Programming Language Course

06 mar. 2021

Please deliver on the OneDrive shared folder named after you last name:

- A folder named cardgame containing C code files and C header file, and a CMakeLists.txt file
- The files can be organised into subfolders
- The command `cmake --build .` should generate an executable file (provided that the correct path have been set with `ccmake` or `cmake-gui`)
- The libcard project should be delivered **in a separate folder**
- The raylib dependency should **not** be delivered

Expected features: *(-2)*

- Draw cards from a deck
- Move cards across the board
- Move cards to a discard pile
- Use the types and functions from the libcard
- Reset button that restores the initial state

Bonus features: *(+1)*

- CMake project works with VS on Windows and with Makefiles on Linux
- libcard is linked as a shared library (dll/so)
- Cards can be drag-and-dropped (they follow the mouse until button is released)
- Clicking on an object never triggers an action an another object than the obvious one
- The board has bounds and cards can not be moved outside them
- Buttons and cards change color when hovered
- “Add deck” button that adds another deck to the screen *(+2)*

Coding safety: *(-2)*

- Free every memory that you have allocated
- Check the success of every memory allocation
- Don’t write code that relies on uninitialised values
- Don’t let the possibility of dereferencing an invalid pointer happen
- Don’t let the possibility of accessing an array out of bounds happen

Coding style: *(-2)*

- Don’t duplicate code
- Use proper and consistant indentation
- Use a consistant naming scheme
- Use type and function names that make sense
- Use comments to indicate anything that isn’t obvious
- For each structure, have a .h and a .c file following an oriented-object approach
- Have a clean main function calling functions like update and draw on object
- Mark as `const` all pointer function parameters that are not modified by the function *(+1)*