MIREVI MotionHub

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

This project is sponsored by: German Federal Ministry of Education and Research (BMBF) under the project numbers **16SV8182** and **13FH022IX6**. Project names: HIVE-Lab (Health Immersive Virtual Environment Lab) and Interactive body-near production technology 4.0 (german: 'Interaktive körpernahe Produktionstechnik 4.0' (iKPT4.0)).

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
- 5 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** and **2019**. *(CMake has only been tested with these IDE versions.)*

1. Download or clone MMH
2. Download Azure Kinect Sensor and Body Tracking SDK
3. In CMake, set source to the MMH path
4. Set binaries to (MMH path)/build
5. Click "configure" (click Yes, select your installed VS version and Finish). This will take a few minutes (and is expected to FAIL) so go on with 5 and 6.
6. Download Azure Kinect Sensor SDK
7. Download Azure Kinect Body Tracking SDK
8. Install both and remember the install paths!
9. Meanwhile, the configuration process in CMake should have failed, because it couldn't find Azure Kinect dependencies.
10. In CMake, set the flags "advanced" and "grouped", you should see all dependencies listed, including K4A (Sensor SDK) and K4ABT (Body Tracking SDK)
11. Under K4A set K4A_INCLUDE_DIR to the include folders path (something like this in your freshly installed directory: /Azure Kinect SDK v1.4.0/sdk/include)
12. Repeat for K4A_LIBRARIES_DIR (/Azure Kinect SDK v1.4.0/sdk/windows-desktop/amd64/release/lib).
13. Repeat both for K4ABT (/AzureKinectBodyTrackingSDKv1.0.1/sdk/include) and (/AzureKinectBodyTrackingSDKv1.0.1/sdk/windows-desktop/amd64/release/lib)
14. Again, click "Configure" (This should be successful), then click "Generate".
15. You can now click "Open Project" or open the Solution with VS (/MMH/build/MireviMotionHub.sln)
16. In the VS Solution Explorer, right click on the project "MireviMotionHub"->"Set as StartUp Project" and "MireviMotionHub"->"build"
17. After the code is compiled, you can push F5 (Local Windows Debugger) in VS or execute /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