Install Instructions for Team 21 Project 3 Code

These are the install instructions specifically for a 64-bit Windows machine.

Installing Geany and MinGW

First you need to install the compiler that will allow the computer to be able to understand and run the code. If you already have a C compiler installed, it may not work with this code, so it is best to follow the instructions below.

1. Open your browser.
2. Go to <https://www.geany.org/>.
3. Click ‘Download Geany 1.36’. If this page doesn’t have a button saying this, go to <https://download.geany.org/> instead, and scroll to find ‘geany-1.36\_setup.exe’ and ‘geany-1.36\_setup.exe.sig’, and click both of these to download them.
4. Go to <https://sourceforge.net/projects/mingw-w64/files/Toolchains%20targetting%20Win64/Personal%20Builds/mingw-builds/7.3.0/threads-posix/seh/x86_64-7.3.0-release-posix-seh-rt_v5-rev0.7z/download>.
5. This website will automatically download the latest version of MinGW as a .7z file.
6. Go to the Windows store and download ‘BreeZip: Rar, Zip & 7z Extractor’ (or a 7z extractor equivalent that you trust).
7. Right-click the .7z file you downloaded in step 4 (it will be in the Downloads folder of your directory) and go to Open With > BreeZip (or equivalent) and click. This will open the application (close any ads that pop up as possible).
8. Click ‘Extract’ at the top (if you are using BreeZip, otherwise find an equivalent button to extract the files) and specify where you would like it to be extracted to. Just set this to ‘C:\\MinGW’ (put it in its own folder called MinGW in the C: drive).
9. Go inside the ‘MinGW’ folder (or if you didn’t rename it it will be called ‘x86\_64-7.3’) and go into the ‘mingw64’ folder. Here, select all the contents, right-click and select ‘Cut’.
10. Return to the main ‘MinGW’ folder (if you haven’t renamed it yet, you should), right click and select ‘Paste’. Then delete the now empty ‘mingw64’ folder.
11. Go to the search bar in Windows (bottom left corner of your screen, the magnifying glass icon), and type ‘path’. Click ‘Edit the system environment variables’. This will bring up a new window.
12. Click ‘Environment Variables…’ in the bottom right hand corner of the new window. In the top box, select the ‘Path’ variable and click ‘Edit…”.
13. In this new window, select ‘New’, and paste in the path to your MinGW bin folder, which should be ‘C:\MinGW\bin’ if you’ve been following along correctly.

Installing the SFML Library

Next, you need to download the library that our code works with. This means that the compiler will recognise new commands (from this library) that it doesn’t currently have installed.

1. Go to <https://www.sfml-dev.org/download/sfml/2.5.1/>.
2. Download ‘GCC 7.3.0 MinGW (SEH) – 64-bit’ from this website.
3. Once it has finished downloading, unzip the file by right-clicking on it in the ‘Downloads’ folder of your file system and clicking ‘Extract All’. This will bring up a window asking you where to extract the files to. Put it into a new folder. This is where we will also put the code.
4. Go inside the ‘SFML-2.5.1’ folder and select all of the contents.
5. Right-click and select cut.
6. Go into the main ‘SFML’ folder, right-click and select ‘Paste’.
7. Delete the ‘SFML-2.5.1’ folder (it should be empty now). Do this by right-clicking on it and selecting

Downloading the Files

Now that you have the compiler set up, you need to have the files to open in it.

1. Download the AVC\_robot.zip and AVC\_server.zip files from the Team 21 GitHub repository (<https://github.com/AmyBroeders/ENGR101Project3>).
2. Right click AVC\_robot.zip in your Downloads folder and select Extract All, and extract it to the folder where you put the SFML library. Repeat for the AVC\_server.zip file.

Setting up the makefile Files

Next we need to get the computer to understand where to find the files it needs to make the project.

1. Go into the AVC\_robot folder and right-click the file called makefile. Select ‘Open with Geany’.
2. Go into your file explorer and find the ‘include’ folder within the SFML folder.
3. Right-click it and select ‘Properties’.
4. Click and drag over the entire Location and press Ctrl + C to copy it.
5. Where it says ‘C:\\SFML\include’ on the makefile in geany, replace it with the location you just copied (select it and press Ctrl + V).
6. Paste the location again into the section which currently reads ‘C:\\SFML\lib’ and replace the ‘include’ on the end of the pasted location with ‘lib’
7. Close the makefile.
8. Go into the AVC\_server folder and find the makefile for that folder. Select ‘Open with Geany’.
9. Replace the path ‘C:\\SFML’ with the location again, taking off the ‘\include’ on the end entirely.

Compiling/Building

Here we specify how the computer should translate the code into stuff it can understand and run.

1. Right click on the robot.cpp file in the AVC\_robot folder and select ‘Open with Geany’.
2. Click the arrow next to the build button in the menu at the top.
3. Select ‘Set Build Commands’ from the drop down list.
4. Under ‘Independent commands’ in the ‘Command’ column and the ‘Make’ row, write ‘mingw32-make’.
5. Again under ‘Independent commands’ in the ‘Command’ column and the ‘Make Object’ row this time, write ‘mingw32-make %e.o’.
6. Click ‘OK’ in the bottom right corner of the window.
7. Click the arrow next to the build button.
8. Select ‘Make All’ from the drop down list.
9. The compiler should say ‘Compilation finished successfully.’ At the bottom of the Geany window.
10. Right click on the server3.cpp file in the ‘AVC\_server’ folder and select ‘Open with Geany’.
11. Repeat steps 7 and 8 for the server.cpp file.

Running

Finally, we have to deal with the server and client setup to make the application run with both components (the maze and the robot).

1. Close Geany and open the directory, going into the ‘AVC\_server’ folder.
2. Double click on the ‘server3.exe’ application (it may just say ‘server3’, but check the file type says Application).
3. A window should start up with the graphics pane, and a terminal with the bottom line reading ‘Listening…’.
4. Go into the ‘AVC\_robot’ folder.
5. Double click on the ‘robot.exe’ application (as above, it probably just says robot but check the file type).
6. Reopen the graphics pane, the program should be running.
7. To switch between mazes for the robot to run, open the ‘AVC\_server’ folder, find the ‘config.txt’ file, and edit the first line of it (in a text editor, or Geany will do) where it says ‘core.txt’ to say either ‘completion.txt’ or ‘challenge.txt’, or revert it to ‘core.txt’ from there as you like. The robot is set to run whichever algorithm you want by asking you in the command line, but it won’t work as well if it is mismatched to the maze, so you should change this to run it properly.