# MotionInput Deployment Manual

Michal Bravansky Shu Han Ho Dexin Fu Seojeong Hong

### **General Deployment**

The MotionInput application can be executed through a standalone executable file. First, start by cloning the repository:

git clone https://github.com/MotionInput/MI\_3.4.git

Make sure you have installed all dependencies:

```
pip install -r requirements.txt
```

Users are afforded the convenience of directly running the application via an .exe file. This executable can be generated by executing the following command within the main directory of the repository:

python build.py motioninput.py

Upon the execution of the generated .exe file, the MotionInput application initializes, adopting the configurations specified within the config.json file. This configuration file is designed to be user-editable, thereby allowing for customization according to user preferences.

To evaluate the functionalities of the MotionInput v3.4, users are encouraged to modify the config.json file to select from the available game modes.

#### Migrated Features Deployment

Users are adviced to evaluate the migrated features and functionalities by switching among the 20 models that have been successfully migrated to the current version. These models are listed in Table 1 below. Detailed descriptions and operational guidelines for each model can be found in the User Manual.

Game Mode	File Name
Zoomevent	Zoomevent.json
Samurai_swipe_event	Samurai_swipe_event.json
Gun_move_event	Gun_move_event.json
Head	Head.json
Mr_swipe_event	Mr_swipe_event.json
Nose_tracking_event	Nose_tracking_event.json
Force_field_event	Force_field_event.json
Nose_scroll_event	Nose_scroll_event.json
Spiderman_thwip	Spiderman_thwip.json
Pitch_click	Pitch_click.json
Head_trigger	Head_trigger.json
Nosebox_display	Nosebox_display.json
Face_display	Face_display.json
Display_element	Display_element.json
Circle_trigger	Circle_trigger.json
Body_points	Body_points.json
Sound_pose	Sound_pose.json
Mr_swipe	Mr_swipe.json
Brick_ball	Brick_ball.json
Gun_move	Gun_move.json

Table 1: Migrated Game Modes and Their Respected File Names

## Python Package Deployment

First, we need to clone the MotionInput repository:

```
git clone https://github.com/MotionInput/MI_3.4.git
```

Next, install the MotionInput package. This command checks for all dependencies before compiling the MotionInput repository into the data/api folder and installing that folder as a package:

pip install .

Afterwards, you can create a new Python .py file to interact with the MotionInput API through the motioninput\_api package. However, make sure to copy the data folder to the same directory before executing the following code:

```
from motioninput_api import MotionInputAPI
MotionInputAPI.start()
MotionInputAPI.run()
```

$\sim$ Test	🕏 test.py
$\checkmark$ data	1 from motioninput_api import MotionInputAPI
<ul> <li>data</li> <li>about</li> <li>api</li> <li>assets</li> <li>games</li> <li>gestures</li> <li>help</li> <li>logging</li> <li>ml_models</li> <li>modes</li> <li>poses</li> <li>speech_submodes</li> <li>yolo</li> <li>config.json</li> </ul>	<pre>1 from motioninput_api import MotionInputAPI 2 3 MotionInputAPI.start() 4 MotionInputAPI.run()</pre>
<ul><li>{} speakers.json</li><li>d test.py</li></ul>	

Figure 1: Example Setup for the MotionInput Python Package

## **DLL** Deployment

If you have access to the submitted .zip file, you are already equipped to begin with the MotionInput Dynamic Link Library. Everything you need to utilize it is present in the example folders.

To specifically run the DLL, first unzip DLL.zip. As this folder needs to contain everything a developer might need, it can take couple of minutes. Then copy and paste the data folder from the main DLL folder into the Release directory of any of the example applications, following the path:

example/app/MotionInputConsole/bin/Release/net8.0. There, you will need to include the data folder and start the provided console application.

In the event that you do not have access to the submitted materials, you can set up the DLL by adhering to the instructions outlined below.

#### 0.1 DLL Setup From Git Repository

This section outlines the procedure for setting up the DLL on your machine. Start by cloning the DLL repository using the command:

```
git clone https://github.com/MotionInput/MI_DLL.git
```

Next, obtain the embedded Python package, preferably version 3.12, from Python's official website<sup>1</sup>. After downloading, install it in the MI\_DLL directory.

Following the installation, open a terminal in the MI\_DLL folder and install all necessary packages with the command below:

<sup>&</sup>lt;sup>1</sup>https://www.python.org/downloads/release/python-3120/

#### python -m pip install -r paths/to/motioninput/requirements.txt

Should you encounter any issues with the gamepad driver, it is recommended to download the additional Windows extension<sup>2</sup>. In case of problems with the cv2 package, ensure to copy the python3.dll file into the cv2 directory located within lib/site-packages.

Furthermore, transfer the "data" folder from MotionInput into the directory "example/c#/MotionInputConsole/bin/Release/net8.0". Also, ensure to move the Python interpreter into:

"example/c#/MotionInputConsole/bin/Release/net8.0/data/python".

To conclude, you may either directly execute the MotionInputConsole.exe located in the Debug folder or open and build + run the project in Visual Studio.

Name	Туре	Date modified	Size
🚞 data	File folder	3/17/2024 12:55 AM	
MIDLL.dll	Application extens	3/20/2024 3:27 PM	14 KB
notioninput_api.pyd	Python Extension	3/18/2024 7:12 PM	3,032 KB
MotionInputConsole.dll	Application extens	3/20/2024 3:36 PM	6 KB
MotionInputConsole.exe	Application	3/20/2024 3:36 PM	140 KB
python311.dll	Application extens	12/4/2023 8:03 PM	5,651 KB

Figure 2: DLL and Console Application Directory Structure

<sup>2</sup>https://ds4-windows.com/download/vigembus-driver/

Name	Date modified	Туре	Size
DLLs	3/17/2024 12:53 AM	File folder	
Doc	3/17/2024 12:53 AM	File folder	
📒 include	3/17/2024 12:53 AM	File folder	
📒 Lib	3/17/2024 12:54 AM	File folder	
📒 libs	3/17/2024 12:54 AM	File folder	
Scripts	3/17/2024 12:54 AM	File folder	
share	3/17/2024 12:54 AM	File folder	
tcl	3/17/2024 12:54 AM	File folder	
Tools	3/17/2024 12:54 AM	File folder	
LICENSE.txt	12/4/2023 8:03 PM	Text Document	37 KB
NEWS.txt	12/4/2023 8:04 PM	Text Document	1,495 KB
ne python.exe	12/4/2023 8:03 PM	Application	101 KB
python3.dll	12/4/2023 8:03 PM	Application extens	66 KB
bython311.dll	12/4/2023 8:03 PM	Application extens	5,651 KB
📄 pythonw.exe	12/4/2023 8:03 PM	Application	100 KB
vcruntime140.dll	12/4/2023 8:03 PM	Application extens	117 KB
vcruntime140_1.dll	12/4/2023 8:03 PM	Application extens	49 KB

Figure 3: Embedded Python Directory Structure