SynthPOIsizer
Poi Instrument

Janina Schroeder, version 2

6/12/2022

Contents

1 Formal Information

1.1 Student

Name	Janina Schroeder
E-Mail	jelona@posteo.net janina_jessika_jelena.schroeder@smail.th-koeln.de
Student ID No	11132206
Phone	+49 178 3130752

1.2 Organizational Information

Proposed start date	11/01/2022: registration of thesis topic
Proposed end date	01/31/2023: submission of thesis Nachteilsausgleich: 03/31/2023
Weekly workload	40 hours per week

2 Introduction

2.1 Aim of Project

I want to create an electronic musical instrument based on poi juggling. Basically, the movement of the pois while playing them should be interpreted into electronic sounds by self-programmed software. Ideally, the variance in movement should be audible in the generated sounds.

2.2 Scope

The instrument should can be best played by artistic performers who can handle various poi choreographies. A laptop and my programmed software is needed to generate these sounds with the poi instrument. On stage, the generated sound output can be used as additional sound source to refine DJ sets or the instrument can serve as a stand-alone synthesizer.

3 Description

The poi instrument uses sensor data of an accelerometer and gyroscope in each poi as control input for sound parameters of a software synthesizer in form of a VST plugin written in C++. The sensor data should be sent via wifi and UDP protocol.

3.1 Hardware Technology

Each of the 2 pois should have the following hardware built-in:

- Arduino Nano 33 IoT Microcontroller
- u-blox NINA-W102 radio module for wifi transmission on Arduino IoT
- IMU LSM6DS3 3D accelerometer, 3D gyroscope module for movement data on Arduino IoT
- rechargable battery
- LEDs for night visibility, possibility to turn the ligths on/off
- power button to turn on/off sending of sensor data

3.2 Software Technology

The software stack consists of the following libraries:

- capPlusPlus C++ library for packet capturing and parsing UDP packets
- Juce C++ Framework for VST plugin with software synthesizer
- WiFiNINA Arduino library for radio module
- VST plugin should support Windows, MacOS and Linux platforms
- VST plugin should be installable and used by common DAWs like Ableton Live, Cubase, Logic Pro, etc.

4 MCU Program Requirements

4.1 Microcontroller Program Functions

The program flashed on each Arduinos should meet the following requirements:

- Connecting to the user's wifi
- Reading accelerometer x, y and z axis values
- Reading gyroscope x, y and z axis values
- Packing of accelerometer and gyroscope data in UDP packets
- Sending an contious stream of UDP packets to the user's computer via Arduino's radio module

5 VST Plug-in Requirements

5.1 VST Plug-in Functions

The VST plug-in must realize the following functions:

- Receiveing of UDP packets and interpreting of sensor data
- Mapping functionality of individual or a combination of sensor values to non-binary synthesizer parameters (individual mapping for each poi)
- Calculation of speed, acceleration, velocity and orientation of pois based on sensor data
- Interpretation of calculations to reasonable sound paramaters, the sensor values need to be converted to reasonable sound parameter values
- User Interface with visual feedback of sound parameters being modified by the pois
- Connection possibility with pois by providing wifi password of user's wifi
- Sound Synthesis functionality
- Manual sound parameter Modification without use of poi instrument (like normal VST plug-in)
- Recording functionality
- Conversion of sound parameters to Midi messages (interface functionality to DAW)

5.2 VST Plug-in mappable Synthesizer Functions

The functions of the synthesizer are mostly optional and are only limited by time. At least the oscillator 1 and 2 frequency and volume should be implemented for demonstration purposes of the instrument. The list can be continued.
Poi-mappable synthesizer parameters:

- Oscillator 1: Frequency, Phase, Volume
- Oscillator 2: Frequency, Phase, Volume
- Filter 1: Frequency, Drive
- Filter 2: Frequency, Drive
- LFO 1: Depth, Resonance
- LFO 2: Depth, Resonance
- Effects like: Chorus, Reverb, Delay, Flanger, Phaser or Distortion

5.3 VST Plug-in non-mappable Synthesizer Functions

The synthesizer should be at least able to generate sound without the use of the pois.
Optional additional non-mappable synthesizer parameters:

- Oscillator 1: SubOsc on/off, waveform (Saw, Triangle, Sine, Square), Unison
- Oscillator 2: SubOsc on/off, waveform (Saw, Triangle, Sine, Square), Unison
- Filter 1: Filterform (HPF, BPF, LPF)
- Filter 2: Filterform (HPF, BPF, LPF)
- Load wavetables

6 Project Schedule

This plan is valid for 3 months, since the Nachteilsausgleich was not yet approved. The writing of the Bachelor thesis starts from the beginning and continues throughout the whole project time.

time before 11/01/2023	tutorials about Juce (youtube), Arduino test programs, project research, learn how to write a Bachelor thesis
11/01/2022 - 11/14/2022 (2 weeks)	Arduino send program, VST plug-in receive program, wifi connection,
11/15/2022 - 12/12/2022 (4 weeks)	Synthesizer functionalities without mapping, User Interface
12/13/2022 - 12/26/2022 (2 weeks)	Construction of pois with Arduinos
12/27/2022 - 01/09/2023 (2 weeks)	Mapping functionality, calculations and interpretation of data
01/10/2023 - 01/31/2023 (3 weeks)	Recording functionality interface to DAW, conversion to Midi

7 Resources

https://store.arduino.cc/products/arduino-nano-33-iot
https://github.com/arduino-libraries/WiFiNINA
https://pcapplusplus.github.io
https://docs.juce.com/master/index.html
https://github.com/mtytel/vital

project Github Repo (public):
https://github.com/JaninaAlona/Poi-Instrument