# Code Architecture View for Class Project

### Directory Structure for the class project

Design

<layer>\_Design.docx

Requirements

Implementation

productionbuild

<application>.exe

<application>

<application>.sln

<application>.vcxproj

integrationtest

<testname>

<testname>.sln

<testname>.vcxproj

\*.cpp

\*.h

<layer>

include

<class>.h

interface

<interface>.h

lib

<library>.lib

<library>

<library>.sln

<library>.vcxproj

src

<class>.cpp

unittest

<module>Test

<module>Test.sln

<module>Test.vcxproj

\*.cpp

\*.h

<module>.exe

IMPL1 and IMPL2:

Each **<layer>/unittest** directory contains:

* a directory for each <module>Test, which contains:
* the .cpp and .h files for the test driver and stub(s) needed to do a unit test of this module
* <module>Test.sln
* <module>Test.vcxproj
  + contains .cpp and .h files for all classes used to implement the module (point to them, don’t copy them!)
  + contains .cpp and .h files for the test driver and stub(s) needed to do a unit test of the module (point to them, don’t copy them!)
  + has the appropriate directories in the include path.[[1]](#footnote-1) You may need some interface and/or include directories:  
    $(SolutionDir)../../../GUILayer/include

$(SolutionDir)../../../GameLogicLayer/interface

$(SolutionDir)../../../GameLogicLayer/include

$(SolutionDir)../../../ModelLayer/interface

$(SolutionDir)../../../ModelLayer/include

* + builds <module>Test.exe

Note that not all modules have a driver (e.g. main()), and not all modules have stubs.

Each factory is linked into its own library, along with the class it creates. The **<layer>/lib** directory contains libraries and projects that build them, so it should contain:

* <factory>.lib
* a directory named <factory>, which contains
* <factory>.sln
* <factory>.vcxproj
  + contains .cpp and .h files for the factory (point to them, don’t copy them!)
  + has the interface directories in the include path. Set these up in the same way as for the <module>Test projects.
  + builds <factory>.lib in the parent directory[[2]](#footnote-2)

For your factory module, you will still create a <factory>Test directory containing a unit test of the factory, set up just as the <module>Test above. (This is completely separate from the VS project that creates a library containing your factory.)

INT1 and INT2:

Each **<layer>/lib** directory contains libraries and projects that build them:

* <library>.lib
* a directory named <library>, which contains
* <library>.sln
* <library>.vcxproj
  + contains .cpp and .h files for the modules in this library (point to them, don’t copy them!)
  + has the interface directories in the include path. Set these up in the same way as for the <module>Test projects.
  + builds <library>.lib in the parent directory[[3]](#footnote-3)

The **integrationtest** directory contains .NET projects for each integration test <testname>.

* Each directory <testname> contains:
* <testname>.sln
* <testname>.vcxproj
  + contains .cpp and .h files for the classes that are not linked into libraries (point to them, don’t copy them!)
  + has the interface directories in the include path.[[4]](#footnote-4) You may need some interface and/or directories, e.g.:  
    $(SolutionDir)../../GUILayer/includ  
    $(SolutionDir)../../GameLogicLayer/interface  
    $(SolutionDir)../../GameLogicLayer/include  
    $(SolutionDir)../../ModelLayer/interface  
    $(SolutionDir)../../ModelLayer/include
  + links in all needed libraries[[5]](#footnote-5)
  + builds <testname>.exe in the parent directory[[6]](#footnote-6)

The **productionbuild** directory contains:

* The directory <application> contains:
* <application>.sln
* <application>.vcxproj
  + This is set up in the same way as the integration tests, except that it builds <application>.exe in the parent directory.

1. Project; Properties; C/C++; General; Additional include directories [↑](#footnote-ref-1)
2. Project; Properties; Librarian; General; Output File: $(SolutionDir)../X.lib [↑](#footnote-ref-2)
3. Project; Properties; Librarian; General; Output File: $(SolutionDir)../<library>.lib [↑](#footnote-ref-3)
4. Project; Properties; C/C++; General; Additional include directories [↑](#footnote-ref-4)
5. Project; Properties; Linker; General; Additional Library Directories:

   $(SolutionDir)../../GameLogicLayer/lib; ../../ModelLayer/lib

   Project; Properties; Linker; Input; Additional Dependencies:

   X.lib Y.lib... [↑](#footnote-ref-5)
6. Project; Properties; Linker; General; Output File:

   $(SolutionDir)../<testname>.exe [↑](#footnote-ref-6)