MoonSafe: Collision Avoidance System for Lunar Construction Sites

STRL Team

User Manual

---

1. System Overview

MoonSafe consists of two independent programs that work together:

- MoonSafe AutoBrake (*MoonSafe_AutoBrake.py*) : runs inside UC-win/Road through its Python API / SDK. It reads the vehicle's and nearby objects' position, velocity, and heading in real time, predicts collision risk, and directly controls the vehicle's throttle and brake.
- MoonSafe Radar UI (*MoonSafe_UI.py*) – This is a standalone desktop application. It visualizes nearby objects and risk level on a radar display, and gives the driver visual and audial (Text-to-Speech) warnings.

---

2. Development environment configuration

To use this system, the following programs and libraries must be installed on your computer.

- **UC-win/Road 19.0.1:** Driving simulation program
- **UC-win/Road SDK package:** Python API Plugin, UCwinRoadCOM, UCwinRoadUtils, LoggerProxy, CallbackHandlers
- **VisualStudioCode(VS Code):** Code editor
- **Python:** Programming language (version 3.14.2 or higher recommended)
- **Python Libraries:** Open a terminal (CMD) and enter the following commands to install.

2-1. Python environment setup for MoonSafe AutoBrake

UC-win/Road's Python API only supports Python 3.7 - 3.11.

- When AutoBrake is started from the Python API tab, UC-win/Road runs it using this system-installed Python underneath which is not a version bundled inside UC-win/Road itself.

- If a newer or otherwise incompatible Python version is already the system default, install a compatible version alongside it.

Use the code beneath in the code editor.

```
py install 3.11

py -3.11 -m pip install pywin32
```

If *py install 3.11* is not recognized, the machine has an older version of the py launcher that does not support this subcommand. In that case, download Python 3.11 directly from python.org instead, making sure to check *Add python.exe to PATH* during installation, then run the pywin32 install command above.

2-2. Python environment for MoonSafe Radar UI

UC-win/Road's Python API only supports Python 3.7 - 3.11.

- When AutoBrake is started from the Python API tab, UC-win/Road runs it
- The Radar UI is a separate, standalone program and does not go through UC-win/Road's SDK, so the Python 3.7–11 restriction in Section 2.2 does not apply here. Whatever Python 3.x is already the default on the machine — including newer versions such as 3.13 or 3.14 — is fine. There is no need to install or switch to Python 3.11 for this part.

Use the code beneath in the code editor.

```
py -m pip install PyQt5 pyttsx3
```

2-3 Visual Studio Code setup for the MoonSafe Radar UI

Installing a package with pip does not guarantee that VS Code will use it when running or analyzing *MoonSafe_UI.py*, since VS Code selects its own Python interpreter independently of the terminal. If PyQt5 or pyttsx3 still show as unresolved in VS Code after running the install command, the following steps confirm that VS Code is pointed at the same Python installation the command was run against.

General note: *MoonSafe_UI.py* specifically requires PyQt5. PyQt6 is a separate, incompatible package, and installing it would not resolve an unresolved-import error for PyQt5.

1. Install the Python extension (Microsoft) in VS Code, if not already installed.
2. Open the folder containing *MoonSafe_UI.py* in VS Code.
3. Press Ctrl+Shift+P, run “Python: Select Interpreter,” and choose the same Python installation where the command above was run.
4. If packages still show as unresolved, open VS Code's own integrated terminal (View → Terminal) and run the install command there, to guarantee it installs into the interpreter VS Code is actually using.

2-4 English TTS(text-to-speech) voice setup (Windows)

English TTS is required for the spoken alerts to be pronounced correctly. Without an English voice installed, Windows will read the English alert text using the system's default voice, which mispronounces it.

1. Open Settings (Win + I).
2. Go to Time & Language → Language & region.
3. Under “Preferred languages”, click “Add a language” and select English (United States).
4. In the install options, make sure “Text-to-speech” is checked, then click Install.

2-5 AutoBrake.py file placement

Copy the following files into UC-win/Road's Python API script folder. The following files come from the UC-win/Road SDK package which can be found on link below:

```
C:\UCwinRoad_SDK 19.0.1\Delphi\Plugins
```

In this link, copy the following files below:

- UCwinRoadCOM.py
- UCwinRoadUtils.py
- LoggerProxy.py
- CallbackHandlers.py

Then paste the copied files in the link below:

```
C:\UCwinRoad Data 19.0\Plugins\PythonAPIPlugin
```

In this folder, copy and paste MoonSafe_AutoBrake.py to the folder. All five files must be located in this same folder. MoonSafe_AutoBrake.py imports the other four directly.

2-6 Radar UI files

Keep the files below together in one folder (for example, the folder you open in VS Code):

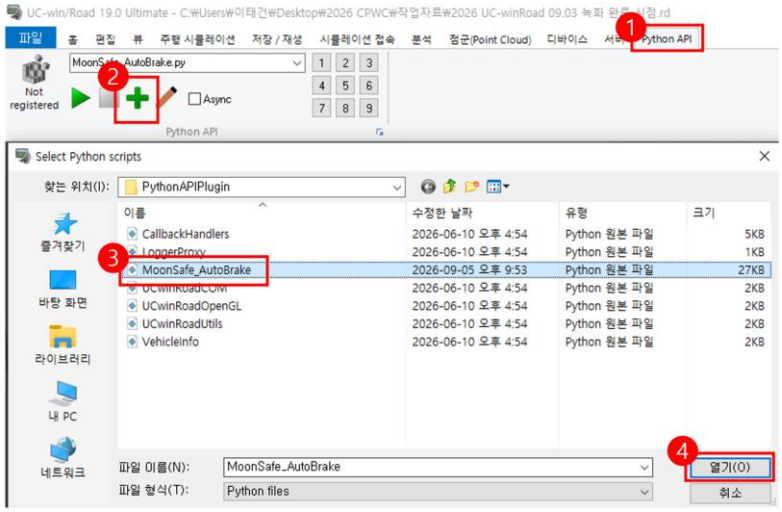
- *MoonSafe_UI.py*
- KoPubDotumMedium.ttf – optional custom font; if missing, the UI falls back to a default font automatically.

---

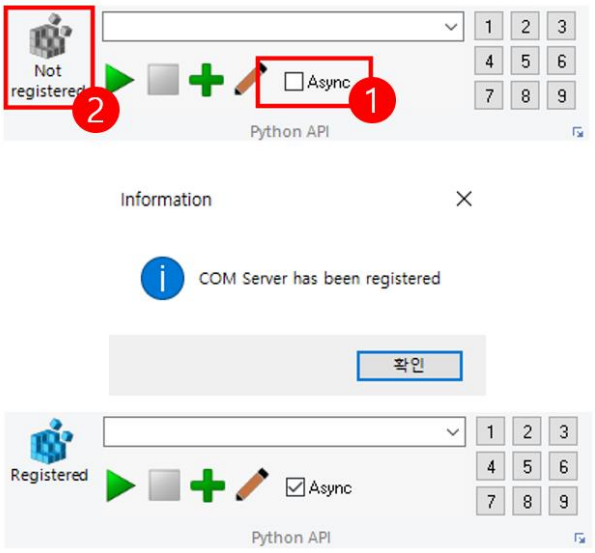
3. UC-win/Road settings

3-1 The settings required to run AutoBrake PythonAPIPlugin

To utilize the *MoonSafe_AutoBrake.py,* the following steps are required.



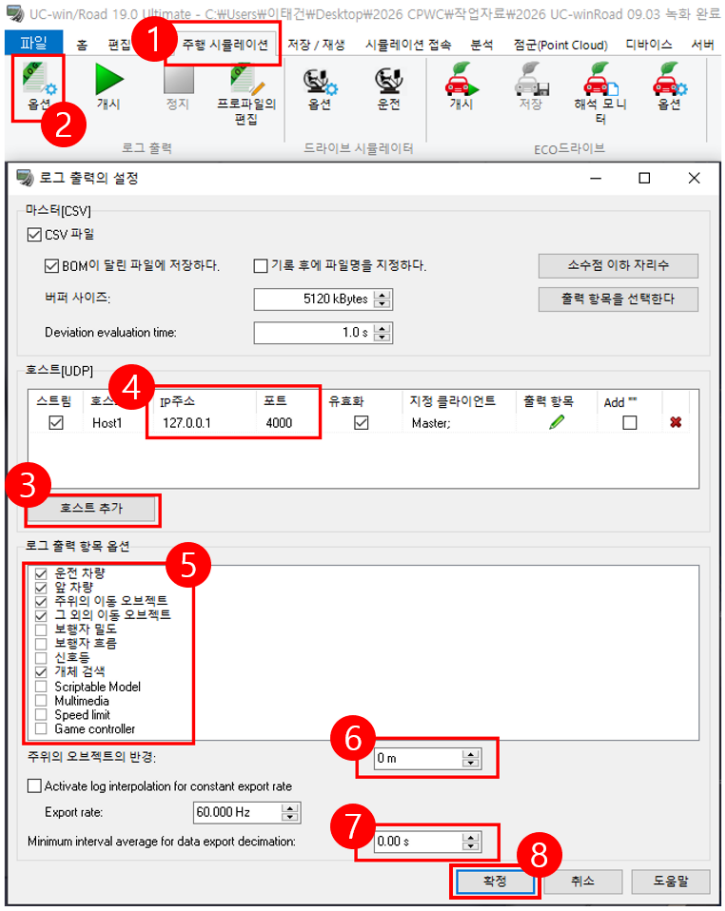
1. In UC-win/Road, open the Python API ribbon tab, click “+” (Add New Script), and select *MoonSafe_AutoBrake.py* from the PythonAPIPlugin folder, then click Open. This adds it to the script dropdown list.



2. Click ‘Async’ and the registration icon, which initially shows “Not registered.” A dialog confirming “COM Server has been registered” will appear; click OK. The icon will then show “Registered.”

3-2 UC-win/Road settings for the Radar UI

To allow the *MoonSafe_UI.py* to receive data, the following configuration settings is necessary in UC-win/Road.



1. Open the Driving Simulation tab, then click Options in the Log Output group.
2. In the Log Output Settings dialog, under Host [UDP], click “Add Host.”
3. Set the IP address to 127.0.0.1 and the port to 4000.
4. Under Log Output Item Options, enable at least: Driving Vehicle, Vehicle Ahead, Surrounding Moving Objects, Other Moving Objects, and Object Search.
5. Set “Radius of Surrounding Objects” and “Minimum interval average for data export decimation” to 0 m and 0.00 s respectively
6. Click Confirm.

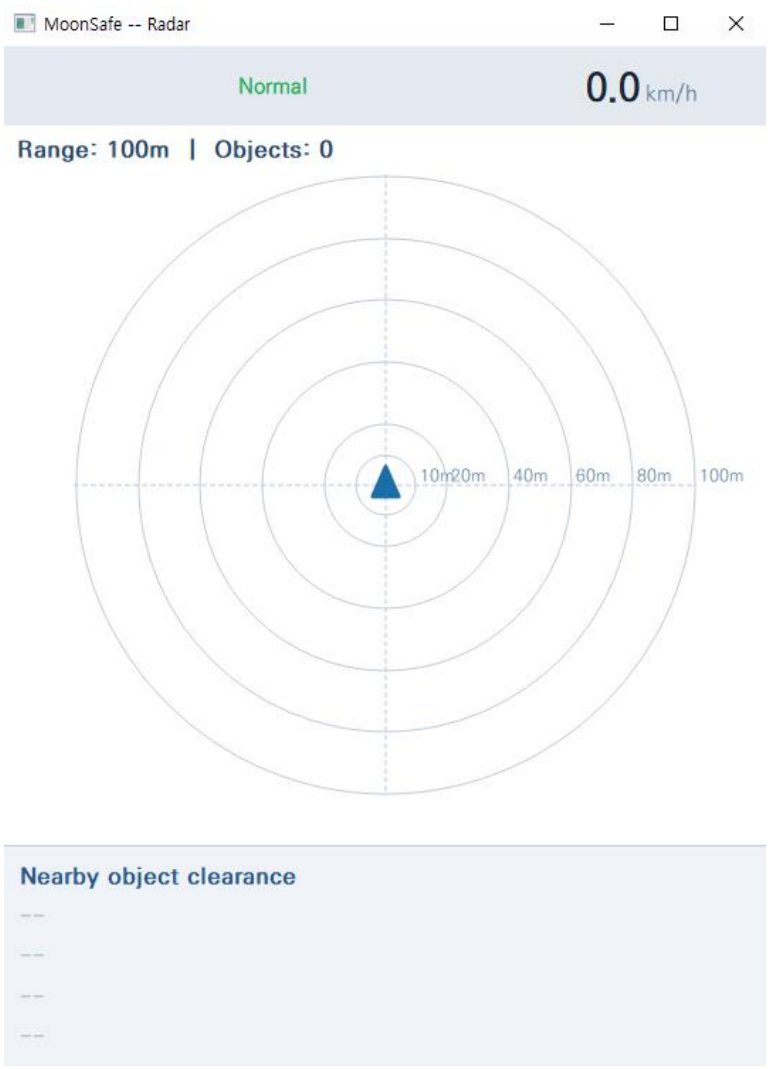
---

4. Running the MoonSafe Radar UI

1. Save the source code on your computer as *MoonSafe_UI.py*
   (The source code for MoonSafe_AutoBrake is inside UC-win/Road)
2. Open a terminal (CMD) and use the cd command to navigate to the folder where you saved the code.
3. Enter the command below to run the Python Radar UI program.
4. Then paste the copied files in the link below:

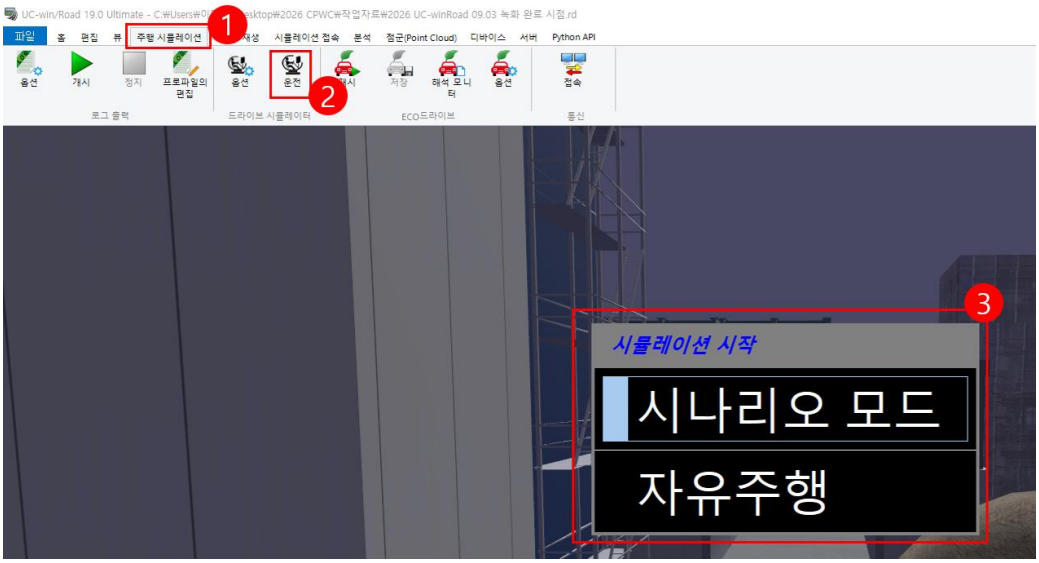
```
python MoonSafe_UI.py
```

5. If it runs successfully, a GUI window titled "MoonSafe – Radar" will appear as shown below .

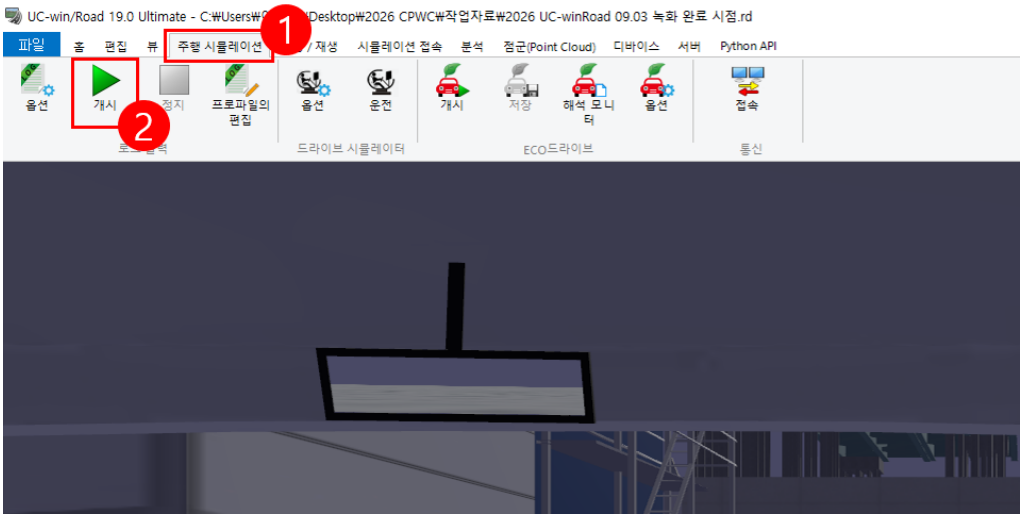


5. Running the UC-win/Road Simulation

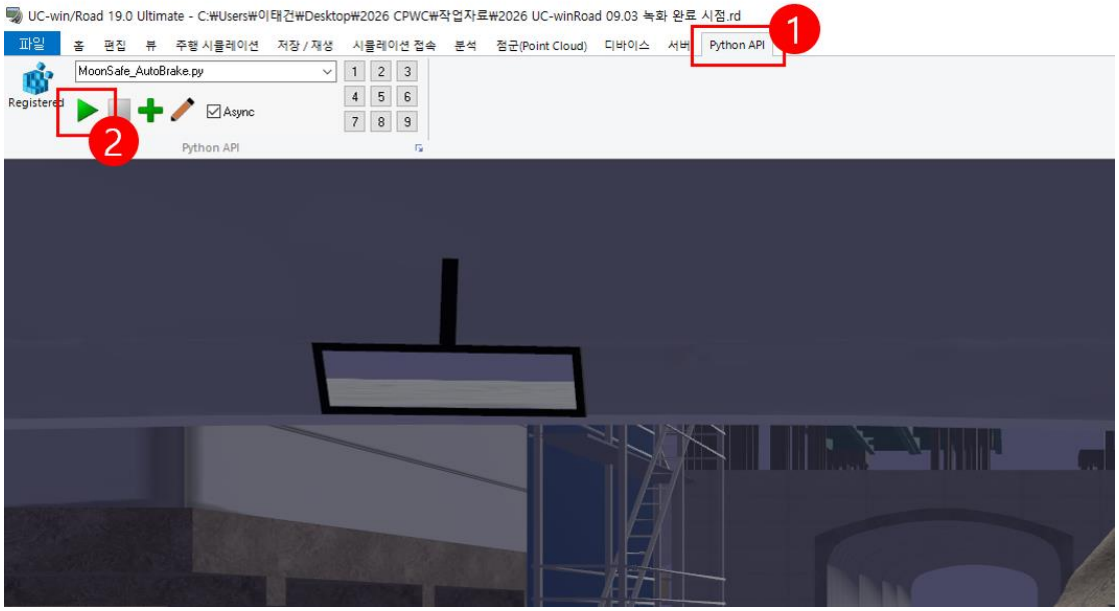
1. launch the UC-win/Road program using “Run as administrator”.
2. From the top tab in UC-win/Road, click to start the 'Driving Simulation'.
3. Choose whether to execute scenario mode or free driving mode



4. Before driving, start the Log Output to send log data to MoonSafe UI.



5. Next, start the PythonAPIPlugin to activate MoonSafe Autobrake



6. As you drive, the objects nearby will be detected via UI MoonSafe radar. When the object is near, the UI will display visual and audial warning. If the collision is imminent, the MoonSafe Autobrake will stop the vehicle.

