Implement the very basic interpolation functionality in C++ to run on the Teensy. Also begin generating the base API for the Wavetable class, ensuring that all functionality is properly exposed and formatted for external use.

**2.1 Schedule of Iteration Workflows**

|  |  |  |  |
| --- | --- | --- | --- |
| **Workflow** | **Start Date** | **End Date** | **Duration (days)** |
| Requirements | 11/01/16 | 11/15/16 | 14 |
| Analysis and Design | 11/14/16 | 11/28/16 | 14 |
| Implementation | 01/09/17 | 01/23/17 | 14 |
| Testing | 01/09/17 | 01/23/17 | 14 |

**Table 1 :** Iteration Workflow Schedule

**2.2 Iteration Schedule Breakdown**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Assigned To** |
| **Requirements** |  |  |  |
| Acquire confirmation from Paul | 11/14/16 | 11/15/16 | Requirements Team |
|  |  |  |  |
| **Analysis and Design** |  |  |  |
| Begin research topics / write summaries | 11/07/16 | 11/14/16 | All team members |
| Drafting design documents | 11/17/16 | 11/28/16 | All team members |
|  |  |  |  |
| **Implementation** |  |  |  |
| Python script decoding SF2 files | 01/09/17 | 01/18/17 | Subteam 1 |
| C++ Wavetable class implementation | 01/09/17 | 01/23/17 | Subteam 2 |
| Basic Interpolation algorithm | 01/09/17 | 01/23/17 | Subteam 3 |
| **Testing** |  |  |  |
| Python script decoding SF2 files | 01/09/17 | 01/18/17 | All team members |
| C++ Wavetable class testing | 01/09/17 | 01/23/17 | All team members |
|  |  |  |  |
| **Documentation** |  |  |  |
| Python SF2 decoding script developer documentation | 01/20/17 | 01/23/17 | SF2 Decoding Team |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table 2:** Iteration Plan Task Breakdown by Workflow

**2.3 Iteration Artifacts**

|  |  |  |
| --- | --- | --- |
| **Task Name** | **Deliverable** | **Responsible** |
| **Requirements** |  |  |
| Acquire confirmation from Paul | Requirements document | Requirements Team |
|  |  |  |
| **Analysis and Design** |  |  |
| Complete Research topics | Research Write-ups | All team members |
| Finalize design document | Finalize design document | All team members |
|  |  |  |
| **Scheduling** |  |  |
| Schedule for iterations 1 and 2 | Scheduling document (this) | Scheduling subteam |
|  |  |  |
| **Implementation** |  |  |
| Python script decoding SF2 files | decoder.py (extract audio data) | Decode team |
| C++ Wavetable class | wavetable.h, wavetable.cpp | Teensy team |
|  |  |  |
| **Testing** |  |  |
| Python script decoding SF2 files | Decoding Test Plan | Decode team + all**[[1]](#footnote-0)** |
| C++ Wavetable class initial tests | Prototype Test Plan | Teensy team + all |
|  |  |  |
| **Documentation** |  |  |
| Python SF2 decoding script developer documentation | Developer Documentation documents for SF2 decoding script | SF2 Decoding Team |

**Table 3:** Artifacts to be Delivered in this Iteration

1. The subteam responsible for the development of an artifact will play the lead role in its testing, but all subteams will play a role in testing each artifact. [↑](#footnote-ref-0)