UNIVERSITÄT BAYREUTH

Fakultät Mathematik, Physik, Informatik
Institut für Informatik
Lehrstuhl für Serious Games
Prof. Dr. Jörg Müller

Big Master Project

The Haptic Printer

Author : **Anuj Sharma**,
Akshaya Kumar Krishnappa Ramakrishna

Supervisor : **Viktorija Paneva**

September 27, 2021
Version: Final

Abstract

Over the years of evolution of human computer interaction, we have seen various kinds of interaction techniques. The recent trend of contactless haptic technology has been on the rise. In our project we focus to render custom shapes on the user palm, using different kinds of rendering techniques. The main objective of our project is to map various kinds of user input like CSV, SVG, drawing from canvas and inputs from leap motion. We try to accept these kind of inputs and process them in order to render custom shapes. Also, we try to analyse and compare the different techniques of rendering with validation techniques and deduce results from the study.

Contents

1 Introduction

««< Updated upstream =======

»»»> Stashed changes

1.1 Postcards: My Address

Ricardo Langner
Alfred-Schrapel-Str. 7
01307 Dresden
Germany

1.2 Motivation and Problem Statement

1.3 Results

1.3.1 Some References

[**WEB:GNU:GPL:2010**; **WEB:Miede:2011**]

1.4 Thesis Structure

Chapter 2

Chapter ??

Chapter 3

Chapter 3

Chapter 6

Related Work 2

> *A picture is worth a thousand words. An interface is worth a thousand pictures.*
>
> — **Ben Shneiderman**
> (Professor for Computer Science)

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

2.1 Related Work Section 1

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

2.2 Related Work Section 2

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

2.3 Related Work Section 3

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

2.4 Conclusion

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

3 Features

> *A good end product is not just a collection of features, Its how it all works together.*
>
> — **Tim Cook**
> -CEO Apple Inc.

As mentioned previously the main objective of the application is to consume multiple types of inputs from user and render it to the Ultrahaptics device. The features are built keeping those multiple inputs in mind. A display of multiple inputs can be seen in the image below. The features that are supported in the application are

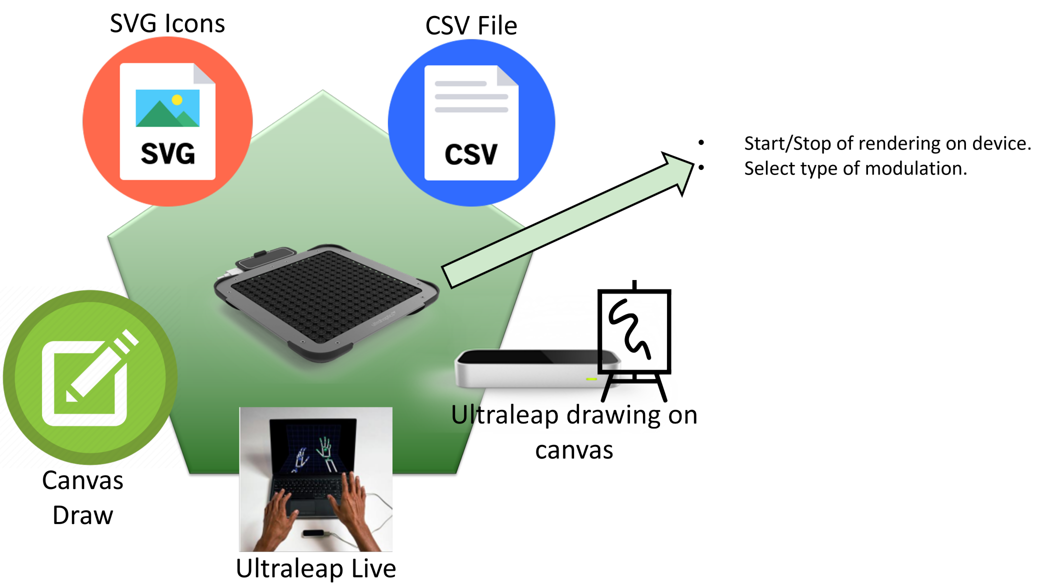

Fig. 3.1: Figure: Features of the application

categorized in three part:

a) Output: Ultrahaptics rendering.
b) Inputs: multiple types of inputs supported.
c) Controls: how user can control rendering.

and are explained in brief further on.

3.1 Output

Since this application revolves around one major output, it is important to mention the output in the beginning. The single output of the application is the custom dynamic shape to be rendered/emitted by the ultrahaptics device. Since, the device has limited size each input coordinate/s should be scaled accordingly.

3.2 Inputs

Multiple types of inputs are accepted in the application and tabs are created for different inputs, these tabs can be seen in fig 3.2:

Fig. 3.2: Figure: Tabs in WebApp

CSV File

A CSV file which has the coordinates of points in an x,y plane in column family. It can be uploaded in File tab. It assumed that the coordinates are mentioned in meters. But since the size of ultrahaptics is 16mm $\times$ 16mm the values in CSV are expected to be in that range. For eg the coordinate (4,5) mm should be 0.004, 0.005 in CSV file. An example of the csv input plotted on an xy plane can be seen in figure 3.2 only for the visual purpose.
In other forms of inputs to the application, the coordinates are scaled and centered

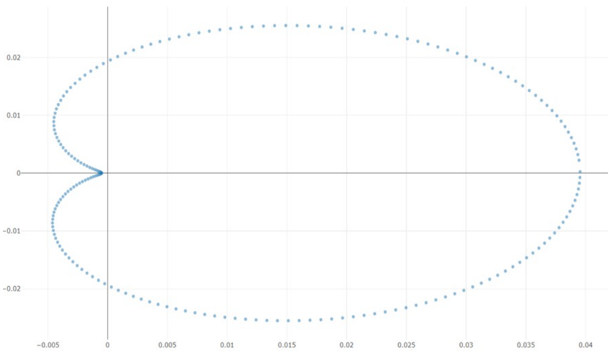

Fig. 3.3: Figure: example csv Plotted on xy plain

to size of the Ultrahaptics device by the application. However in csv input, it is assumed that the coordinates are scaled and centered. As soon as the user clicks on the render button the coordinates are transferred to the Ultrahaptics device and it starts rendering.

SVG File

An SVG is also accepted as input format in the same tab as csv. As soon as the user uploads the SVG its thumbnail is shown on the same page to confirm the visibility. SVG coordinates are then communicated to the ultrahaptics device for that shape to be emitted after user clicks on render button.

Canvas Draw

This feature can be accessed under Canvas tab. A HTML canvas is provided in which user can draw basic shapes and patterns. The coordinates from the canvas is then communicated to the backend and rendered on the ultrahaptics device.

Ultraleap Live

Ultraleap[ultraleap.com] is an advanced hand tracking hardware sensory device that accepts hand and finger gestures as input, similar to a mouse, but without the need for physical contact. It is is a tiny USB peripheral device that is meant to be put on a physical desktop with its face forward. It's also compatible with virtual reality headsets. The gadget observes a roughly hemispheric region to a distance of about 1 meter using two monochromatic infrared cameras and three infrared LEDs[Wei+13]. The LEDs emit patternless infrared light, and the cameras capture almost 200 frames per second of reflected data[Lea]. This is then transferred to the connected computer via USB connection, where it is processed by the company's proprietary hidden code, synthesizing 3D position data by comparing the 2D frames recorded by the sensors.

Fig. 3.4: Figure: Leap motion controller

We consume the data provided by the leap controller and mimic the motion, movement and direction of moving index finger on the ultrahaptics board. another requirement of the feature is also an ultraleap device connected and setup on the system where the webpage is accessed. This feature is accessible in the tab leap live.

Ultraleap

Similar to the ultraleap live user can draw shapes on a canvas provided in WebApp using the leap motion controller. As soon as the user clicks on render button the shape is rendered on ultrahaptics board.

Note: To start tracking the index finger by the device on WebApp. First click on the Start Leap button provided in our WebApp and then do the click/tap gesture by your index finger above Ultraleap motion controller device. After that you will get a banner notification that the system is tracking your finger. Do the similar gesture to stop the tracking.

3.3 Controls

Some basic controls are also provided in the applications where user can secect and control other factors, such as:

Stop button:

On top right of our WebApp a stop button is provided, the rendering on the ultrahaptics device can be stopped anytime using this button.

Selecting type of rendering:

On top left of the webapp there is a dropdown select. Here user can switch between Amplitude Modulation or Time Point streaming for rendering. Once user has selected AM, and now wants to observe TPS, rendering has to be stopped by the stop button before switching.

4 Architecture

> *Any application that can be written in JavaScript, will eventually be written in Javascript.*
>
> — **Jeff Atwood**
> Code Horror (blog)

This section describes the usage and mapping of hardware and software of this application. This section will clear out some questions about the application like. How the application is consuming the inputs? How the application is transforming the data states to finally in the output? How are the middle layers processing the data? including some other questions about communication of different components. The aim of this section is to explain in detail the architecture of the application and flow of features. This will also explain the requirements to execute the application on local system.

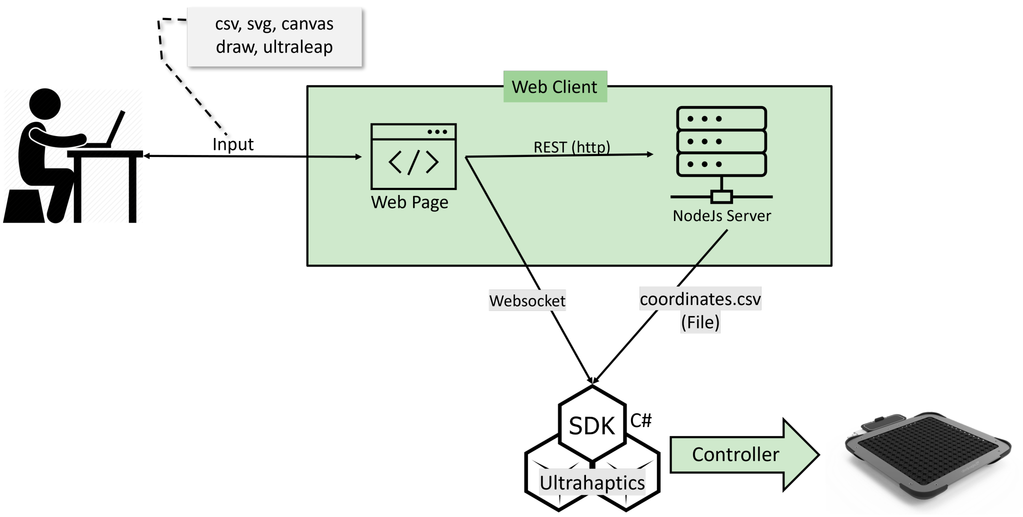

Fig. 4.1: Figure: The Haptic Printer (Application architecture)

As seen in the architectural diagram(Fig 4.1) there are multiple components of the application, from user input to ultrahaptics device to the communication through websocket and REST. The components are mentioned in detail below:

4.1 Frontend

The front end of the application is the web. The web application is built in vanilla-JS served over a nodeJs Server. However, for compiling js code for browser babeljs[Bab] is used.

The WebApp and Inputs

The webpage is responsible for taking multiple inputs from the user and start and stop Ultrahaptics device. The inputs are in form of the canvas draw, SVG icons, CSV file with (x, y) coordinates, Ultraleap inputs. The web page is responsible to scale and center the coordinates (according to x/y plane) taken from multiple input types to the Ultrahaptics dimensions and units.
In case of SVG some more processing is required. Since, there is no direct feature provided by the browser to extract the coordinates from an SVG. The code written for this is a custom implementation inspired from a spotify tool[Spo] which is under Apache license 2.0 . After extracting the coordinates from the given input the coordinates are scaled according to the output device and sent to the backend server in NodeJs, the processing in the nodeJs is explained in the section backend.
After, the successful response from the Node Server the front end is also responsible to communicate to the Ultrahaptics SDK implementation to start and stop the Ultrahaptics rendering.

4.2 Backend

The Backend is further divided into two parts. The First, is the Node Js Server. Second is, the system implementation in C-sharp, which is responsible to control and render shapes in Ultrahaptics device.

NodeJs Server

The functionality of nodeJS server is minimal. It hosts the WebApp at localhost:3000 in its root and also consumes the scaled coordinates from the WebApp. After getting the coordinates it creates a CSV file and writes all the coordinates in a csv as x,y as two columns. Post this functionality it responds to the webapp with a success message. This CSV file is later consumed by the SDK API implementation to render it on Ultrahaptics device.

The SDK extension: Ultrahaptics API's

Ultrahaptics SDK[Ult] has provided some of the classes to manipulate the control points, intensity and how the emitter will respond to the control points and positions provided to it. We built a system using these API's to continuously render custom points and shapes provided. All the previous implementations and tests we found were for a shapes which were pre defined in the code and it was a challenge to dynamically change the emitter points at a given time.

This part of the system implements two types of rendering Amplitude Modulation(AM) and Time Point Streaming(TPS). Both the implementations consumes dynamic inputs and detail explanation of how it is implemented can bee seen below in code documentation in Files AM.cs and TPS.cs. The choice of rendering type between AM and TPS can be selected from the WebApp.

The system also implements a websocket for communication which will be discussed further.

4.3 Communication

Websocket

The websocket communication happens between the web app and Csharp Ultrahaptics system to consume API's. Where Csharp implements the websocket server $(forreference, seefileUltrahapticsOrchestrator.cs)$ and webapp implements the websocket client $(refseecswebsocket.jsbelow)$. The webapp sends a message to backend system to start or stop the Ultrahaptics device rendering, using websocket. It also sends what type of rendering to use(AM or TPS). the websocket communication message is in JSON format and an example message can be seen below.


```
{
  "action":"start",
  "type":"AM"
}
```


Fig. 4.2: Figure: Json example to start AM rendering

REST

The REST communication in between the webapp and the NodeJs server. After the input processing in the webapp the coordinates are sent to the NodeJs usin this REST channel at $localhost : 3000$.

CSV file I/O

After the NodeJs server receives the coordinates, the coordinates are written to a csv file in the column format. Column names (x,y). These coordinates file is picked up

5 Validation

You can put your declaration here, to declare that you have completed your work solely and only with the help of the references you mentioned.

Bayreuth, September 27, 2021

The Haptic Printer

6 Conclusion

You can put your declaration here, to declare that you have completed your work solely and only with the help of the references you mentioned.

Bayreuth, September 27, 2021

The Haptic Printer

7 Future Work

You can put your declaration here, to declare that you have completed your work solely and only with the help of the references you mentioned.

Bayreuth, September 27, 2021

The Haptic Printer

Bibliography

[Bab] *Babel · The compiler for next generation JavaScript* (cit. on p. 14).

[Lea] *Controller — Leap Motion JavaScript SDK v3.2 Beta documentation* (cit. on p. 11).

[Spo] *GitHub - spotify/coordinator: A visual interface for turning an SVG into XY coördinates.* (Cit. on p. 14).

[Ult] *GitHub - spotify/coordinator: A visual interface for turning an SVG into XY coördinates.* (Cit. on p. 15).

[Wei+13] Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. „Analysis of the Accuracy and Robustness of the Leap Motion Controller“. In: *Sensors* 13.5 (2013), pp. 6380–6393 (cit. on p. 11).

List of Figures

List of Tables

Declaration

You can put your declaration here, to declare that you have completed your work solely and only with the help of the references you mentioned.

Bayreuth, September 27, 2021

The Haptic Printer