

User Manual

Tim Visée & Nathan Bakhuijzen

October 2018



1 Goal

The purpose of this manual is twofold:

- To inform research how to use the platform to execute experiments and conduct research
- To inform future developers how to continue developing the platform

2 Problem Definition

What physical motions are natural, effortless and easy, in order to control a computer or other digital device?

Questions of this nature can be answered using the *Can't Touch This* platform. *Can't Touch This* aims to be a platform for researchers that would like to conduct research in the field of touchless computer systems. We believe that our platform allows researchers to build a strong foundation for the future of touchless control. Giving researchers the opportunity to conduct research improves the chance for touchless control of computers only seen in futuristic movies and tv shows.

2.1 Motivation

At the start of the KB-80 minor, students were given a choice in the subject of the research. Mister Hani introduced us to a series of subjects, of which the LeapMotion project was the most interesting to us. The idea of the LeapMotion was to create or extend existing software to enable people to control a computer without touching any peripherals, like keyboards and mice.

2.2 Background information

Research in the field of touchless computer systems is motivated by the desire for these systems in sterile environments. For example, surgeons often make use of computer systems to aid them during their surgeries by providing crucial information such as CT, MRI and X-ray scans. This is where touchless computer systems come in. These systems allows surgeons on control a computer without the need for physical peripherals.

3 Installation Guide

3.1 Requirements

- A computer with the Windows (7+), OSX (10.7+, Lion+) or Linux (kernel 2.6.18+) operating system
- An installation of the LeapMotion SDK
- An installation of the Rust programming language
- The physical LeapMotion device itself

3.2 Software Dependencies

The *Can't Touch This* platform is written using the Rust programming language. This means that the operating system that the platform will run on must support the Rust language. Fortunately, Rust runs on all popular operating systems today, shown above in the list of requirements. An up-to-date list of all supported versions can be found on the Rust website. Additionally, the *Can't Touch This* platform requires the LeapMotion SDK to provide all necessary sensor data. Just like the Rust programming language, the LeapMotion SDK can be installed on all platforms.

3.3 External resources

To be written.

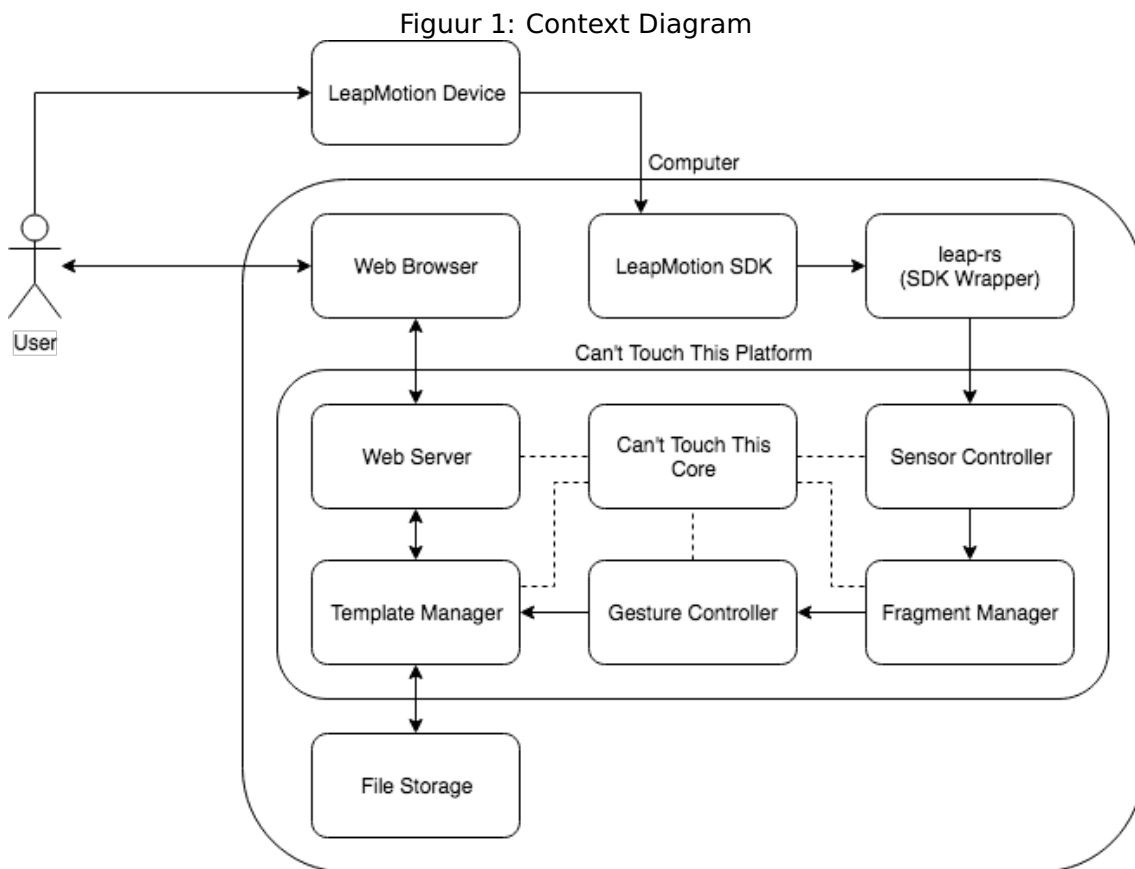
3.4 External development tools

To be written.

4 User Instructions

This chapter gives users instructions on how to use the *Can't Touch This* platform. It assumes that the user has followed the instructions found in the *Installation* chapter. The following instructions will detail how to setup the platform so that you can conduct the *experiments* found later in this manual.

5 Architecture Diagrams



In this diagram, the *Can't Touch This* platform displayed globally. It Shows three main objects:

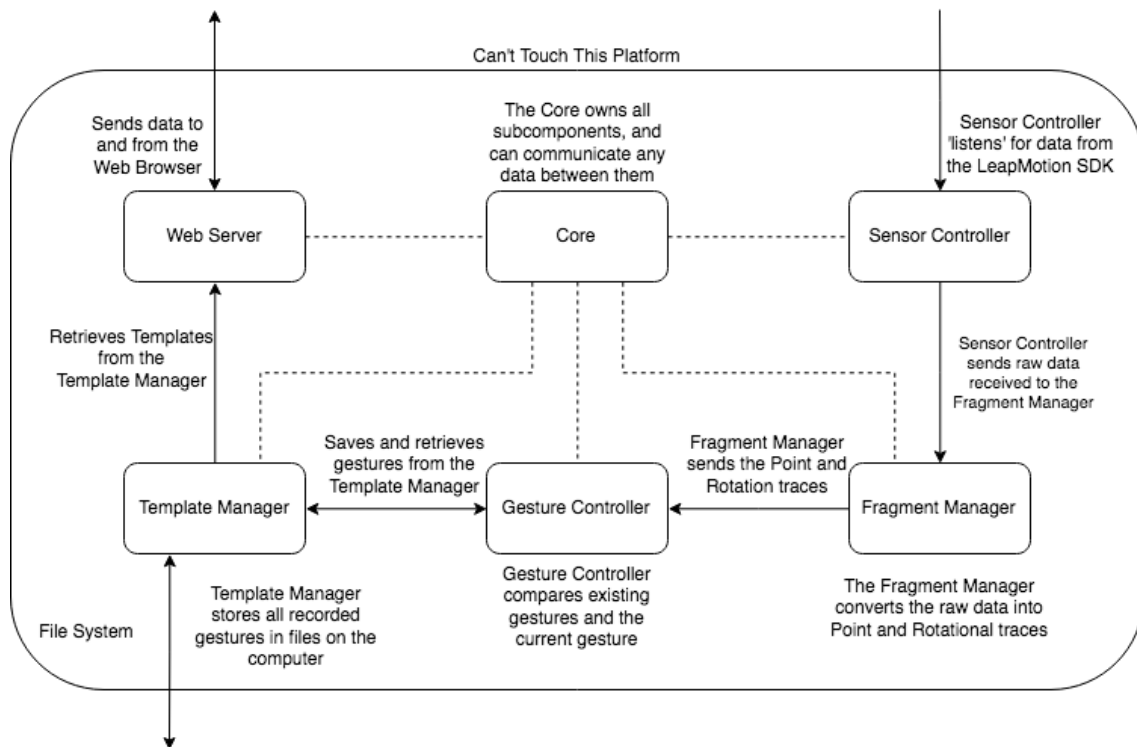
- The user
- The LeapMotion device
- The Can't Touch This platform

The user attaches the LeapMotion device to the computer, installs the platform and moves it's hand above the sensor to conduct an experiment. The LeapMotion device captures the data of the hand, and passes it on through the LeapMotion SDK, to the leap-rs wrapper. This leap-rs wrapper maps all functions made available through the LeapMotion SDK to the Rust programming language. Normally, a platform like *Can't Touch This* would have to be programmed using the C programming language, as this is the language the SDK is written in.

The leap-rs wrapper enables the SDK functionality in our platform, which we use in the Sensor Controller. The Sensor controller is our gateway of information, of our bits, bytes and coordinates of the hand scanned by the LeapMotion device. The Sensor controller then passes this data on to the Fragment Manager, which records all data and converts it into Points, Rotational Points (known as RotPoints,) and Traces of both kind. It even improves the recorded points in the trace through a method called sampling. Sampling is the act of ...

After this conversion, the Gesture Controller receives the transformed data and compares the existing gestures, stored in the Template Manager, with the current gesture.

Figuur 2: Context Diagram



6 Known Issues

- *Can't Touch This* may crash upon running the release version of the executable
- On macOS, the LeapMotion device may never give data to begin with
- On macOS, the LeapMotion device may stop recording data randomly
- On macOS, the application may not run well when minimalizing the backend application