Code Standards Document

* Descriptive and accurate names of classes, methods etc.
* Use long names instead of abbreviations.
* Assign a controller class to classes. For example: Triangle needs TriangleController.
* Create new classes or subclasses if possible and needed. Make sure not to make classes too wide.

Take advantage of inheritance.

* Create simple Unit Tests to test the functionality of complicated methods or classes before implementation, if code base is relatively big.
* Variables, fields and methods should be private as often as possible, especially in model classes. Use Get methods to access private fields, and let the relative Controllers handle information regarding its model class(es).
* Make separate Controllers, Facades etc. handle interaction between model class Controllers. There is no need to create dependency between model class Controllers (see above).
* Use Interfaces if any two classes have something in common, and if it “makes sense”. Use interfaces to create a contract that the classes must live up to.
* Use a MainController class to instantiate other Controllers, if the application is “big enough”.
* Use /// (code comments) to describe complicated methods.
* Don’t create a model class Controller for each class in an inheritance hierarchy. Instead just make a general one.