

## Lesson 4: Conditionals

### 3D Game Programming With C++

#### *Digital Media Academy (Summer 2011)*

**Written by:** Andrew Uzilov ([andrew.uzilov@gmail.com](mailto:andrew.uzilov@gmail.com))  
Feel free to contact me with any questions.

A program that always does the same thing is (usually) no fun. It needs to react based on user input. For that, we have conditional statements.

Conditionals decide which pieces of code to run depending on an expression that can be either `true` or `false`. So, the expression evaluates to a boolean type. For example:

```
if (tempF > 75.0) {  
    // This will only print if "tempF" is strictly above 70.  
    cout << "It sure is hot today!" << endl;  
}
```

The block of code that will run (or not run) is between the braces (or “curlies”) which look like this: `{ }`

Each conditional has two parts:

- 1) The test, which is an expression that evaluates to either `true` or `false`. The test is between the parentheses that go after `if` or `else if` keywords. In the above example, the test is this expression: `tempF > 75.0`
- 2) The code block (in braces) that will run if and only if the test evaluates to true.

You can put several blocks together for more complicated conditionals:

```
if (tempF > 75.0) {  
    cout << "It sure is hot today!" << endl;  
}  
else if (tempF > 60.0) {  
    cout << "A very pleasant day." << endl;  
}  
else {  
    cout << "I'm getting my jacket." << endl;  
}
```

Useful conditional (logical) operators:

Operators	Meaning	Example of an expression
==	Are two variables equal?	a == b a == 5
!=	Are two variables unequal?	a != b a != 5
< <= > >=	Are two variables less than, less than or equal to, greater than, greater than or equal to each other?	a < 5 a >= b
!	Reverse the logical value of a variable or expression. If it was true, it becomes false, and vice versa.	!a !(a < b)
&&	Are two logical values the same? (“and”)	(a < b) && (a == 5)
	Is at least one of the two logical values true? (“or”)	(a < b)    (a == 5)

**Caution:** Do not confuse logical operators && (which means “and”) and || (which means “or”) with the operators & and | – they are NOT the same thing!

Also, do not confuse == with = symbols. The first one is a logical operator, the second one is used for variable assignment.

In both cases, your code might compile, but it will probably not work how you intend.

Several conditional expressions can be combined into more complex expressions, even with boolean variables, for example:

```
bool raining = false;  
if ((tempF >= 65) && (tempF <= 105) && !raining) {  
    cout << "This is beach weather!" << endl;  
}
```

Conditionals can be nested inside each other, which allows richer control over the game flow. Here is an example: if the player goes off the left or right side of the screen (which is 1920 pixels across), the player loses a life as long as they have more than one life left; otherwise, it's game over.

```
if (playerX < 0 || playerX > 1920) {  
    cout << "Player is off the screen!" << endl;  
    if (numLivesLeft > 1) {  
        cout << "Lose a life, but keep playing." << endl;  
        numLivesLeft--;  
    }  
    else {  
        cout << "Game over!" << endl;  
    }  
}
```

#### Exercise 4.1

In the code example above, figure out what will print if:

- playerX is -100 and numLivesLeft is 5
- playerX is 550 and numLivesLeft is 3
- playerX is 2100 and numLivesLeft is 2
- playerX is -312 and numLivesLeft is 1

DO NOT answer this question by typing code into Eclipse! Call your instructor over and tell them the answers by just reading the code and thinking about it – no typing.

#### Exercise 4.2

Write a program that asks the user to input an integer, then writes to the console whether the input is positive, negative, or zero.