Exceptions in Curses Wrapper

Time it took Matthew

10 - 20 minutes

What to submit

A **FOLDER** named CursesWrapper that contains

- A CMakeLists.txt that can compile your program
    - You should start with the one that I've given you in the handout
    - Likely the only changes that you are going to need to make to it are in the add_library command if you add more files to your solution
- All of the .cpp and .h files in the CursesWrapper subfolder that I have given you
- Any additional .cpp and .h files that make up your solution

Problem Description

In order to make our Curses Wrapper code more robust we are going to add bounds checking and if an access is out of bounds, throw an exception. We will do bounds checking on all methods where the user gets to specify an arbitrary position to move the cursor. Those methods would be

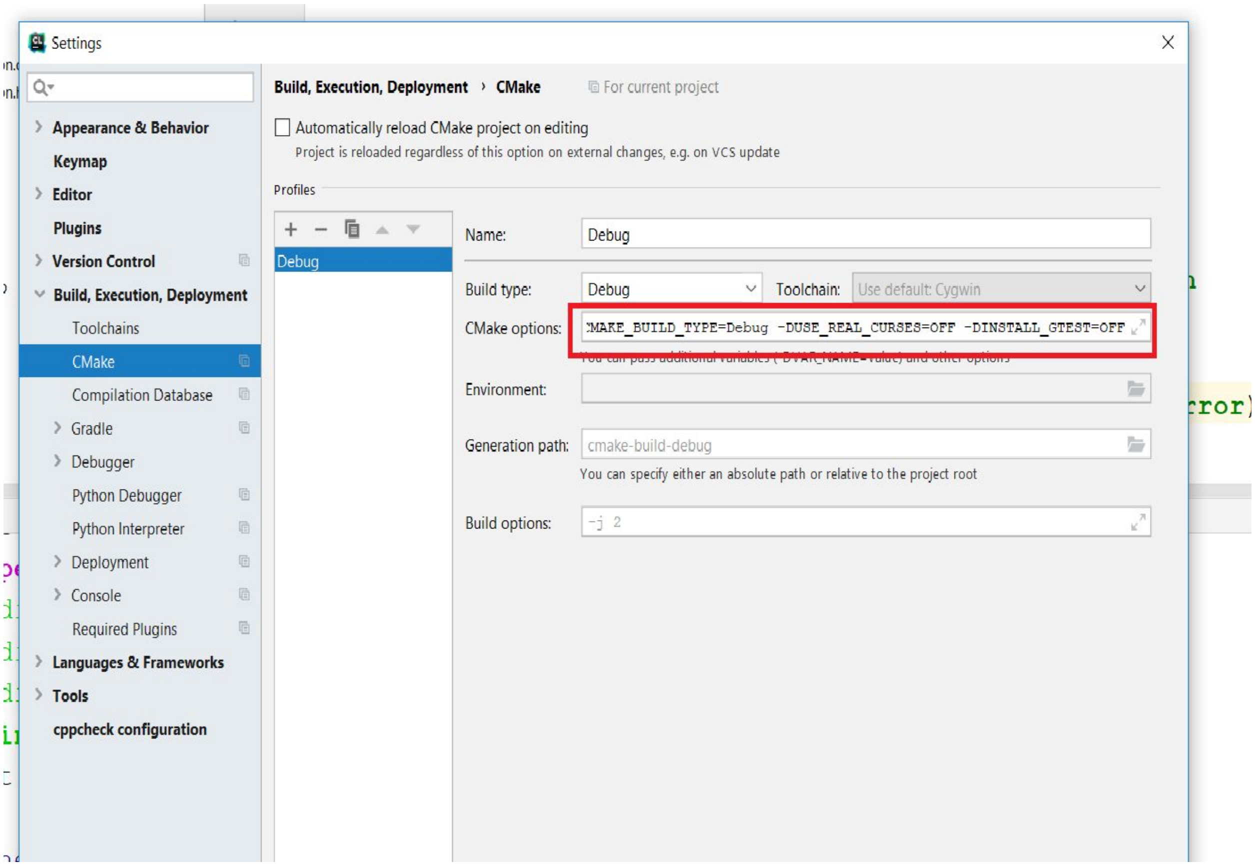
- `virtual char getWindowChar(int row, int col)`
- `virtual char getCharInput(int row, int col)`
- `virtual std::string getStringInput(int row, int col)`
- `virtual void addCharacter(int row, int col, char value)`
- `virtual void addString(int row, int col, const std::string& str)`
- `virtual void moveCursor(int row, int col)`
- `Accessing`

An access is only illegal if the specified row,col location is out of bounds. If an access is out of bounds you should throw an std::out_of_range exception or your own personal exception that inherits from std::out_of_range. Std::out_of_range is defined in <stdexcept>.

You are free to add code to any of the files in the CursesWrapper subfolder including Window.h to implement the above behavior.

CMake Options

We will once again be using Fake Curses to help out with the testing so you will have to add the -DUSE_REAL_CURSES=OFF option to CMake to allow the testing to compile. Again that option is found under Settings->Build, Execution, Deployment-> Cmake in the CMake options field in CLion or if you are using command line just add -DUSE_REAL_CURSES=OFF when running CMake.



□ Automatically reload CMake project on editing
Project is reloaded regardless of this option on external changes, e.g. on VCS update

Name: Debug

Build type: Debug Toolchain: Use default: Cygwin

CMake options: `:MAKE_BUILD_TYPE=Debug -DUSE_REAL_CURSES=OFF -DINSTALL_GTEST=OFF`

You can pass additional variables (-DVAR_NAME=value) and other options

Environment:

Generation path: cmake-build-debug
You can specify either an absolute path or relative to the project root

Build options: -j 2

Hints

- Start with the code that you used for the CursesWrapper assignment and add in the bounds checking to it. It should be fairly simple and only take about half an hour to an hour to do
- If you did not complete the CursesWrapper assignment you can find my solution on Canvas after the late submission period has passed.