Coding Guidelines for Odysseus

**General Rules**

1. Consistent case of methods and properties
2. Descriptive methods and class names
3. Keep methods at a reasonable length
4. Place A brief description of what method or class in comments
5. Follow the conventions
6. Use comments wisely and appropriately
7. DO NOT DUPLICATE CODE
8. Avoid hard coding strings and numbers, unless it is necessary
9. Test-first programming (make small changes at a time)
10. When a task is completed (methods, classes, etc), debrief with team.

**Style Guidelines** (To keep code consistent please refer to the following “good” and “bad” examples.)

1. **Indentation:** Nested “for” loops should be indented one after the other to show the order. “if”, “else if”, and “else” statements do not need to be indented they should all be aligned.

Example: for (i = 0; i < 10; i++) {

if (!something (i))

continue;

do\_more ();

}

Code should be written this way to save space.

1. **Performance and Readability:** it is more important to be correct and maintainable rather than fast. “fast” code is difficult to maintain and looked down upon in the industry.
2. **Spaces:** only use a space before the opening parenthesis when calling functions or indexing. DO NOT use spaces after them.

Example (good): method (a);

array [10];

var list = new List<int> ();

Example (bad): method ( a );

array ( 10 );

var list = new List <int> ();

1. **Braces:** inside code block, put the opening braces on same line as the statement.

Example (good): if (a) {

code();

code();

}

Example (bad): if (a)

{

code(a);

code(a);

}

1. **Multiline Parameters**
2. **Whitespaces**
3. **File Headers**
4. **Multiline Comments**
5. **Casing**
6. **Line Length and Alignment**
7. **Initializing Instances**
8. **Best Practices**
9. **Documentation**
10. **Tests**
11. **Check all arguments**