Deployment Manual

Deploying pre-built project (Recommended)

Requirements

- Latest Firefox release
- **Optional:**
  - Any Python3 version
  - Microsoft Visual C++ Redistributible: https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170. Install this if the `MFC-UCL-MI3-Settings.exe` doesn't work

Running the project

1. Once you've extracted the main project zip file, there should be 3 folders: `code`, `website`, and `video`. Navigate to the `code` directory to see the full source code of the project
2. Inside the `code` directory you should see a folder called `release`, which should look like this:

3. In the `release` directory, click on the `MFC-UCL-MI3-Settings.exe` file to open up the main admin frontend/launcher. This frontend allows you to change core backend settings, and launch the MotionInput backend. You can also exit MotionInput from this admin frontend. For now, click on the Launch button to start MotionInput. This will start with the default settings, in which the controlling hand is the right hand. You can change any settings and press Launch again, which will relaunch MotionInput with the new settings

4. The next step is to install the WebExtension frontend to Firefox. Right click the `MotionInputWebExtension.xpi` and open it in Firefox. Firefox should then prompt you to add the extension, and you should press the add button to do so. The extension is permanently installed, and can be accessed via the Firefox sidebar.
5. Upon first installation, it should automatically open the sidebar, but if not, you can manually open it. If you don't know how to access the sidebar, you can follow the instructions at this link: https://support.mozilla.org/en-US/kb/use-firefox-sidebar-access-bookmarks-history-synced. Once the sidebar is open, you can click on the dropdown arrow on the sidebar titlebar, and switch to the `UCL MotionInput` extension.
6. You are done, the MotionInput backend as well as the WebExtension front-end is running. If you stuck with the default admin frontend settings, the controlling hand should be the right hand. Bring up your right hand to the camera and you should be able to see a preview of your hand in the Firefox sidebar, as well as the detection boundaries. Move your hand from the center to one of the boundaries to register a keyboard arrow press, and make a fist with your hand to register a keyboard enter press. You can use this with any website that you open in Firefox. Have a go at reading a book: https://www.gutenberg.org/files/98/old/2city12p.pdf. **Note**: If you click on the WebExtension sidebar to make any changes, click back on the webpage to re-focus on the page, so that keyboard presses act on the webpage, instead of on the sidebar
7. If you want a more fine-tuned experience, you can test out the project with a Hospital Kiosk Demo website that we built. **Note**: You need python for this to work. Open the main `release` folder, and double click on the `Start-Hospital-Demo.exe` file. This will open up a console window that will host the website. Keep it open if you want to access the website. Once the exe starts, you can type `localhost:8000` in the Firefox address bar to view the Hospital Kiosk Demo.

Stopping the project

1. To stop the MotionInput backend, you can open the `MFC-UCL-MI3-Settings.exe` admin frontend and click on `Exit MotionInput`
2. To stop the WebExtension frontend, you can simply close the sidebar.
3. If you started the Hospital Kiosk Demo, you can simply close the window that opens when you click the `Start-Hospital-Demo.exe`, and it should stop the website.

Notes

- The WebExtension uses Firefox specific APIs to generate the sidebar, so make sure you use Firefox.

- If you accidently close the WebExtension frontend sidebar, you can open it again by following step 5 in the `Running the project` section above

---

Building from source (Not recommended)

Note: It is not recommended to build from source due to the complexity involved. Use the instructions for the pre-built project above if you want to get the project up and running as quickly as possible

Requirements

- NodeJS 18.15.0 LTS
- Python 3.9.13
- Visual Studio 2022, with Desktop Development build tools and Windows 10/11 SDK installed

Building

1. Once you've extracted the main project zip file, there should be 3 folders: `code`, `website`, and `video`. Navigate to the `code` directory to see the full source code of the project
2. To start the build process, open up Powershell and `cd` to the `code` directory mentioned in the above step.
3. Then, type `python build_all.py` to start the build process for most of the components. Once this is complete, you should see a folder named `release` in the main `code` folder. **Note:** This will install all the python requirements to the main python install, which may not be ideal. If you want to instead use a virtual environment such as venv, make sure to activate it first in Powershell, and then type `python build_all.py` to install all the requirements to the virtual environment.
4. The next step is building the MFC frontend. In the main `code` folder you should see a folder called `src`. Navigate to this folder, and then navigate to the `MFC-UCL-MI3-Settings` folder. Inside this folder, you should see a file called `MFC-UCL-MI3-Settings.sln`. Double-click this file to open it in the Visual Studio IDE. Once the IDE opens, go the the top toolbar, click `Build > Build Solution`.
5. Navigate back to the main `code` folder and open the `release` folder. It should look like this:

6. You are done building. Now, just follow the steps in the **Deploying pre-built project** section at the top of the document