CIS 660: Advanced Topics in Computer Graphics and Animation

Homework Assignment 1 (Spring 2014)

Developing Maya Plug-ins

Due: Jan. 27, 2014

The goal of this assignment is to help you setup your development environment for creating Maya Plug-ins and to start to familiarize you with building plugins in both C++ and Python.

Steps

1. Read the following:
   - Autodesk Maya API White Paper
   - Complete Maya Programming I - Chapters 2,3,4
   - Consult as needed:
     - Maya Documentation (HELP File)
     - Video Tutorials (Autodesk, Digital Tutor, Gnomon Workshop, …)
     - MEL/Python Command Reference (available online)

2. Set up Maya 2012 and Visual Studio 2010 for Maya Plugin Development (as per Appendix B).

3. Develop Hello Maya C++ Maya Plugin.

   Functionality of plugin to be implemented:

   a) Display a Dialog box entitled "Hello Maya" by typing the custom command "helloMaya"

   b) Provide arguments "name" and "id" to the hello Maya command, which are then printed out in the dialog box

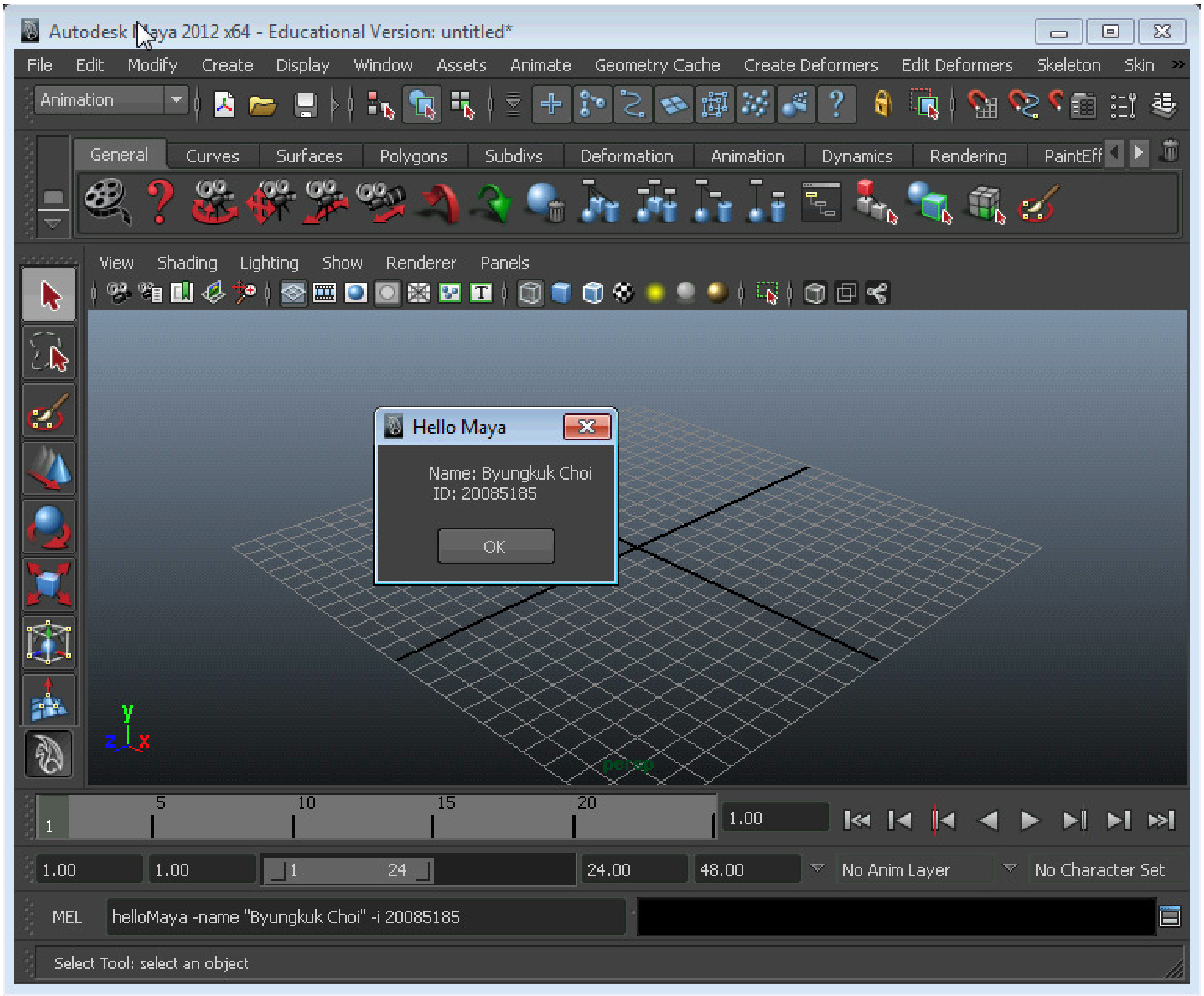
   Once written and compiled in Visual Studio, load your plugin into Maya by going to the *Window -> Settings/Preferences -> Plug-in Manager* window, browsing to the directory containing the compiled .mll file, and pressing Open. If the plugin loaded successfully, it will appear in the Plug-in Manager window with a check next to "Loaded".

4. Using the provided Python plugin framework, develop the Hello Maya Python Plugin. The python plugin can be loaded into Maya in the same way as the C++ plugin from the *Window -> Settings/Preferences -> Plug-in Manager* window. You can write the python plugin from either within the Maya script editor itself or another editor of your choosing.

   Functionality of the plugin should be the similar to the C++ version:

   a) Display a Dialog box entitled "Hello Maya Python" by typing the custom command "pyHelloMaya".

b) Provide arguments “name” and “id” to the command, which are then printed in the dialog box.



Hello Maya

Name: Byungkuk Choi
ID: 20085185

OK

MEL
```
helloMaya -name "Byungkuk Choi" -i 20085185
```

Select Tool: select an object

5. ZIP your .cpp, .h, .vcproj, .sln, .mll, and .py files and submit to Blackboard. For full credit, your plug-in must build without modifications for MSVC 2010. To grade your assignment, the grader will compile your source, load the helloMaya and pyHelloMaya plugins into Maya, and then run them from the MEL command line.

Appendix A

Maya C++ Plugin Framework (Python framework available on Blackboard)

Plug-in framework

```
#define EXPORT _declspec(dllexport)
// define EXPORT for exporting as .dll,
// (Do not need if you use ‘Maya PluginWizard‘)

// custom Maya command
class helloMaya: public MPxCommand
{
public:
      virtual MStatus doIt( const MArgList& args)
      {
            MStatus status;

            // <<<your code goes here>>>

            return status;
      }
      static void *creator() { return new helloMaya; }
};
// Initialize Plugin upon loading
EXPORT MStatus initializePlugin(MObject obj)
{
      MStatus stat;
      MFnPlugin plugin( obj, "CIS660", "1.0", "Any");
      stat = plugin.registerCommand("helloMaya", helloMaya::creator );
      if (!stat) stat.perror( "registerCommand failed" );
      return stat;
}

// Cleanup Plugin upon unloading
EXPORT MStatus uninitializePlugin(MObject obj)
{
      MStatus stat;
      MFnPlugin plugin(obj);
      stat = plugin.deregisterCommand("helloMaya");
      if(!stat) stat.perror( "deregisterCommand failed" );
      return stat;
}
```

If you are using the Maya Plugin Wizard, the wizard takes care of creating the initializePlugin, unitializePlugin and helloMaya::creator functions for you. For simple commands, it is unlikely that you will need to change the initialize and uninitialized functions.

Appendix B

Setting up Visual Studio 2010 for Maya Plugin Development

To set up the Visual Studio environment, you should download the Maya Plugin Wizard 2.0.1 from the Blackboard site.

Below are the instructions to set up the plugin wizard on a 64bit machine. (If you are running on a 32bit machine, instructions to set up the environment can be found in the README that comes with the plugin wizard.)

Maya Plug-in Wizard 2.0.1 (64 bit version)

The Microsoft Visual Studio 10.0 version of the Maya plug-in wizard for this release is contained in the MayaPluginWizard 2.0.1zip file. This wizard is HTML based and contains several files and directories that must be copied into the Microsoft Visual Studio 10.0 install. To install and run the Maya plug-in wizard do the following:

1. Unzip the MayaPluginWizard 2.0.1 zip file.
2. Copy the following files to the "C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\vcprojects" directory:
   MayaPluginWizard.vsdir
   MayaPluginWizard.vsz
   MayaPluginWizard.ico
3. Copy the "MayaPluginWizard" directory to "C:\Program Files\Microsoft Visual Studio 10.0\VC\VCWizards".
4. Start Microsoft Visual Studio 10.0 and invoke File -> New -> Project -> Visual C++ Projects and select MayaPluginWizard.
5. Enter a name and solution name and select the OK button.
6. Fill in the information for the "Plug-in setup" screen and then select the "Plug-in type" and "Included libraries" links to also enter the required information.
7. The project will be created and then the solution can be updated and built.
8. Select menu item Build -> Configuration Manager...
9. In the Active Solution Platform list box, select <New...>
10. In the new platform box, select "x64" and then close the window. You can now build the plug-in as 64 bit.

NOTES:
- At this time the Maya install process will not copy the wizard files into the appropriate Microsoft Visual Studio 10.0 directories. This process is manual.
- If you are working with a non-english installation of Microsoft Visual Studio 10.0, then you may have to alter the directories that the files are copied into above.

IF you are interested in setting it up manually, below are the steps:

STEP BY STEP PROCESS:

Below is the step by step procedure to manually setup the development environment using visual studio 2010. In project property settings:

Configuration

Configuration Properties -> Target Extension -> .mll
Configuration Type -> Dynamic Library (.dll)

C/C++

Preprocessor -> Preprocessor Definitions -> NT_PLUGIN
General -> Additional Include Directories -> [MayaInstallDir]\Include

Linker

General -> Additional Library Directories -> [MayaInstallDir]\lib
Input -> Additional Dependencies -> (You need to append Maya lib based on your plug-in)

- Foundation.lib;
- OpenMaya.lib;
- OpenMayaUI.lib;
- OpenMayaAnim.lib;
- OpenMayaFX.lib;
- OpenMayaRender.lib;
- Image.lib;
- opengl32.lib