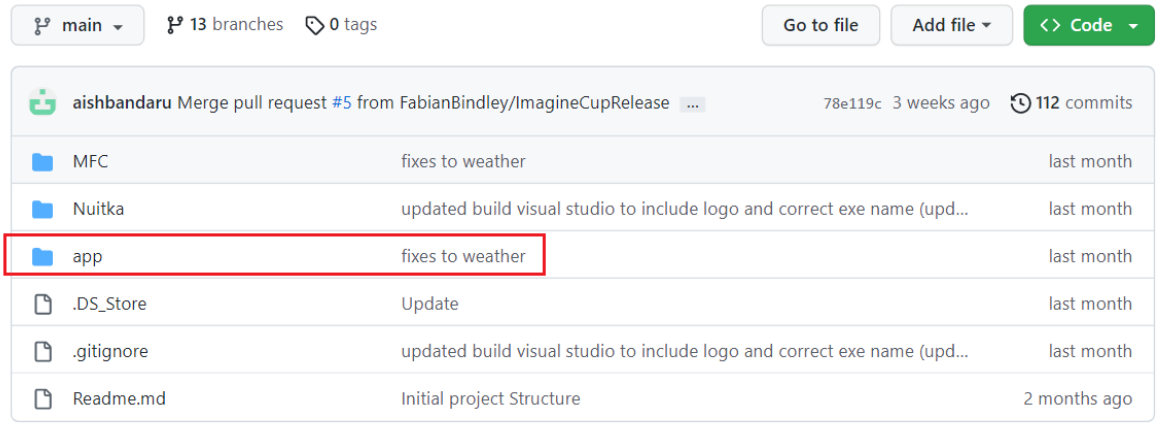
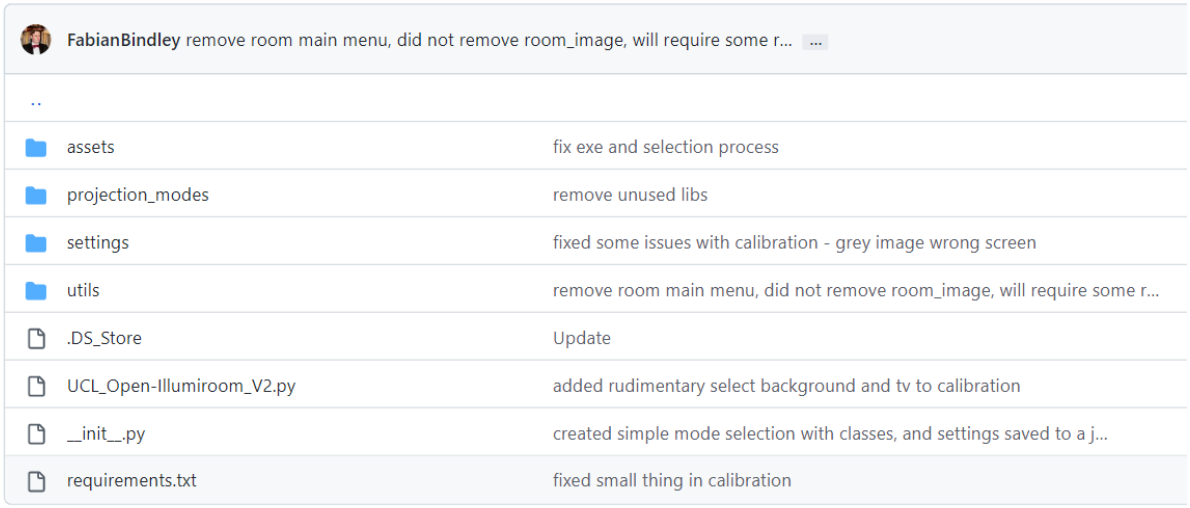
System Development and Compilation

Original Author: Fabian Bindley

The UCL Open-Illumiroom V2 system source code is contained within the app folder in the root directory of the project's repository.



The *main* file for the app is the UCL_Open-Illumiroom_V2.py file, it contains the app's initialisation functions, and main loop. Projection mode classes, util classes, assets and settings are separated into the 4 folders.



The required python libraries are included in the requirements.txt

Compilation

To compile UCL Open-Illumiroom V2, the Nuitka library is used. Nuitka compiles Python source-code to C source code, on a platform dependent basis.

To compile UCL Open-Illumiroom V2 to a standalone file that can be distributed, run the following bash command:

- `python -m nuitka --standalone --enable-plugin=tk-inter  --enable-plugin=pyside2  --output-dir=release --remove-output --disable-console  app/UCL_Open-Illumiroom_V2.py; cp -r app/settings release/UCL_Open-Illumiroom_V2.dist; cp -r app/assets release/UCL_Open-Illumiroom_V2.dist`

This will create a release folder in the root of the directory and place a UCL_Open-Illumiroom_V2.dist folder there with the compiled build, and UCL_Open-Illumiroom_V2.exe.

The assets and settings folders are also copied over to the `.dist` folder.

To compile a build with a console output for debugging, you may run the following bash command:

- `python -m nuitka --standalone --enable-plugin=tk-inter  --enable-plugin=pyside2  --output-dir=release --remove-output  app/UCL_Open-Illumiroom_V2.py; cp -r app/settings release/UCL_Open-Illumiroom_V2.dist; cp -r app/assets release/UCL_Open-Illumiroom_V2.dist`

The difference between the 2 commands, is that the –disable-console flag is not present in the debugging command, to allow the console to show.

Additional Notes on compilation:

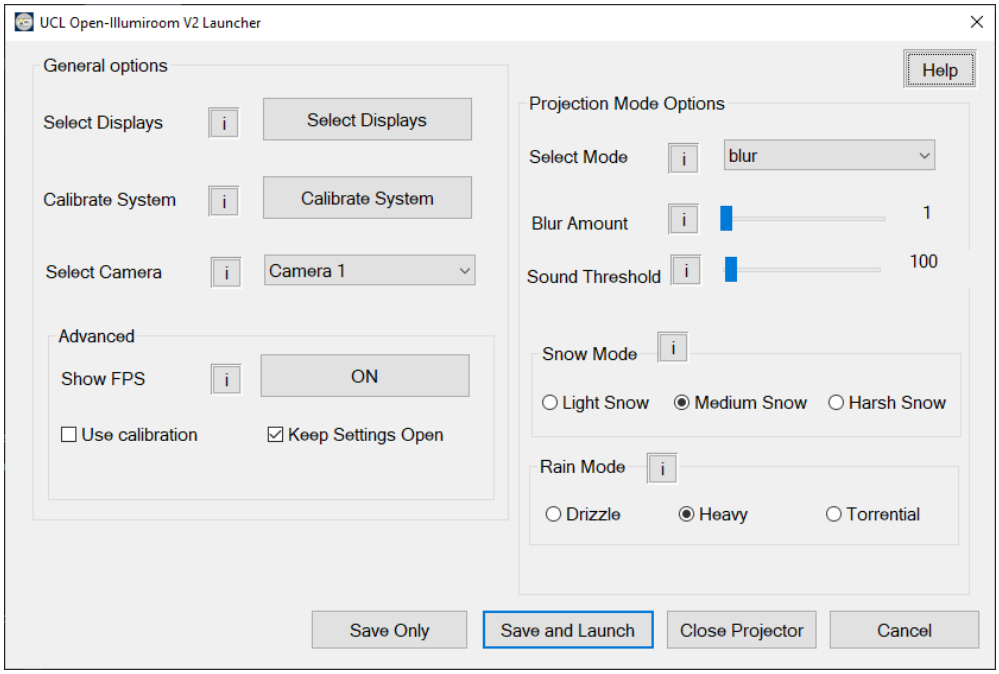
- Currently, the wobble mode and Audio capture containing the machine learning based audio detection cannot be compiled due to Nuitka not supporting the library Librosa. As a result, an older version of wobble is used that does not use the trained machine learning model for detecting sound. Instead, loud sounds trigger the wobble effect.
- The Nuitka folder in the application contains more information about compilation, as well as these 2 commands.

MFC Development and Compilation

MFC C++ Settings application designed for Facial Navigation on Windows.

Original Author: Anelia Gaydardzhieva

Adapted for UCL Open-Illumiroom V2: Fabian Bindley



To Run

Double click on the `UCL Open-Illumiroom V2 Launcher.exe` file

Structure

The structure of the MFC section mimics integration with MotionInput v3.2 (MI_v3.2), with only the essentials for the MFC application files and folders copied from MI_v3.2 This allows to run the MFC application in isolation and ensure intentional performance when integrated with MI_v3.2

- The `UCL_Open-Illumiroom_V2.dist/settings` folder stores `general_settings.json` and `mode_settings.json`. These are the settings files for UCL Open-Illumiroom V2.
- `packages` folder contains `nlohmann.json.3.10.5` which is the JSON library used to read and write into the JSON files.
- The `UCL_Open_Illumiroom_V2.dist` folder contains the compiled code to run the system, including `UCL_Open-Illumiroom_V2.exe,` the main application executable.

Setup

1. In case you have not done so already, install Visual Studio (preferably v2022)
2. Open Visual Studio Installer
3. Select `Modify`. If you have more than one version of VS installed, select `Modify` on the version you wish to access the code from.
4. Select `Individual Components`
5. Find the box with the latest C++ MFC (e.g. `C++ MFC for latest v142 build tools (x86 & x64)`)
6. If the box is not marked with a tick, select it and save
7. To modify the MFC code simply open *`YourGithubRepo`*`\MFC\data\UCL Open-Illumiroom V2/MFC-UCL-MI3-Settings.sln file.`

The MFC is called MFC-UCL-MI3 in places because this was the template used to create the UCL Open-Illumiroom V2 MFC. If possible, the naming should be updated in future versions, and old unused code and resources removed.

View/Modify MFC

Steps in VS:

```
View -> Resource View -> Dialog
```

Manual file navigation:

```
MFC-UCL-MI3-Settings.sln -> MFCUCLMI3Settings.rc -> Dialog
```

Access main file

```
MFC-UCL-MI3-Settings.sln -> MFC-UCL-MI3-SettingsDlg.cpp
```

Compile

1. Open `MFC-UCL-MI3-Settings.sln`
2. Set Solution Configurations to Debug or Release
3. Set option x64 or x86
4. Build Solution (Click green run icon on top bar)
5. The compiled `.exe` file is in root directory

Use the MFC .exe

1. Copy `.exe` file from its position in the repo
2. Paste `.exe` in the same directory as UCL Open-Illumiroom V2.dist