**Code Style**

1. Prefix

For certain variables, classes or functions, some prefixes are required.

|  |  |  |
| --- | --- | --- |
| Prefix | Purpose | Example |
| g | Global Variables | g\_counter |
| m | Member Variables | m\_counter |
| p | Point Variables | pCounter |
| V | Virtual Functions | VDraw |
| I | Interface Classes | IDrawable |

Some prefixed can be combined together. Ex) m\_pCounter

## Use understandable and sensible variable, class, and function names.

Don’t call something like MyFunctionThatHandlesPlayerHealthUnlessTheyAreInGodMode(), but also don’t call it Handler(). If someone else can’t understand the high level purpose of something by its name alone, it should probably be renamed.

Our project follows the following patterns for naming:

|  |  |
| --- | --- |
| **Local Variable** | **variableName;** |
| **Function** | **FunctionName();** |
| **Constant** | **CONSTANT\_NAME** |
| **Class** | **ClassName** |
| **Class File** | **ClassName.cpp** |

1. Every single if/else/for/while block should have its own braces.

|  |
| --- |
| **DO**  **bool found;**  **if (found)**  **{**  **found = false;**  **}**  **else**  **{**  **found = true;**  **}** |
| **DON’T**  **float myVariable = 0.0f;**  **for(int count = 0; count < SomeObject.Instance.numberOfThings; count++){**  **myVariable += Bar();**  **}** |

## Initialize your private variables.

|  |
| --- |
| **private int m\_myVar1,m\_myVar2,m\_myVar3;** |

This is not allowed.

|  |
| --- |
| **private int m\_myVar1 = 0;**  **private int m\_myVar2 = 0;**  **private int m\_myVar3 = 0;** |

This is allowed.

## Whitespace after code block.

## An empty line after a code block can be added depend on situation

|  |
| --- |
| **float myVariable = 0.0f;**  **for(int count = 0; count < SomeObject.Instance.numberOfThings; count++)**  **{**  **myVariable += Bar();**  **}**  **// this line can be added or not**  **BarTwo();** |