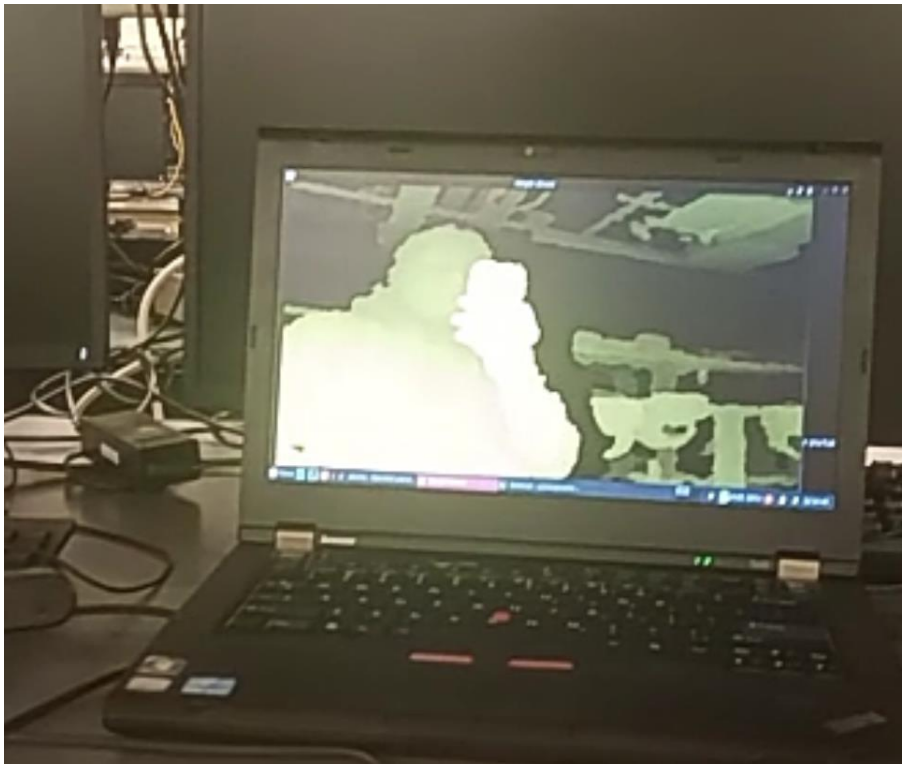
Week 3:

Hours: 14

This week I put together a skeleton program for the graphical portions of the project's software to be built around. My time was mostly spent linking in one library at a time to make sure the development environment is stable and well-organized for the rest of the semester. I picked up a brand-new tool, a build system called SCons, which will be used to manage the build process for this part of the project.



As of right now the program can display depth images captured from the camera. In addition, Jack wrote a program on top of my skeleton program which takes in test images and performs blob detection. The build process is greatly simplified by SCons, which is a tool that I had not heard of or used before this week. With SCons, the equivalent of the Makefile is actually a python script, shown below in its entirety:

```
#fbtest.cpp SConstruct file

env = Environment(CCFLAGS='-fpermissive -I/home/rachel/git/spectator/inc/OpenNI2/Include -I/usr/local/include/opencv4',
        LIBS=['OpenNI2', 'opencv_core', 'opencv_features2d', 'opencv_imgcodecs', 'opencv_highgui'],
                LIBPATH=['/home/rachel/git/spectator', '/usr/local/lib'])
env.Program(['testBlob.cpp'])
```

It's clear from how short this file is, compared to a Makefile or CMakelists file, that SCons will help save me and my team a lot of time this semester.

This week, I also constructed a mock table with attached contact microphones to aid in the testing of Micah's bounce detection code. This is just a piece of thin MDF with a piezo disk attached, with extra long wires so it doesn't need to be constantly moved and stowed away like the full table.