

MIREVI MotionHub

Hochschule Düsseldorf
University of Applied Sciences



Fachbereich Medien
Faculty of Media



SPONSORED BY THE



Federal Ministry
of Education
and Research

MIREVI MotionHub (MMH) is a middleware for merging body tracking data from different systems into one coordinate space in real-time in order to combine and use their individual benefits.

MMH offers support for several body tracking systems and encompasses a game engine plug-in that connects the MMH with Unity by means of a standardized protocol. The plug-in allows for the usage of a single type of skeleton for any body tracking system and, therefore, facilitates the switch between different body tracking systems during app development significantly.

MotionHub is developed at the research lab **MIREVI** from the [University of Applied Sciences Düsseldorf](#) within the scope of the project [HIVE](#).

Acknowledgements

The project [HIVE](#) is sponsored by the [German Federal Ministry of Education and Research](#) (BMBF) under the project number **16SV8182**.

Requirements

Currently MMH is only supported on Microsoft Windows operating systems.

- This version of MotionHub was tested on Microsoft **Windows 10 64 bit**.

Minimum Computer Requirements

- Seventh Gen Intel i5 Processor
- NVidia GeForce GTX 1070
- 4 GB Memory
- 4 GB Storage

Documentation

The developer, API and user documentation including a class collaboration diagram can be found in the `doc` folder.

Supported Systems

MMH currently supports the listed body tracking systems.

Supported Systems

Azure Kinect

Supported Systems

OptiTrack

Setup and Building

The [CMake](#) system is used to generate project files and for downloading all required dependencies. Please use the `CMakeLists.txt` file for generating.

- MMH is developed with Microsoft **Visual Studio 2017**. (*CMake has only been tested with this IDE version.*)
1. Download or clone MMH
 2. Download CMake GUI and start it
 3. In CMake, set source to the MMH path
 4. Set binaries to (MMH path)/build
 5. Click configure. This will take a few minutes so go on with 6 and 7.
 6. Download [Azure Kinect Sensor SDK](#)
 7. Download [Azure Kinect Body Tracking SDK](#)
 8. Meanwhile, the configuration process in CMake should have failed.
 9. Copy the installed Azure Kinect Folders to MMH/deps. You can delete the "tools"-Folder in both, they are very big.
 10. In CMake, set the flags "advanced" and "grouped", you should see all dependencies listed
 11. Under K4A set the paths to the include (something like this: MMH/deps/Azure Kinect SDK v1.4.0/sdk/include) and lib folder (MMH/deps/Azure Kinect SDK v1.4.0/sdk/windows-desktop/amd64/release/lib)
 12. Repeat for K4ABT (MMH/deps/AzureKinectBodyTrackingSDKv1.0.1/sdk/include) and (MMH/deps/AzureKinectBodyTrackingSDKv1.0.1/sdk/windows-desktop/amd64/release/lib)
 13. Again, click "Configure" (This should be successful), then click "Generate".
 14. You can now click "Open Project" or open the Solution with visual Studio (MMH/build/MireviMotionHub.sln)
 15. In the VS Solution Explorer, right click on the project "MireviMotionHub"->"Set as Startup Project" and "MireviMotionHub"->"build"
 16. After the code is compiled, you can push F5 in VS or start MMH/build/bin/MireviMotionHub.exe to start the MotionHub

Please Note that we use Qt Framework for the UI. To build the project you need the [Qt Visual Studio Tools](#) and for editing Qt `.ui` files you need [Qt Designer](#).

Authors

- [Philipp Ladwig](#) - Project Manager and Developer
- [Kester Evers](#) - Developer
- [Eric Jansen](#) - Developer

- [Manuel Zohlen](#) - Developer

Build with

- [Qt](#)
- [Eigen](#)
- [OSC Pack](#)
- [TinyXML](#)

License

Pending