- War game encashment on BS
    - Due feb 5
    - Have abstraction called player
    - Should be no more than 50 lines
    - War can be recursive
    - Have a readme file to describe it
    - Put it in sub directory
    - Also do milestone one and choose a game to work on
        - Need SFML installed on computer
            - Just look up sfml
            - Simple enough to get started quickly
            - Itss gui stuff
            - Get windows subsystems for linux
                - Gives you command line in windows
            - Got a week to do it
            - Might have access to VS studio through university
                - Imagin
- #pragma once
    - Similar to include guards
        - These prevents adding the same text to one text multiple times
    - Shorter way to do the same thing
        - Not part of the language
        - It is windows extension but everyone uses it
- Don't need trailing comma in enums
    - But it makes nice anchor for multiple selection
- Only write a makefile if you don’t have better solution
    - We have better ways now
    - Gcc is C compiler but is smart to switch to c++ if it sees its needed
    - There’s also clang
    - This is not a good way to compile but you can't just do it through the command line
        - Dumps build files in same directory as source code
            - This is not desigered
            - Anything created in compilation should never be added to your repository
    - -E
        - Will show the intermediate file when compiling
            - Replace -c with -E
            - | wc-l
                - Will let you see lines of code
    - -s
        - Will also show what is being done cmake instead
    - Use cmake instead

        - It will generate the makefile
        - Can open CMakeLists.txt
            - add_exicutable (name *.cpp, main.cpp)
        - Always put it into a subdirectory
            - Call it build
        - Then run it cmake ..
            - The .. is to jump up a level
        - Ccmake
            - Is a function to pull up a program to find all
                - 2 build types
                    - Debug
                        - Type it into use
                        - To debug it if it crashes
        - rm -rf build to clean the build
        - Can compile for release as well
            - For better optimization
- For the game also grabe the cmake directory it will find it and work on system
    - Enums class Suit{};
        - Will force namespace rules so you call it like
            - Suit::Hearts
    - Our fake enum will effectively catch problem as runtime but enum catch it on compile time error
        - So no one can compile it wrong
        - It becomes a type error
            - We should be able to make it a type error for common errors
    - With war design the program to be incorrect when run if used wrongly