C++ Programmeermethoden Assignment5

Bas Terwijn

May 7, 2020

1 Introduction

Assignment5 replaces the 2020 C++ Programmeermethoden exams because of ongoing online teaching due to Corona virus measures and will determine your grade for this course. In this assignment you are asked to modify and extend the ***VirusGame*** software.

2 Tasks

You are asked to do the following tasks but first install and run VirusGame on your computer using the installation instructions provided and read the documentation and code to get somewhat familiar with it.

2.1 task1: Polymorphism

Currently in VirusGame.cpp all units are stored in a static array:

```
Virus units[max_nr_units];
```

But we want to be able to add other classes as units besides only the "Virus" class. In addition we want to handle the "player" object as just another unit so the code get simpler. Therefore change the "units" array so that units of different classes can be added to it using dynamic polymorphism.

2.2 task2: Avoid Memory Leaks

With polymorphism you often will dynamically allocate memory when you instantiate objects using the "new" keyword. Avoid memory leaks by de-allocating the memory when it is no longer needed.

2.3 task3: RAII

With ***Resource Acquisition Is Initialization (RAII)*** you can make sure you, and possible other people that later might use your code, won't forget to release or de-allocate any resources such as memory. This is done by putting the code that releases the resource in a destructor that is automatically called when an object goes out of scope. Use RAII to release any resources in your code, for example the memory allocated for units.

2.4 task4: STL Containers

The modern ***Standard Template Library containers*** are the preferred data structures to use. Prefer std::vector over a static array [] as it can grow to arbitrary size, it knows its own size, doesn't decay to a pointer when passed to a function, and has only little additional overhead compared to a static array. Therefore replace any static array in your code (for example: Virus units[max_nr_units];) with a std::vector and prefer stl containers if you choose to add other data structures.

2.5 task5: STL Algorithms

ES.1 of C++ Core Guidelines recommends using the standard library over "handcrafted code". Therefore use as much as possible the functions defined in the ***STL Algorithms Library*** instead of for example raw for-loops. For a gentle introduction to STL Algorithms see the ***"105 STL Algorithms in Less Than an Hour"*** talk by Jonathan Boccara.

2.6 task6: Avoid Duplicate Code

Avoid having duplicate code or expressions, or said differently, don't repeat yourself (DRY). The Virus::step() function is currently already a duplicate of Player::step(). Find a good way to avoid that and other duplication.

2.7 task7: Your Own Creative Extension

The VirusGame is not yet finished. Extend it so it has interesting game play. Maybe the player has to avoid touching the viruses, or shoot them, or bump into them to bounce them into an anti-virus unit. Maybe also add some special effects like explosions or track marks or keep a score. The more creative the better, make it fun. You are free to change any of the provided source code. Write a short description of your extension with references to source code at the bottom of the README.md file just so that I don't miss anything when grading your submission.

3 Grading

Your grade will follow from which tasks you complete to a satisfactory level:

task	**points**
task1: Polymorphism	2
task2: Avoid Memory Leaks	1
task3: RAII	1
task4: STL Containers	1
task5: STL Algorithms	1
task6: Avoid Duplicate Code	1
task7: Creative Extension	3

Points will be deducted if your code is not "simple" such as described by Kate Gregory in her ***"Simplicity: not just for beginners"*** talk, watch it!

4 Rules

You are not allowed to share code with other students, if we detect (manually or with plagiarism checkers) that different submissions have similar structure I will have to report that to the examination board. See the UvA "Fraude en plagiaat regeling" for more details.

Therefore if you optionally choose to fork the VirusGame git repository so you can use git to track and backup your changes, then make sure the repository is private otherwise you could get accused of plagiarism if someone

copies your code. Bitbucket allows you to have private repositories if you use your "@uva.nl" email address for your user.

5 Submission

Submit your code as a zip/tar of the whole VirusGame project before the deadline on May 31 23:59. Remove the compiled executable and other derivatives that I don't need in order to compile your code before you zip/tar to reduce the size. If you use other dependencies (additional libraries) describe them so I can easily install those before compiling your code.