Using Image Processing and Visualization to Teach Object Oriented Programming in C++

Author:
C. Rolstad, Oslo University College, Cort Adelersgate 30, 0254 Oslo, Norway. cecilier@iu.hio.no

Abstract — When teaching engineering students programming a goal is to make them write as many programs as possible, since this is a proven way to develop good programming skills. Textbooks on object orientated programming (OOP) often provide examples that are not very motivating for younger engineering students. The examples may be manipulation of text and numbers, e.g. examples from banks where the various account types are the objects. Such examples certainly illustrate OOP, but they do not motivate the students for further programming exercises. In this project we have introduced basic image processing as examples in the lectures and as exercises in the course. Image processing is used to explain OOP techniques and also to motivate students to write programs. Topics such as classes, inheritance and polymorphism, operator overloading, templates and one- and multidimensional arrays are covered. Freely available software is used in the exercises to display the images to the screen, and the image manipulations are programmed in C++. The students have found the manipulation of images very rewarding, with the result that they have programmed more than required during the course.

Index Terms — C++, Digital Image Manipulation, Object Oriented Programming, Visualization.

INTRODUCTION

Programming is probably the most important topic in the education of a computer engineer, it forms the basis for understanding how a computer works and importantly how the computer can work for you. At Oslo University College the first-year students have mandatory courses in basic programming during their first year. Learning to program is an effort for a beginner, so being motivation is important. Not all the young beginner students are used to having to concentrate and struggle to get through the curriculum, and the programming quickly becomes a challenge. All students are taught Java programming in their first year, but they are also offered an optional course in C++ programming. The course is surprisingly popular, considering that they already know Java, but the students are aware of the advantage in knowing some C/C++ syntax since so much software is developed in this language. We have tried to add some fun to the course by relating the object oriented programming (OOP) to visualization. The exercises are based on simple image processing, starting out by displaying an image and manipulating it in a simple way. At the end of the course they conduct a more challenging exercise where they develop their own digital Christmas calendar.

Pedagogical premises

The course is optional and in autumn 2002 seventy students participated. We offer two hours of lectures and two hours of programming laboratory weekly. The lectures are held in a traditional auditorium, and the lecture notes are available on the Internet. Weekly exercises with hints are also available on the Internet, and solutions are provided about two weeks later. The students must conduct a digital multiple choice test every week, where syntax and output of small programs are tested. This is to ensure that they read the curriculum. The students are in general quite pleased with multiple choice tests as they receive instant feedback. It is also a way for the teacher to get an overview of the students as the results are recorded automatically in a database. In addition to the weekly exercises the students must hand in two mandatory exercises. The final evaluation is through a three hour written exam.

WHAT IS OBJECT ORIENTED PROGRAMMING?

Object oriented programming involves the use of encapsulation, classes, inheritance, and polymorphism. Here encapsulation means hidden information or abstraction. Abstraction means that the details of the class implementations are hidden for other programmers. An abstraction in C++ is formed by creating a class, and using this class one can declare an object. Inheritance
in C++ enables you to write reusable code. Polymorphism is used in connection with inheritance and enables a name to have several meanings. These topics are described further below in addition to some other topics that are taught in the course.

The OOP language Java has it’s own garbage collector, which returns unused memory to the freestore manager. We have focused on memory allocation and teaching the students to program their own garbage collection by implementing classes which contain constructors, destructors, overloaded assignment operators and when necessary a copy constructor.

**Classes**

A class is a type whose values are called objects. A class contains several member variables that are predefined types such as integer, float etc. A class may also contain member functions that are declared in the class but usually defined elsewhere. A member function usually manipulates the data of the member variables. The access to member variables and member functions outside the class is determined by the key words `public` and `private`. If a member value is `private` it can only be changed using one of the member functions. Member variables and member functions are called using the dot operator after the name of the object.

In this course we use digital images as objects, we use a predefined class `RGBpixmap` from OpenGL which stores the image as a one dimensional array. The class has a private variable `pixel` of type class `mRGB`, which has public member variables `r,g,b` of type unsigned character. The class `RGBpixmap` has member functions for displaying the image, reading pixel values into a rectangle and into one pixel only. We also define our own class `image` where the data is stored in three two dimensional arrays, see further descriptions in ‘the Second Exercise’ section.

**Inheritance**

A new class can be derived from an existing class using inheritance. The new class is often called the derived class, and the existing class is called the base class. When deriving a class using inheritance, access to the member values and the member functions in the base class can be restricted using the key words `public` or `private`.

**Templates**

Templates enable reuse of written code, since function and classes can be declared without determining the variable types. A template class is used in the second mandatory exercise.

**Abstract class**

A class is called abstract when it has one or more declared virtual member functions. These functions are not given any definition, they are set to zero. Objects of an abstract class type cannot be declared. In order to create an object based on an abstract class one must use inheritance. A new class must be derived from the abstract class, and the virtual function must be given a definition in the derived class.

The use of templates, inheritance and an abstract class is demonstrated in the second mandatory exercise, where we have programmed a Christmas calendar using inheritance from an abstract template class. Inheritance is necessary to give definition to a pure virtual function in the abstract class, and an object of required type can be declared.

**Polymorphism**

The use of abstract classes and virtual functions enables so called late binding. A virtual function is given a definition for each derived class, and the definition to be used is determined by the object type called. The definition given in the derived class will always be used for an object of this derived-class type. Polymorphism is the ability to associate many meanings to the same function name by the late-binding mechanism.

**Operator overloading**

In C++ one can overload operators, which gives a new functionality to the operator. Overloading can be conducted only if one of the operands is a class.
Overloading is used in this course when manipulating digital images of the two class types image or RGBpixmap. Images can for instance be multiplied together, or an image can be faded out of the screen by repeatedly being multiplied with a factor less than zero and flushed to the screen.

One- and multidimensional arrays

Arrays can be of any type in C++. The indexing starts at zero. Array indexed variables are always placed next to each other in memory, and the computer knows the address of the first indexed variable, a[0]. Referencing a non-existent out of range array-index is a common mistake, and other variables may unintentionally be overwritten. This mistake is pointed out to the students. When arrays are used as function parameters, the address of the first indexed variable is known to the function, but not the array size. The base type of the array must match the formal parameter of the function, therefore the function knows the memory requirements for each indexed array variable. The students are told to include an integer variable representing the number of elements in the array, so that the function knows the total array size. When the array size is known one can check if an index is within range.

Two or multidimensional arrays consist of arrays of arrays in C++. The two dimensional array char page[30][100] is a one-dimensional array of size 30 whose base type is a one-dimensional array of characters of size 100 [1]. When two-dimensional arrays are used as function parameters the size of the first dimension can be skipped since it is ignored by the compiler, only the second dimension must be included to specify the base type of the two-dimensional array. The principle of multidimensional arrays is explained to the students.

Both one and two-dimensional arrays are demonstrated through the use of digital images in the classes image and RGBpixmap.

Namespaces

Programmers can use the same function names when identifying them by placing them in separate namespaces. This is useful when several programmers are working together on projects. The students practice this by working in a group for the second mandatory exercise, implementing a function with the same name and then placing it in their own private namespace.

WHAT IS A DIGITAL IMAGE?

A digital image is usually stored as a two-dimensional array. Each element is called a pixel, which is short for picture element. The images can be in black and white (binary images), or they can contain gray values represented by pixel values from 0 to 256. Color images can be represented by overlaying three images, these three images may then be called channels. One channel would be in red (R), one in green (G), and one in blue (B), thus giving an RGB representation of the image. Each of the three colors is represented by pixel values from 0 to 256. An image file contains pixel values and usually some header information at the beginning of the file.

VISUALIZATION

Image display is done using the free software OpenGL which was developed by Silicon Graphics Inc. in 1992. It is much used as a user interface for graphical applications. The software can be downloaded for free from the Internet at http://www.opengl.org/Downloads/Downloads.html. Literature on the software is found on e.g. http://fly.cc.fer.hr/~unreal/theredbook/.

Mandatory Exercises

For the two mandatory exercises the students are allowed to work in groups of three, which usually makes the process more fun for them. We have also noticed that the students who work together in groups seem to be the cleverest ones, and we are convinced this is because they discuss and exchange their knowledge. Some experienced colleagues say that students learn more from their peers than from their teachers. The two mandatory exercises are described further below.
The first exercise

The aim of the exercise is to combine basic C++ skills with pre-programmed functions from the OpenGL library. We think it is good to be able to use existing functions at an early stage as a motivational factor, because one can achieve more and develop more interesting programs in this way. The basic C++ skills in demand at this stage are classes, separate compilation, call-by-constant reference parameters, arrays and operator overloading.

In this exercise we use the format *.BMP which is a much used format in graphical programs. The BMP format contains header information and three bytes per pixel, where each byte represents the RGB values for the pixel. We work with a predefined OpenGL class called RGBpixmap, which handles *.BMP files, and we conduct simple image manipulations on the image using C++ code developed by the students.

The students are asked to smooth an image by calculating mean values in a 3 × 3 window, and let this window slide across the image. They then multiply two images together using operator overloading on the class RGBpixmap. They are asked to fade out the image from the screen by multiplying the image by a factor using their overloaded operator, and sending this to display in a loop. They are told to use constant call-by-reference parameters when dealing with images, since images take up much memory.

The second exercise

In this exercise we store the image in our own defined class called image which has three two-dimensional arrays, one each for red, green and blue. We use dynamic memory allocation so that the images can vary in size. Classes with dynamically allocated data should have a default constructor, destructor, and overloaded assignment operator ‘=’, which is sometimes called ‘the big three’[1]. For the destructor the students are reminded to include a delete for each new, to avoid leakage of memory. Memory allocated by new is not returned to the freestore manager if delete is not called. They also have to program a copy constructor which conducts deep copy. We use the same OpenGL RGBpixmap class to read files from *.BMP files. The two functions RGB2Image and Image2RGB, which convert from one-dimensional RGBpixmap class to two-dimensional image organization in the image class, and visa versa, must be programmed.

The Christmas presents in the calendar can be small images, Christmas rhymes in an ASCII format, or digital soundtracks. We can therefore use an abstract template class when programming the calendar since it contains a background image with 24 presents, and when using the calendar template one must specify the type of present. Three alternative ways for opening the various types of presents are needed. The template class christmasCalendar is declared with a background image of the class type image, and it has 24 unspecified Christmas presents. It has a pure virtual function OpenPresent(int date). A class imageChristmasCalendar is declared by inheritance from the template class, and the function OpenPresent(int date) is defined for an image in the imageChristmasCalendar class. The function opens as many images as specified by the parameter date reading an image list from a text file, and it overwrites these images on to the background image. Finally the resulting image is converted back to the OpenGL RGBpixmap class and displayed.

To use namespaces each student in the exercise group must program one function called filter which they put in their own namespace. The function should alter the appearance of the background image. The functions are used for various dates so that the background varies through December, and it is called using the personal namespaces. The functions can for instance alter colors, put frames on the image, smoothen or sharpen it.

The program works as follows; the user types a date, and then background and appropriate number of Christmas present images are opened and converted from RGBpixmap class to image class using RGB2Image. The background image is filtered using one of the filter functions determined by the date, the Christmas present images are overwriten on the background image, the resulting image is converted back to RGBpixmap using Image2RGB, and it is finally displayed.

CONCLUSIONS

Digital images have been very suitable for teaching OOP for engineering students. Images fit well into the class concept, and exercises that use the main possibilities in C++ can be developed so that much of the language can be demonstrated and used in practice. Using the predefined library OpenGL gives the students a tool to manipulate and display images early in the course. This is motivating for them and in this way they learn about separate compilation. The need to use existing code in combination with their own will often occur in their professional career so this is good experience for their future profession.

International Conference on Engineering Education

The students really enjoy the image manipulation from the start. They gradually develop knowledge in C++ which enables them to program classes and functions dealing with digital images. At the end of the course they mainly use OpenGL only for display. They hand in programs far extending the demands given in the mandatory exercises, and they show excitement in the PC laboratories as they realize the possibilities that are open for image manipulation.

REFERENCES