**Iteration Plan Document**

TeensyAudio Wavetable Synthesis

Iteration #2

**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**

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 2 of this project.

**1.1 Purpose**

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

* 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.
* 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.
* Looping
  + Provide Python implementation functionality for the “loop” section of the audio data to be looped on demand and indefinitely until instructed to stop.
* SF2 decoding script refinement
  + Optimization and cleaning of SF2 decoding script.

**1.2 Context**

The primary focus of this iteration is to implement the most basic functionality of the Teensy wavetable synthesis library. The basic use cases after completion of this iteration will be playback, looping, and interpolation synthesis of SF2 audio data on the Teensy microprocessor. This core functionality will provide the foundation necessary to implement and test routines to be implemented in later iterations, such as those for tremolo and vibrato.

**2.0 Plan**

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

**SoundFont Decoding**

* Refine script

**Modeling of Wavetable Synthesis Library**

* An audio sample’s sustain data can be looped

**Wavetable Synthesis Library for Teensy**

* Translate code from modeling script (from iteration 1) to C++

**2.1 Schedule of Iteration Workflows**

|  |  |  |  |
| --- | --- | --- | --- |
| **Workflow** | **Start Date** | **End Date** | **Duration (days)** |
| Requirements | 01/24/17 | 01/28/17 | 4 |
| Analysis and Design | 01/24/17 | 01/28/17 | 4 |
| Implementation | 01/29/17 | 02/12/17 | 14 |
| Testing | 01/29/17 | 02/12/17 | 14 |

**Table 1 :** Iteration Workflow Schedule

**2.2 Iteration Schedule Breakdown**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Assigned To** |
| **Requirements** |  |  |  |
| Update any new requirements | 01/24/17 | 01/28/17 | Requirements Team |
|  |  |  |  |
| **Analysis and Design** |  |  |  |
| Update Design plan | 01/24/17 | 01/28/17 | Everyone |
|  |  |  |  |
| **Implementation** |  |  |  |
| Translate python prototype to Teensy code | 01/29/17 | 02/12/17 | Subteam 1 (Teensy) |
| Refine decoding script | 01/29/17 | 02/12/17 | Subteam 2 (Decode) |
| Python prototype script (looping) | 01/29/17 | 02/12/17 | Subteam 3 (Proto) |
|  |  |  |  |
| **Testing** |  |  |  |
| Provide unit tests for Teensy Wavetable Library as it’s being developed | 01/29/17 | 02/12/17 | Teensy Library Developers |
| Refine decoding script | 01/29/17 | 02/12/17 | Decoding Script Developers |
| Test looping on python prototype | 01/29/17 | 02/12/17 | Prototype developers |
|  |  |  |  |
| **Developer** |  |  |  |
| SF2 update developer documentation | 02/09/17 | 02/12/17 | Subteam 2 (Decode) |
| TeensyAudio Wavetable library developer documents | 02/09/17 | 02/12/17 | Subteam 1 (Teensy) |

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

**2.3 Iteration Artifacts**

|  |  |  |
| --- | --- | --- |
| **Task Name** | **Deliverable** | **Responsible** |
| **Requirements** |  |  |
| Update any new requirements | Potentially updated requirements document | Requirements Team |
|  |  |  |
| **Analysis and Design** |  |  |
| Update Design plan | Potentially updated design document | Everyone |
|  |  |  |
| **Implementation** |  |  |
| Translate python prototype to Teensy code | wavetable.h, wavetable.cpp | Subteam 1 (Teensy) |
| Refine decoding script | decoder.py | Subteam 2 (Decode) |
| Python prototype script (looping) | wave\_proto.py (with looping) | Subteam 3 (Proto) |
|  |  |  |
| **Testing** |  |  |
| Provide unit tests for Teensy Wavetable Library as it’s being developed | Teensy Test Plan | Subteam 1 (Teensy) + all**[[1]](#footnote-0)** |
| Refine decoding script | Decoding Test Plan | Subteam 2 (Decode) + all |
| Test looping on python prototype | Prototype Test Plan | Subteam 3 (Proto) + all |
|  |  |  |
| **Documentation** |  |  |
| SF2 update developer documentation | Developer documentation documents for SF2 decoding script | Subteam 2 (Decode) |
| TeensyAudio Wavetable library developer documents | Developer documentation documents for Teensy Library | Subteam 1 (Teensy) |

**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)