12: Project Documentation

CSCI 6626 / 4526 May 2018

1 Goals

- To put it all together.
- To provide minimal project documentation.

2 Requirements

I want you to use all the techniques I have taught; you must choose where and how to use them.

- Define an extension in at least one class to at least one operator OTHER THAN `<<` or `>>`. If you can't find a really good reason to use an operator extension, rename one of the functions you already have. Be sure to USE your operator extension, not just define it.
- Use const wherever you can in your program.
- Use ctors in some of the *optional* places.

3 Documentation

Submit these, in the order given, electronically if possible:

- (5) A table of techniques. For each of the required elements listed above, give the class name(s) and function name((s) in which you used that technique. (If I can't find it, you don't get credit.)
- (5) A proof that you have no memory leaks. Preferably, run the code through Valgrind. Otherwise, add trace comments to every constructor and destructor (including your copy constructor) for every class that you allocate with `new`. Put a static int class variable in these classes; increment it in every constructor, decrement it in the destructor. At the end of the game, print out all the counters, which should be 0.
- (5) Graphic documentation for the program or parts of the program, including a UML diagram (boxes, names, and links, no other details) of all classes, a detailed UML diagram of your dice class hierarchy, a state diagram for the logic in oneTurn, and an interaction diagram for oneTurn.)
- (5) A makefile that will run on both your system and mine.