

# Writing Good Code

October 19, 2005

## 1 Introduction

Good code is like many other good things in life: easy to recognize but sometimes difficult to define. The following sections describe some things which make some programs better than others. All of these things have more to do with style and formatting than what the program does or how it operates. Believe it or not, coding style is important. Writing code which has no style or is poorly formatted will cause other programmers to snicker at you behind your back. Don't let this happen to you. Choose a good style and use it.

## 2 Naming Stuff

The first advice I have is related to naming things in your program. Always name things according to what they represent or what they do. For example, a variable which represents the x velocity of a moving gargoyle could be named `gargoyleXVelocity` or `gargVelocityX` but not `billTheCat`. Good naming makes your program easier to read and understand.

## 3 Code Formatting

The form of your code is also important. Code inside for loops and if statements should be indented two or three spaces. Closing curly braces should also be indented as shown:

```
while ( stillMoving == true ) {
    if ( index == 0 ) {
        gargVelocity += velocityIncrement;
    }
}
```

Be consistent! Consistent indentation may help determine if you are missing a curly bracket. Code with inconsistent indentation looks ragged.

## 4 Commenting Code

Comments should tell a story about what the code is trying to achieve. Keep in mind that comments are not always for you. They are usually for other people (such as the poor teaching assistant grading your program. Take pity and provide good descriptions). Occasionally the comments are for you when you look at your program two weeks later and have forgotten what it does. I recommend writing any comments as you are creating the code (or even before you start actually writing code statements. Sometimes comments can be used as an outline to tell you what code still needs to be written).

At the top of your program should be a comment which describes the purpose of the program, who wrote it (hopefully you) and when it was created and last modified. Each class you create should have a similar comment.

You should also comment each class method or function. Describe what the function expects for input, what it does and what value (if any) it returns. Also describe any assumptions or limitations which you think might cause someone to use the function incorrectly. For example, if the function assumes that the Stage is larger than 500 by 500, you should state that in the comment.

It is usually a good idea to include a comment for each variable you declare. This should be short and state what the variable represents or how it will be used. For example:

```
// This is the maximum number of gargoyles allowed on the screen
var maxGargoyles:Number = 25;
```

```
// This array contains the gargoyle movie clips
var gargArray:Array = new Array( maxGargoyles );
```

It is also a good idea to include comments before a for loop or if statement. Use the comment to tell others what the loop or if statement is going to do. As an example:

```
// Test if the bad gargoyle has collided with another gargolye
// If so, make the other gargoyle smaller
if ( theBadBall.hitTest( gargArray[index] )
```

```
{  
  gargArray[index].xscale = 50;  
  gargArray[index].yscale = 50;  
}
```

This may seem like a lot of extra work. It is. But if your code has style you will become a better programmer. It will also help prevent snickering.