**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:

* Looping (Teensy)
  + Provide functionality for looping “sustain” data on the Teensy, along with the real-time interpolation functionality.
* Voice Allocator
  + Generate a C++ class for handling voice allocation and keeping track about how many and which notes are being played at any one time.
* Base API
  + Refine the base API.
  + Begin creating developer documentation for API.
* SF2 decoding script refinement
  + Allow for pulling multiple samples (ex. two different intensity recordings for the same sample).
  + Optimization and cleaning of SF2 decoding script.

**1.2 Context**

The primary focus of this iteration is to implement looping functionality into the Teensy Wavetable.cpp object, to allow looping of a sample’s sustain phase. The basic use cases after completion of this iteration will be playback 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

**Looping**

* Extend C++ functionality to allow looping of a sample for as long as the user desires.

**2.1 Schedule of Iteration Workflows**

|  |  |  |  |
| --- | --- | --- | --- |
| **Workflow** | **Start Date** | **End Date** | **Duration (days)** |
| Requirements | 01/18/17 | 01/24/17 | 6 |
| Analysis and Design | 01/18/17 | 01/24/17 | 6 |
| Implementation | 01/24/17 | 02/06/17 | 14 |
| Testing | 01/24/17 | 02/06/17 | 14 |

**Table 1 :** Iteration Workflow Schedule

**2.2 Iteration Schedule Breakdown**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Name** | **Start** | **Finish** | **Assigned To** |
| **Requirements** |  |  |  |
| Update any new requirements | 01/18/17 | 01/24/17 | Requirements Team |
|  |  |  |  |
| **Analysis and Design** |  |  |  |
| Update Design plan | 01/18/17 | 01/24/17 | Everyone |
|  |  |  |  |
| **Implementation** |  |  |  |
| Sample looping | 01/29/17 | 02/06/17 | Josh, Nicholas, Ryan |
| Refine decoding script, add ulaw encoding | 01/24/17 | 02/06/17 | Xuan, Jonathan, Ryan |
| Voice Allocator | 01/26/17 | 02/06/17 | Aida, Connor |
|  |  |  |  |
| **Testing** |  |  |  |
| Provide unit tests for Teensy Wavetable Library as it’s being developed | 01/29/17 | 02/06/17 | Teensy Library Developers |
| Refine decoding script | 01/29/17 | 02/06/17 | Decoding Script Developers |
| Ensure voice allocator works as expected by running unit/integration tests around the class. | 02/03/17 | 02/06/17 | Voice allocator developers |
| **Developer** |  |  |  |
| SF2 update developer documentation | 02/01/17 | 02/06/17 | Subteam 2 (Decode) |
| TeensyAudio Wavetable library developer documents | 02/01/17 | 02/06/17 | 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** |  |  |
| Sample looping | wavetable.h, wavetable.cpp | Josh, Nicholas, Ryan |
| Refine decoding script | decoder.py | Xuan, Jonathan, Ryan |
| Voice Allocator | voice\_allocator.h, voice\_allocator.cpp | Aida, Connor |
|  |  |  |
| **Testing** |  |  |
| Provide unit tests for Teensy Wavetable Library as it’s being developed | Validated Teensy code | Subteam 1 (Teensy) + all**[[1]](#footnote-0)** |
| Refine decoding script | Validated decoding code | Subteam 2 (Decode) + all |
| Ensure voice allocator works as expected by running unit/integration tests around the class. | Voice Allocator test plan/validated voice allocation code | Subteam 3 (Decode) + 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)