



# Nitrogen Interactive Coding Standard

---

## Variables

- Avoid using single or double letter variables (except in iteration)
- Variable names should use camelCase, starting with a lowercase letter. Every word following the first word should be uppercase.
- Boolean variables should use is or has prefixes.
- Counting variables should use postfix logic.
- Constants should be all caps, with underscores for spaces.
- Names must clearly describe the object or value they represent.

### Examples:

```
example  
aVariable  
hasAnExample  
EXAMPLE_CONSTANT
```

---

## Classes and Structs

- Use PascalCase for all types. Each word in the name starts with an uppercase letter.
- Names must clearly describe the object or value they represent.

### Examples:

```
public class ClassExample {}  
public struct StructExample {}
```

---

## Functions

- Use PascalCase for all types. Each word in the name starts with an uppercase letter.
- No spaces inside parentheses when writing parameters.

- For class accessors, use `getVar()` and `setVar()` names.
- Names must clearly describe the action they represent.

### Examples:

```
FunctioningExample(int example)
getExample()
setExample
```

---

## 3. Program Flow

### if, else if, else

- Place a space after each keyword but not inside parentheses.
- Open curly braces `{` go on a new line after the condition.=
- If the condition spans multiple lines, break based on operators.
- Must have spaces in between operators (`||`, `&&`, etc.).
- No empty newlines in between related if, else if, and else statements.

```
if (condition1
    && condition2
    && condition3)
{
    statements;
}
else if (condition)
{
    statements;
}
else
{
    statements;
}
```

---

### Loops

- Follow the same conventions as if statements.
- Must have spaces in between operators (`>`, `<`, `<=`, etc.)

- Use i and j for numeric counters and it for object iteration.

```
while (condition)
{
    statements;
}
```

```
for (int i = 0; i < 3; i++)
{
    statements;
}
```

---

## Indentation

- Use 4 spaces per indent.
- Use no more than one whitespace newline in between sections

### Example:

```
int main()
{
    while(condition)
    {
        statements;
        if(condition)
        {
            statements;
        }
        statements;
    }
}
```

---

## 4. Comments

### Inline Comments

Use `//` for inline comments. Write comments only when code is not immediately clear. Start with a space and follow correct sentence punctuation.

**Example:**

```
// Here is a comment.
```

## Multiline Comments

Use `/*` for multiline comments. Write comments only when code is not immediately clear.

**Example:**

```
/*  
Here  
is  
a  
comment.  
*/
```

---

## 5. Error Handling

- Use `Debug.LogError()` or `Debug.LogWarning()` for runtime error reporting.
- Use the `assert.h` library to handle other errors.

**Example:**

```
Debug.LogError("Error");  
assert(scene != Null);
```