CS101-2021

Additional Concepts used in the Project

Some concepts used in the project starter code have not yet been taught in the course. This document will give you a brief overview of these concepts.

Header files (.h)

As programs grow larger, it becomes increasingly tedious to write all the code in a single .cpp file. So we split our program into multiple files. In the project, we split different classes into separate files called header files which usually have a .h extension, and use a single main.cpp file.

We include the contents of the header files in the main.cpp file using the command  
#include "header.h"

Similarly, the command #include <simplecpp> also includes the content of the simplecpp library header files into your .cpp file.

Later in your programming life you will learn about code split into multiple cpp and h files that will have to be compiled separately and linked together to form a larger program. But for now we can use this simple include mechanism to split code into multiple code and still compile only once.

vector

A collection of data of similar types, for example a list of integers or a list of names can be stored into an array. We will learn about arrays in the course. However, arrays that we will learn about have a fixed number of elements. Sometimes you may want to store multiple elements of a type but do not know how many such elements you will have to store before the program runs. In such a case, memory has to be allocated for such dynamically growing lists or collections during the time when the program is running. We can extend arrays with dynamic memory allocation.

However, there is a neat solution to this problem using a container data type in C++ (i.e., it can contain elements of other data types).

In the project starter code, to store multiple elements of a particular data type, we use the vector container data type. To create a vector of data type int, we can use the following command

vector<int> v;

To add elements in the vector v, we can use the push\_back command. For example to add 3, 1 and 5 in the vector v, we can use the following commands

v.push\_back(3);  // v contains 3

v.push\_back(1);  // v contains 3, 1

v.push\_back(5);  // v contains 3, 1, 5

You can access the elements in the vector using the index. For example to access the 2nd element of the vector, we can use the following command

v[1];  // indexing starts from 0, so 1 is the 2nd element

To delete an element in the vector, we can use the erase command. To delete the 3rd element (index 2) in the vector, we can use the following command

v.erase(v.begin()+2);        // v.begin() points to the first element in the vector

More information about vector can be found at [https://www.cplusplus.com/reference/vector/vector/](https://www.google.com/url?q=https://www.cplusplus.com/reference/vector/vector/&sa=D&source=editors&ust=1644938707262182&usg=AOvVaw3i5Ei62iFDkrxZRWazXDIN).

string

A string is a collection of sequential characters. C++ has a string data type which can be used as follows

string text = "Hello World";  // string literals are placed between double quotes

More information about string can be found at [https://www.cplusplus.com/reference/string/string/](https://www.google.com/url?q=https://www.cplusplus.com/reference/string/string/&sa=D&source=editors&ust=1644938707263604&usg=AOvVaw0Um0sdVEn6VJgFeF5pb-oC).

XEvent

In simplecpp graphics, we have seen the getClick() command that captures a mouse click. In the project, we capture keyboard presses to play the game. The mouse click and a keyboard press are examples of events. To capture such events we use the XEvent command. The starter code is already completely set up to use XEvent for capturing keyboard events, so you need not write any additional code. To know more about event based programming, you can refer to [https://drive.google.com/file/d/1f4Is5PE-7RSfPtYW2Hn1Xemg9BEKQv7u/view?usp=sharing](https://www.google.com/url?q=https://drive.google.com/file/d/1f4Is5PE-7RSfPtYW2Hn1Xemg9BEKQv7u/view?usp%3Dsharing&sa=D&source=editors&ust=1644938707265019&usg=AOvVaw3VA0GgojsmX4_X4EsJ_irs).