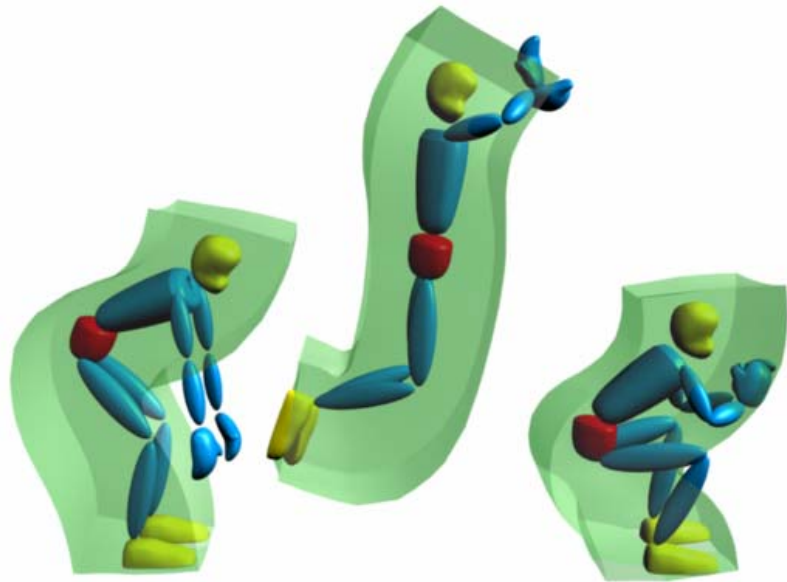


Playing with Maya thru MEL/API



Min Gyu Choi

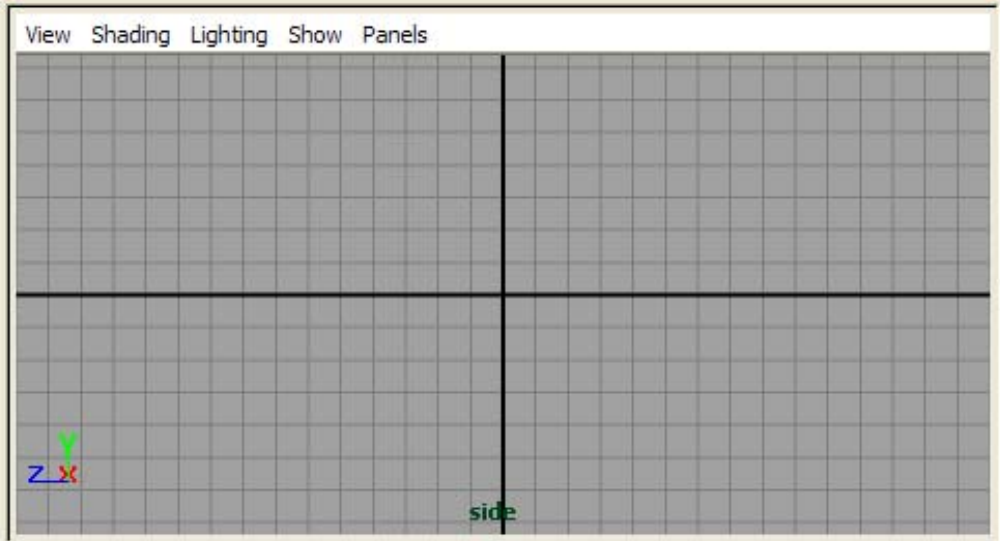
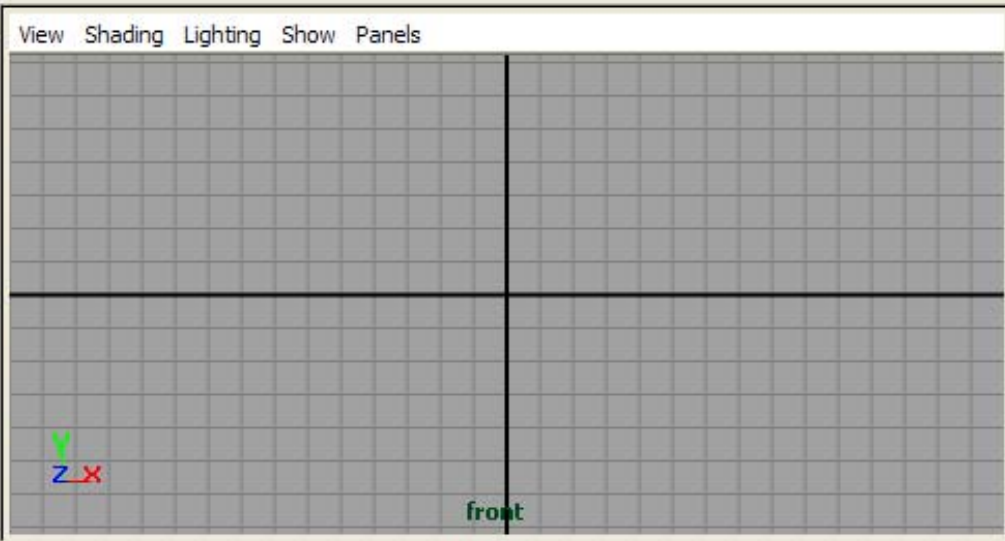
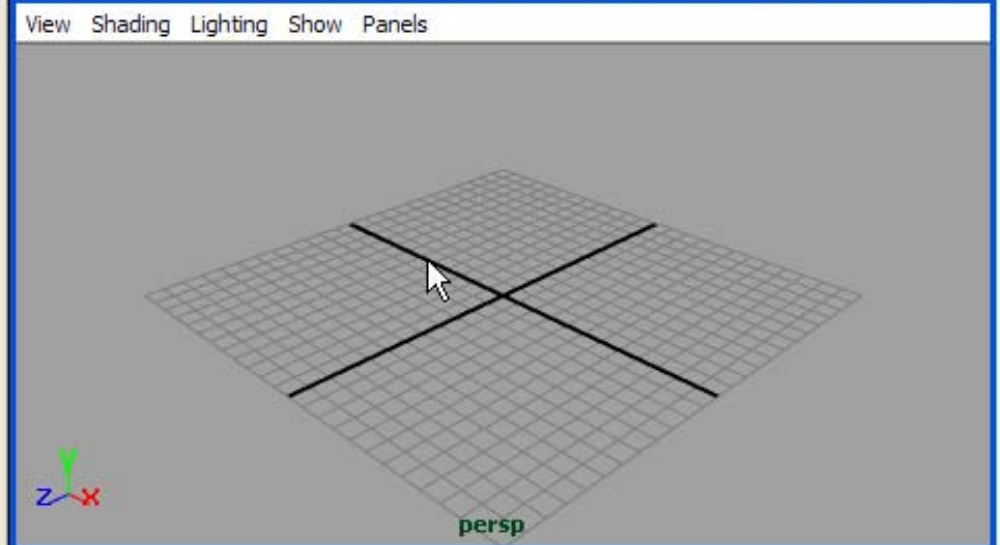
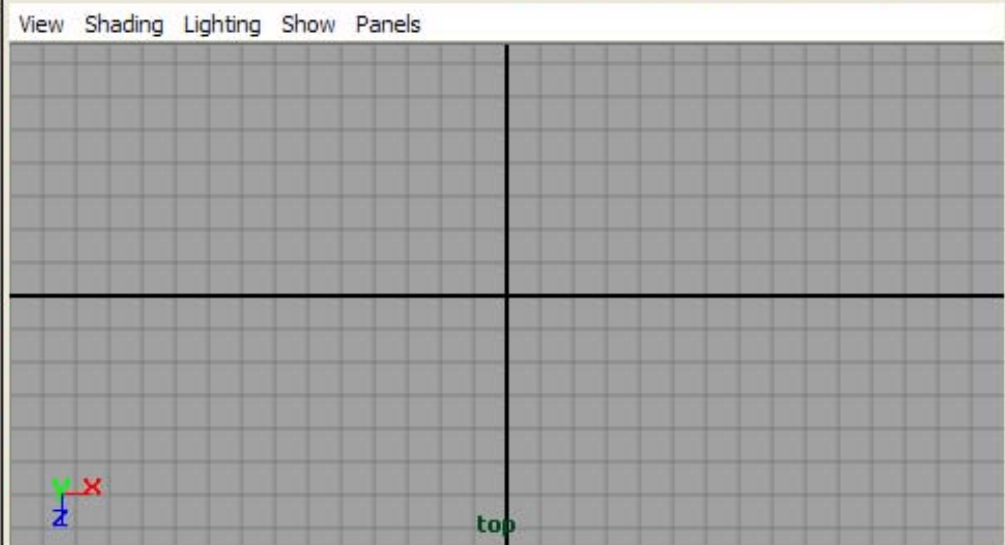
Kwangwoon University

Alias Maya

- **Alias|Wavefront Maya** is one of the greatest and most complex computer programs ever made.



Rendering





0 50 100 150 200 250 300 350 400 450 500 550 600 650 700 700.00 700.00 700.00 No Character Set

Select Tool: select an object

Alias Maya

- Maya has many features as default.



```
popupMenu -mm 1 -b 1 -p viewPanels  
-alt $modKeys[0] -ctrl $modKeys[1] -sh  
menuItem -rp "S" -l "Normals" -subMenu  
polyBrushMenuItem "NE" 0;  
polyBrushMenuItem "N" 2;
```

Maya Complete

Maya Unlimited



Alias Maya

- **However, Maya cannot do everything!**
 - You can “bend” Maya in your direction.

 - **Develop your own features.**
 - Automate and simplify tasks
 - Advanced macros and your own GUI

 - Add new features that are not incorporated
 - New file formats, object types and behaviors
 - New dynamics and animation components

 - e.g., [Maya Fur](#), [Live](#), [Cloth](#), [Hair](#), and [Fluid Effects](#)
-

Developing Maya Plug-Ins

□ Types of Maya plug-ins

- MEL (Maya Embedded Language) commands
- Nodes/commands, manipulators, contexts, locators

□ Maya API (Application Programmer Interface)

- Gives *almost* unlimited access to the internal Maya interface

□ You can also use MEL through Maya API.

Maya API

- **OpenMaya** for defining nodes/commands
 - Also for assembling them into a plug-in

 - **OpenMayaUI** for creating new user interface elements
 - Manipulators, contexts, and locators

 - **OpenMayaAnim** for animation
 - Including deformers and inverse kinematics

 - **OpenMayaFX** for dynamics

 - **OpenMayaRender** for performing rendering functions
-

Getting Started

To create a new plug-in

1. Start Visual Studio .NET
2. Invoke **File→New→Project** and select **Visual C++ Projects**
3. Select **MayaPlugInWizard**
4. Enter the solution name and location
5. Fill in the information
 - Plug-in setup, Plug-in type, Included libraries**

Refer to Appedix I

By the way, what to develop?

What to Develop?

No More “Hello World!”

Elastodynamic Deformation

□ Elastodynamic equation for FEM

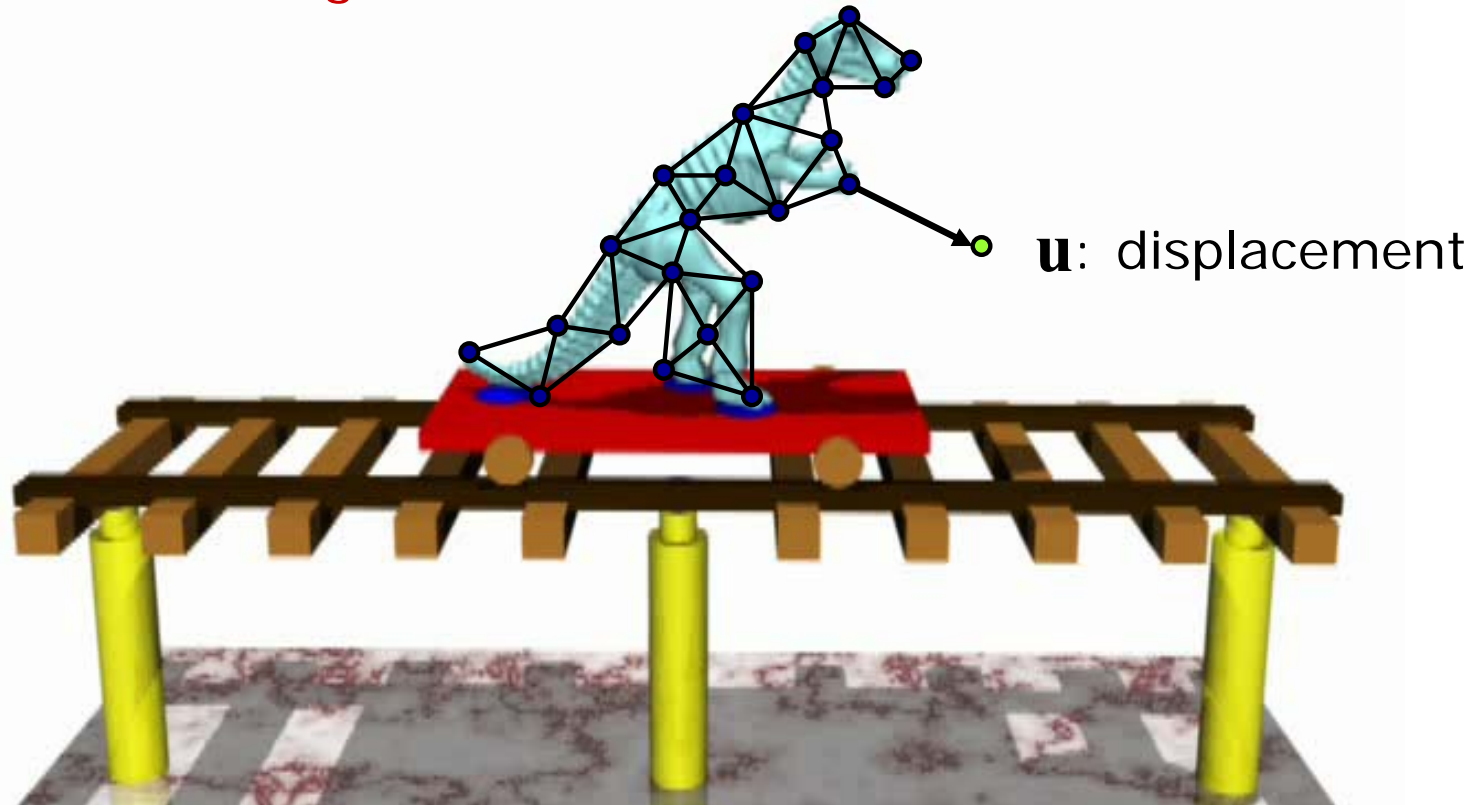
3n x 3n matrix, 3n x 1 vector

$$\underline{\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}(\mathbf{u}) = \mathbf{F}} \quad \longrightarrow \quad \underline{\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{F}}$$

Time-consuming!!!

Linearization

Not real-time!!!



Modal Analysis

[Pentland & Williams '89]

□ Elastodynamic equation for FEM

3n x 3n matrix, 3n x 1 vector

$$\underline{\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}(\mathbf{u}) = \mathbf{F}} \quad \xrightarrow{\text{Linearization}} \quad \underline{\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{F}}$$

Time-consuming!!! Not real-time!!!

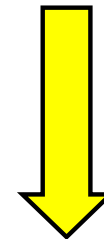
G. eigenvalue problem

$$\mathbf{M}\Phi_i = \lambda_i \mathbf{K}\Phi_i$$



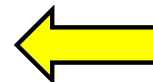
Modal basis

$$\mathbf{u}(t) = \Phi \mathbf{q}(t)$$



$$\underline{m_i \ddot{q}_i + c_i \dot{q}_i + k_i q_i = Q_i}$$

Real-time performance!!!



$$\mathbf{M}_q \ddot{\mathbf{q}} + \mathbf{C}_q \dot{\mathbf{q}} + \mathbf{K}_q \mathbf{q} = \mathbf{Q}$$

Diagonal matrices

There can be linearization artifacts!!!

Modal Warping

[IEEE TVCG 2005]

- Modal analysis in *local coordinate frames*

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}(\mathbf{u}) = \mathbf{F} \quad \xrightarrow{\text{Linearization}} \quad \mathbf{M}\ddot{\mathbf{u}}^L + \mathbf{C}\dot{\mathbf{u}}^L + \mathbf{K}\mathbf{u}^L = \mathbf{R}^T \mathbf{F}$$

Linearization

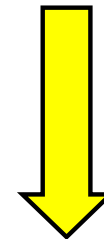
G. eigenvalue problem

$$\mathbf{M}\Phi_i = \lambda_i \mathbf{K}\Phi_i$$

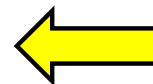


Modal basis

$$\mathbf{u}^L(t) = \Phi \mathbf{q}(t)$$



$$\mathbf{u}^k = \int_0^{t^k} \mathbf{R}(t) \Phi \dot{\mathbf{q}}(t) dt$$



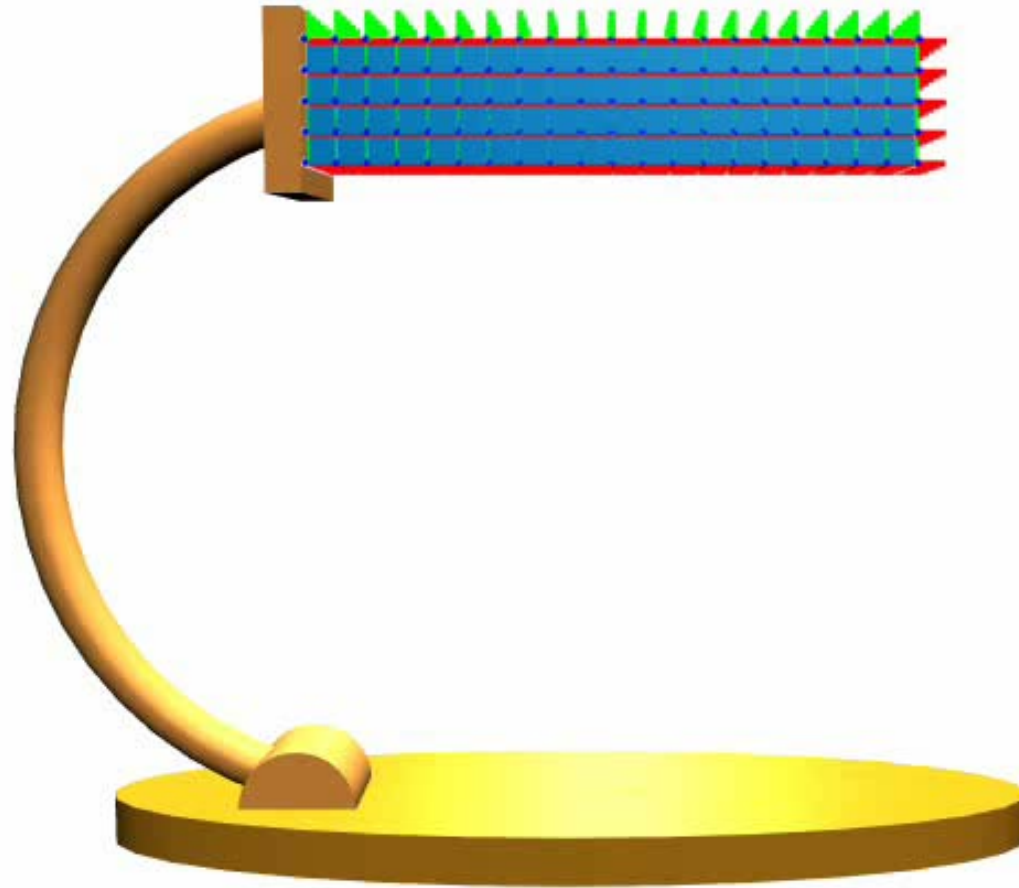
$$\mathbf{M}_q \ddot{\mathbf{q}} + \mathbf{C}_q \dot{\mathbf{q}} + \mathbf{K}_q \mathbf{q} = \mathbf{Q}$$

Modal warping

Real-time performance without linearization artifacts!!!

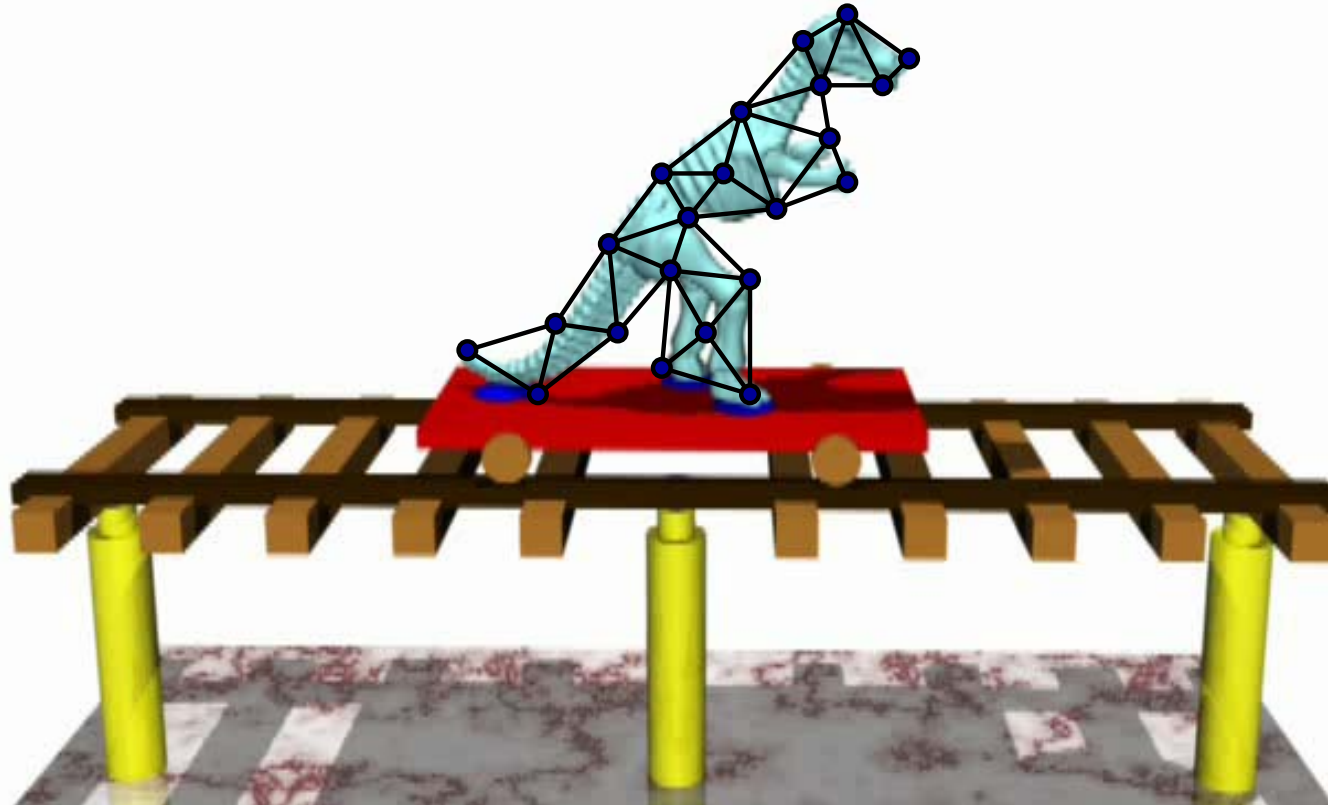
Modal Warping

Gravity = -9.8

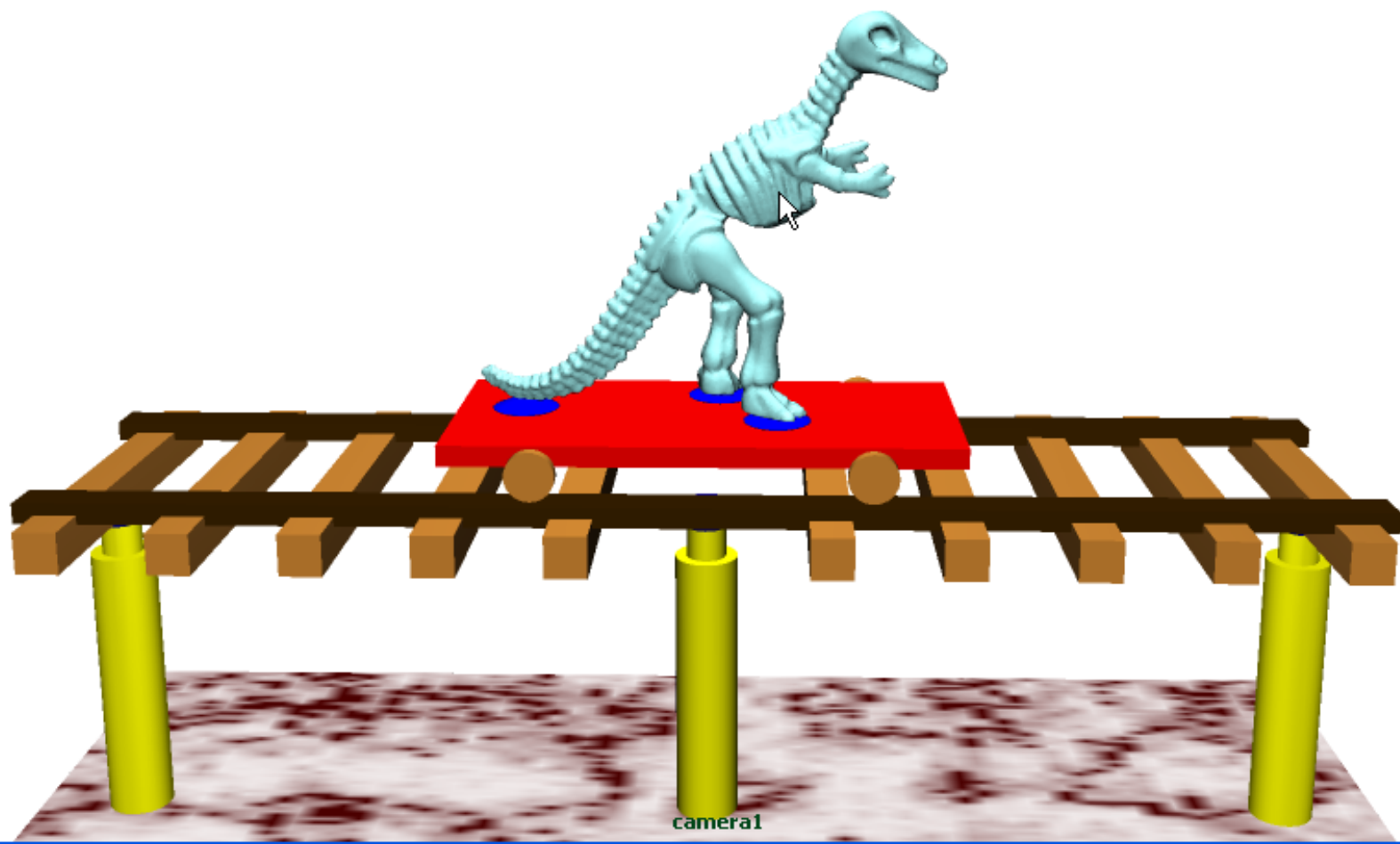


The First Plug-In

- How to obtain volume meshes for FEM?
 - **NETGEN** converts polygonal meshes into volume meshes.
- ➔ How to obtain polygonal meshes?



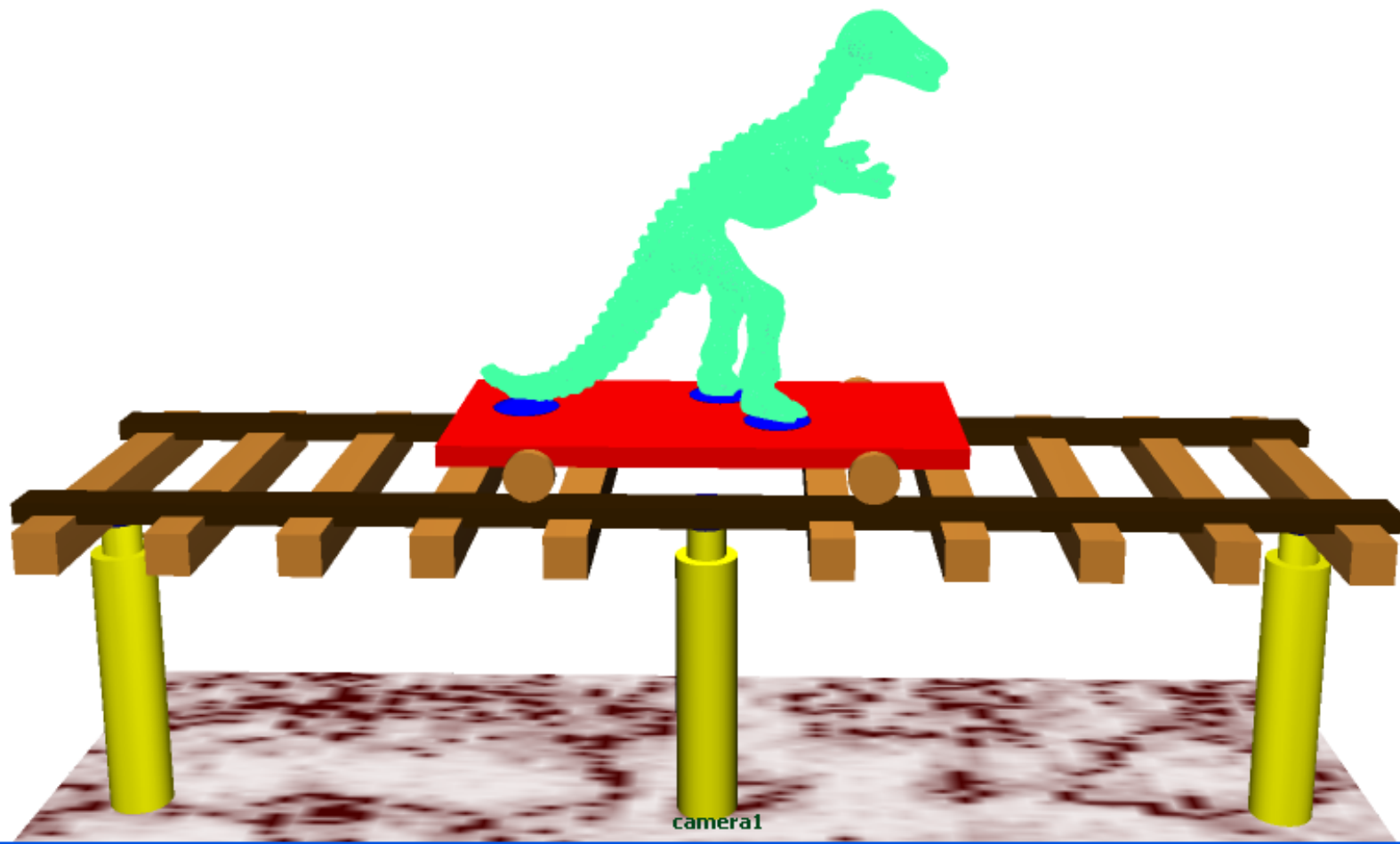
Verts: 28210 0 0
Edges: 84456 0 0
Faces: 56276 0 0
Tris: 56360 0 0
UVs: 196 0 0



Timeline and playback controls. The timeline shows a scale from 0 to 300. Playback buttons include stop, play, and fast forward. A 'No Character Set' button is visible on the right.

Tool shelf with a selection tool icon highlighted by a red box and a red arrow pointing to it. The word "Click" is written in red text next to the arrow.

Verts: 28210 28098 0
Edges: 84456 84288 0
Faces: 56276 56192 0
Tris: 56360 56192 0
UVs: 196 0 0



Maya interface bottom bar including a timeline with markers at 20, 40, 60, 80, 100, 120, 140, 160, 180, 200, 220, 240, 260, 280, and 300. It also features playback controls (stop, play, etc.), a character set dropdown menu currently set to "No Character Set", and other utility icons.

Verts: 28210 28098 0
Edges: 84456 84288 0
Faces: 56276 56192 0
Tris: 56360 56192 0
UVs: 196 0 0

Export

D:\WORK\PROJECTS\scenes\DATA

Look in: DATA

BACKUP	blockShape.fep	dinosaurShape.fep
data	blockShape.node	dinosaurShape.stl
attributes.txt	canShape.ele	flubberGirlFineShape.fep
ballShape.fep	canShape.face	flubberGirlFineShape.stl
ballShape.stl	canShape.fep	flubberGirlShape.fep
barShape.fep	canShape.node	flubberGirlShape.stl
barShape.stl	clothShape.ele	flubberManShape.fep
BigGuyPotbellyShape.fep	clothShape.face	flubberManShape.stl
BigGuyPotbellyShape.stl	clothShape.fep	flubberShape.fep
BigGuyPotbellyShape.tag	clothShape.node	flubberShape.stl
BigGuyShape.fep	cubeShape.fep	jumperShape.fep
BigGuyShape.stl	cubeShape.fep.backup	jumperShape.stl
BigGuyShape.tag	cubeShape.stl	jumperShape.tag
blockShape.ele	cubeShape.stl.backup	longBlockShape.ele
blockShape.face	data.zip	longBlockShape.face

File name: dinosaurShape.stl

Files of type: Best Guess (*.*)

Export Cancel Set Project... Options...



camera1

Timeline: 0 20 40 60 80 100 120 140 160 180 200 220 240 260 280 300.00

0.00 0.00 300.00 300.00 300.00 No Character Set

Shelf area with a button icon highlighted by a red box. A red arrow points from the text "What is this?" to the icon.

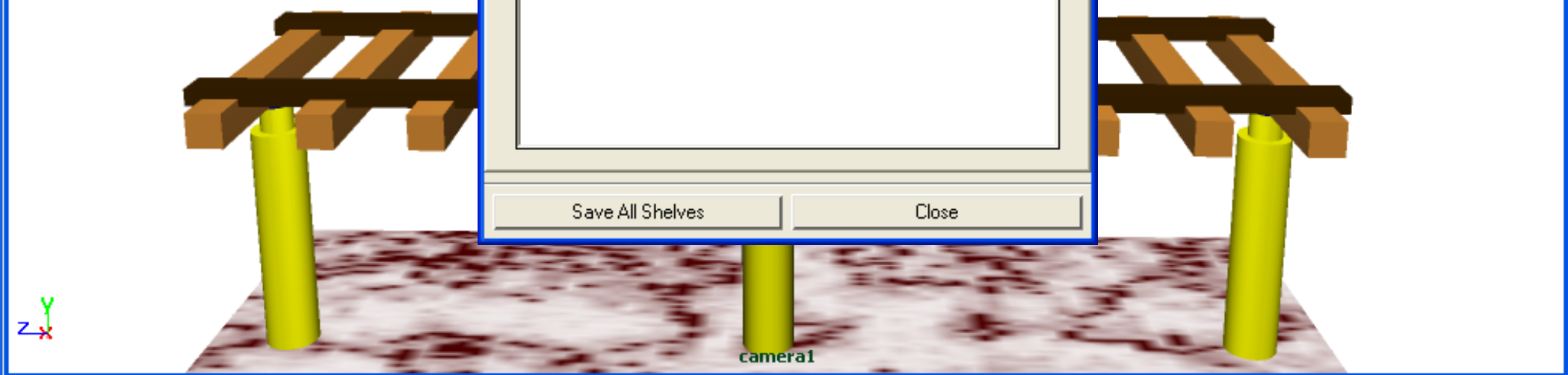
What is this?

Verts: 28210 0 0
Edges: 84456 0 0
Faces: 56276 0 0
Tris: 56360 0 0
UVs: 196 0 0

Click and then choose "Shelf Editor..."

MEL command, ExportMesh

Shelves dialog box with tabs: Shelves, Shelf Contents, Edit Commands. The text "ExportMesh;" is entered in the command field and circled in blue. A blue arrow points from the text "MEL command, ExportMesh" to the circled text.



Maya interface bottom bar including a timeline (0 to 300), playback controls, and a status bar showing "No Character Set".

MEL Script Associated with a Shelf Item

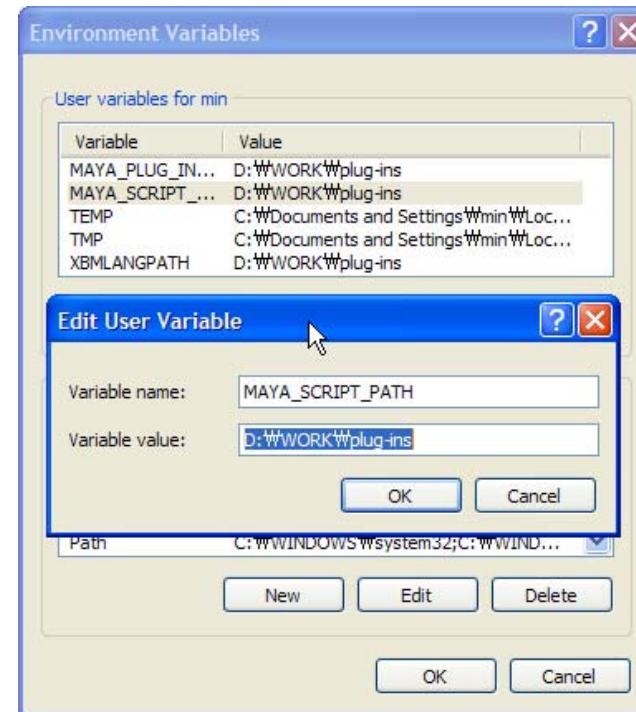
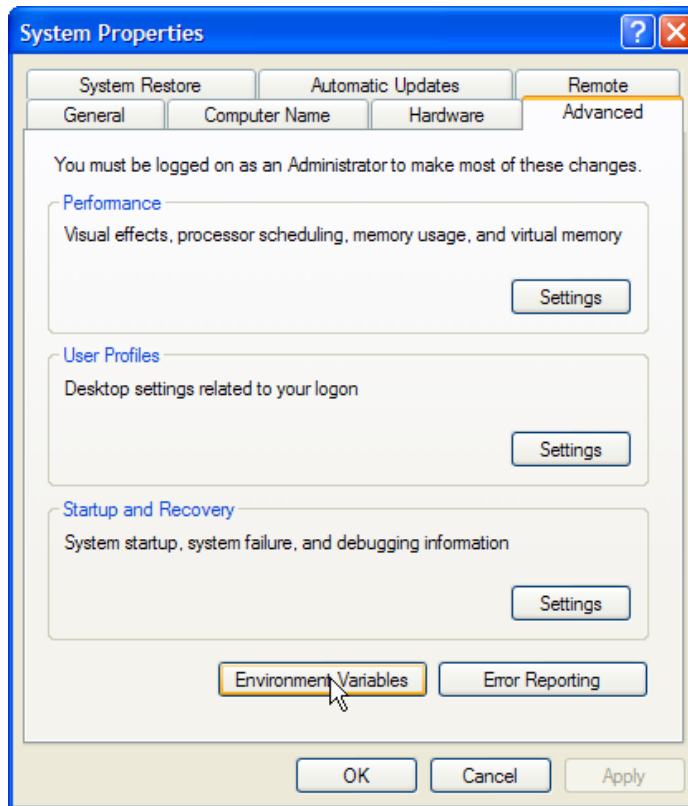
- **ExportMesh** defined in the File “**ExportMesh.mel**”

```
global proc string export(string $filename, string $filetype )
{
    meshExport $filename; // We will develop this command using Maya API.
    return $filename;
}

global proc ExportMesh()
{
    fileBrowserDialog -m 1 -fc "export" -an "Export";
}
```

MEL Script Path

- How to locate the script file “ExportMesh.mel”?
 - Maya searches `MAYA_SCRIPT_PATH` first.



Command for Polygonal Mesh Export

The screenshot displays the Microsoft Visual C++ IDE with the following components:

- Code Editor:** Shows the `meshExport.cpp` file with the following code:

```
#include <fstream.h>
#include <maya/MPlug.h>
#include <maya/MFnPlugin.h>
#include <maya/MArgDatabase.h>
#include <maya/MDagPath.h>
#include <maya/MDagPathArray.h>
#include <maya/MFnMesh.h>
#include <maya/MIntArray.h>
#include <maya/MSelectionList.h>
#include <maya/MGlobal.h>
#include "meshExport.h"

// Initialize and Uninitialize Plugin
//

MStatus
initializePlugin(MObject obj)
{
    MStatus status;
    MFnPlugin plugin(obj, "Maya Dime", "1.0", "Any");

    status = plugin.registerCommand( "meshExport", meshExport::creator, meshExport::newSyntax );
    if (!status) { status.error( "registerCommand dimeeExport" ); return status; }

    return status;
}

MStatus
uninitializePlugin(MObject obj)
{
    MStatus status;
    MFnPlugin plugin(obj);
```
- Solution Explorer:** Shows the project structure for 'MeshExport' with files `meshExport.cpp` and `meshExport.h`. A red arrow points from the `#include "meshExport.h"` line in the code editor to the `meshExport.cpp` file in the Solution Explorer.
- Properties Window:** Shows the file properties for `meshExport.cpp`, including its name, content type (False), file type (C/C++ Code), full path, and relative path.

Command for Polygonal Mesh Export

```
#include <maya/MArgList.h>
#include <maya/MDagPath.h>
#include <maya/MPxCommand.h>
#include <maya/MString.h>
#include <maya/MSyntax.h>

// Export
//
class meshExport : public MPxCommand
{
public:
    MString fileName;

public:
    meshExport() {};
    ~meshExport() {};

    MDagPath    getSelectedObject(int index = 0);

    MStatus meshExportPolygonMeshSTL(MFnMesh&);
    MStatus meshExportPolygonMeshPOLY(MFnMesh&);
    MStatus meshExportPolygonMeshSMESH(MFnMesh&);

    MStatus doIt(const MArgList& args );

    static void*    creator() { return new meshExport(); }

    static MSyntax newSyntax();

private:
    MStatus parseArgs(const MArgList& args);
};
```

meshExport.h

Command for Polygonal Mesh Export

```
meshExport.cpp

MDagPath
meshExport::getSelectedObject(int index)
{
    MSelectionList list;
    MGlobal::getActiveSelectionList(list);

    MDagPath dagPath;
    list.getDagPath(index, dagPath);

    return dagPath;
}

MStatus
meshExport::doIt(const MArgList& args)
{
    MStatus status = parseArgs(args);
    if (status != MStatus::kSuccess) return status;

    MDagPath path = getSelectedObject();
    if (!path.isValid()) return MStatus::kFailure;

    MFnMesh mesh(path, &status);
    if (status != MStatus::kSuccess) { status.perror("meshExport"); return status; }

    meshExportPolygonMeshSTL(mesh); // NETGEN
    //meshExportPolygonMeshPOLY(mesh); // TETGEN
    //meshExportPolygonMeshSMESH(mesh); // GRUMMP

    return status;
}
```

DagPath?



Command for Polygonal Mesh Export

```
meshExport.cpp
MStatus
meshExport::meshExportPolygonMeshSTL(MFnMesh& mesh)
{
    MVector    normal;
    MIntArray  vertexList;
    MPoint     pos;

    ofstream  os(fileName.asChar());
    if (!os.is_open()) return MStatus::kFailure;

    os << "solid" << endl;
    for (int i = 0; i < mesh.numPolygons(); i++) // For all polygons in the mesh
    {
        mesh.getPolygonNormal(i, normal, MSpace::kWorld);
        os << " facet normal " << normal[0] << " " << normal[1] << " " << normal[2] << endl;

        os << "   outer loop" << endl;
        mesh.getPolygonVertices(i, vertexList);
        for (unsigned j = 0; j < vertexList.length(); j++) // For all vertices in the polygon
        {
            mesh.getPoint(vertexList[j], pos, MSpace::kWorld);
            os << "       vertex " << pos[0] << " " << pos[1] << " " << pos[2] << endl;
        }
        os << "   endloop" << endl << " endfacet" << endl;
    }
    os << "endsolid" << endl;
    os.close();

    return MStatus::kSuccess;
}
```

Press F7 to compile and Link!

Verts: 28210 0 0
Edges: 84456 0 0
Faces: 56276 0 0
Tris: 56360 0 0
UVs: 196 0 0

- General Editors >
- Rendering Editors >
- Animation Editors >
- Relationship Editors >
- Settings/Preferences >
 - Preferences...
 - Tool Settings...
 - Performance Settings...
 - Hotkeys...
 - Colors...
 - Marking Menus...
 - Shelves...
 - Panels...
 - Plug-in Manager...
- Attribute Editor...
- Outliner...
- Hypergraph...
- Paint Effects...
- UV Texture Editor...
- Playblast...
- View Arrangement >
- Saved Layouts >
- Save Current Layout...
- Frame All in All Views A
- Frame Selection in All Views F
- Minimize Application
- Raise Main Window
- Raise Application Windows

MAYA_PLUG_IN_PATH

Plug-in Manager

Help

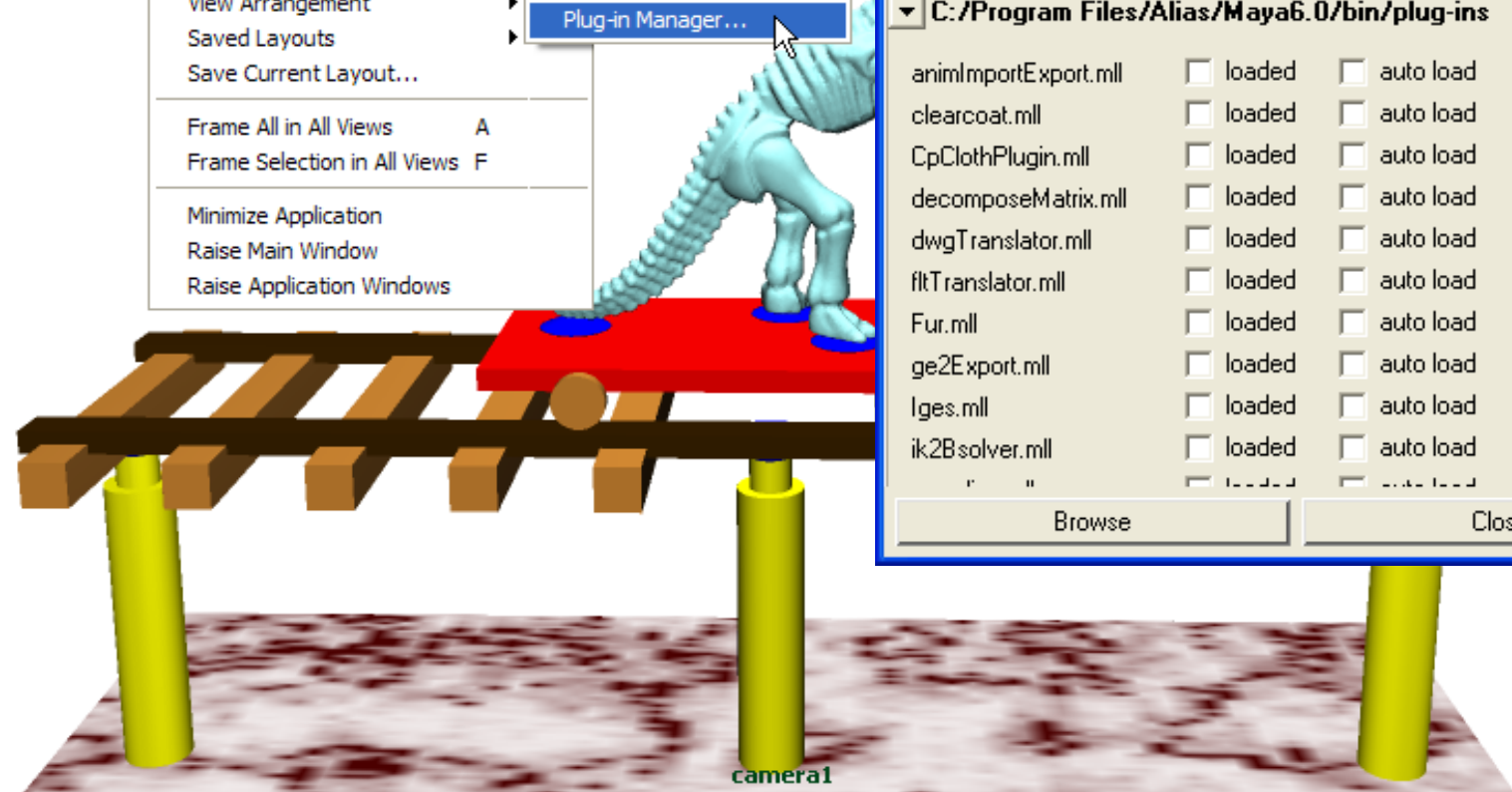
D:/WORK/plug-ins

mayaDime.mll	<input checked="" type="checkbox"/> loaded	<input type="checkbox"/> auto load	
mayaDime4Constant.mll	<input type="checkbox"/> loaded	<input type="checkbox"/> auto load	
meshExport.mll	<input checked="" type="checkbox"/> loaded	<input type="checkbox"/> auto load	

C:/Program Files/Alias/Maya6.0/bin/plug-ins

animImportExport.mll	<input type="checkbox"/> loaded	<input type="checkbox"/> auto load	
clearcoat.mll	<input type="checkbox"/> loaded	<input type="checkbox"/> auto load	
CpClothPlugin.mll	<input type="checkbox"/> loaded	<input type="checkbox"/> auto load	
decomposeMatrix.mll	<input type="checkbox"/> loaded	<input type="checkbox"/> auto load	
dwgTranslator.mll	<input type="checkbox"/> loaded	<input type="checkbox"/> auto load	
fltTranslator.mll	<input type="checkbox"/> loaded	<input type="checkbox"/> auto load	
Fur.mll	<input type="checkbox"/> loaded	<input type="checkbox"/> auto load	
ge2Export.mll	<input type="checkbox"/> loaded	<input type="checkbox"/> auto load	
Iges.mll	<input type="checkbox"/> loaded	<input type="checkbox"/> auto load	
ik2Bsolver.mll	<input type="checkbox"/> loaded	<input type="checkbox"/> auto load	

Browse Close



0 20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 0.00

0.00 0.00 0 300 300.00 300.00 No Character Set

loadPlugin meshExport.mll;

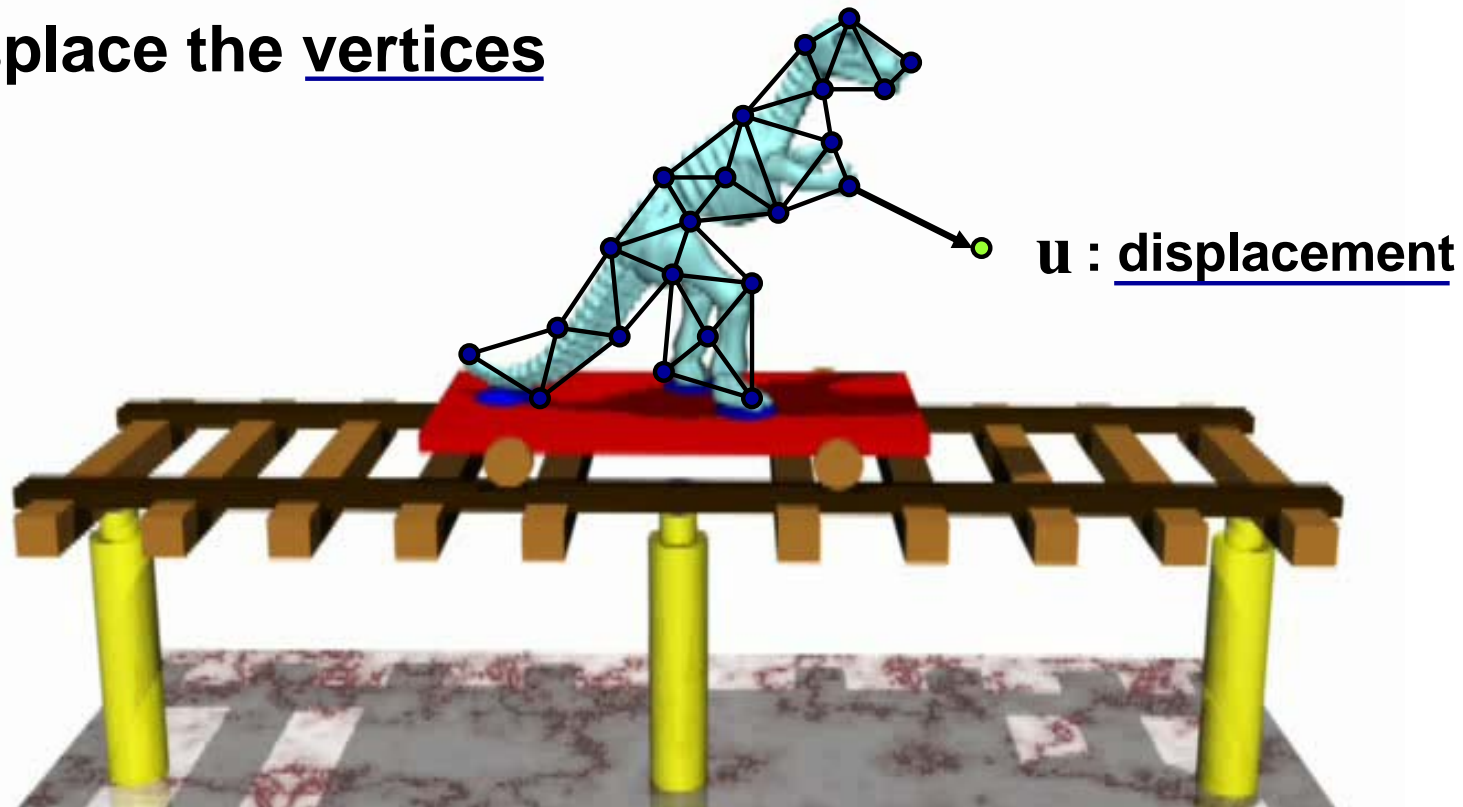
Result: meshExport

The Second Plug-In

- Integrate the elastodynamic equation through time

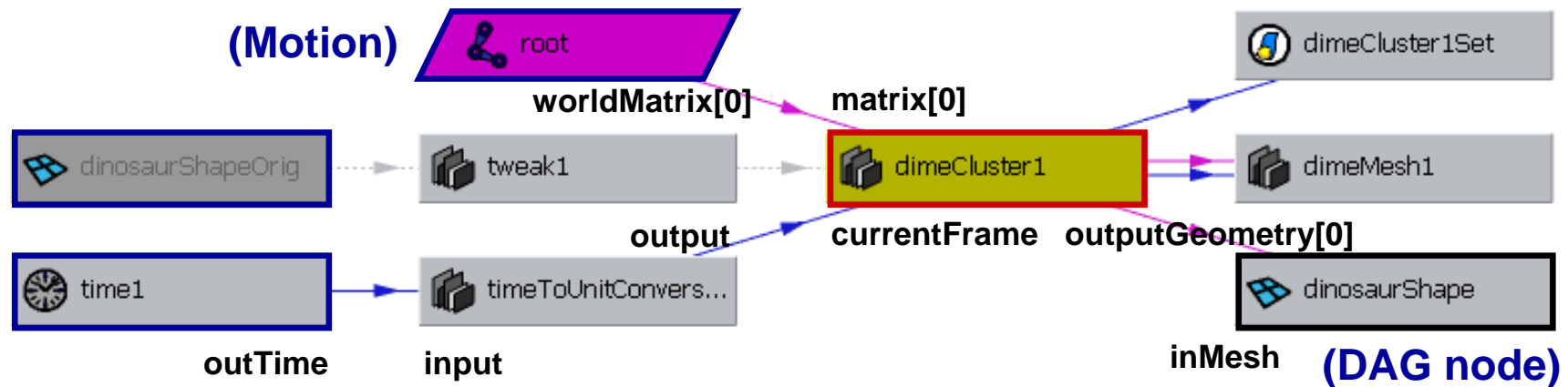
$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}(\mathbf{u}) = \mathbf{F} \leftarrow \text{Excitation (external motion)}$$

- Displace the vertices



Solid Deformer

- **The dependency graph is the heart of Maya.**
 - DG nodes are used for almost everything in Maya.
 - e.g., model creation, deformation, animation, simulation, ...



Connections

```
time1.outTime → dimeCluster1.CurrentFrame
root.worldMatrix[0] → dimeCluster1.matrix[0]
dimeCluster1.outputGeometry[0] → dinosaurShape.inMesh
```

Plugs

Registration of the Deformer Node

```
// initializePlugin and uninitializePlugin
//
MStatus
initializePlugin(MObject obj)
{
    MStatus status;
    MFnPlugin pn(obj, "Maya Dime", "1.0", "Any");

    // Strand
    status = pn.registerShape("dimeStrandShape", DStrandShape::id, DStrandShape::creator,
                             DStrandShape::initialize, DStrandShapeUI::creator);
    if (!status) { status.perror("registerNode DStrand"); return status; }

    // Solid Deformer
    status = pn.registerNode("dimeCluster", DSolidDeformer::id, DSolidDeformer::creator,
                             DSolidDeformer::initialize, MPxNode::kDeformerNode);
    if (!status) { status.perror("registerNode DSolidDeformer"); return status; }

    // Shell Deformer
    status = pn.registerNode("dimeShell", DShellDeformer::id, DShellDeformer::creator,
                             DShellDeformer::initialize, MPxNode::kDeformerNode);
    if (!status) { status.perror("registerNode DShellDeformer"); return status; }

    // Control Mesh
    status = pn.registerShape("dimeMesh", DControlMesh::id, DControlMesh::creator,
                             DControlMesh::initialize, DControlMeshUI::creator);
    if (!status) { status.perror("registerShape DControlMesh"); return status; }

    // Constraint
    status = pn.registerNode("dimeConstraint", DConstraint::id,
```

Deformer Node

```
class DSolidDeformer : public MPxDeformerNode
{
public:
    static MTypeId id;

    MObject currentFramePlug;
    MObject matrixPlug;
    MObject rigidityPlug;
    // ...

public:
    DSolidDeformer();
    ~DSolidDeformer();

    static void* creator();
    static MStatus initialize();
    virtual MStatus deform(MDataBlock&, MItGeometry&,
                           const MMatrix&, unsigned);
    // ...
};
```

dSolidDeformer.h

} Plugs

Deformer Node

- This method is called once after the plug-in is loaded.
 - Define the inputs and outputs of the node

```
MStatus
DSolidDeformer::initialize()
{
    currentFramePlug = nAttr.create("currentFrame", "cf",
                                    MFnNumericData::kLong);

    nAttr.setDefault(0);
    nAttr.setHidden(true);
    MStatus status = addAttribute(currentFramePlug);
    if (!status){ status.perror("addAttr()"); return status; }
    // ...

    attributeAffects(currentFramePlug, outputGeom);
    // ...
}
```

Deformer Node

```
MStatus  
DSolidDeformer::deform(MDataBlock& block, MItGeometry&  
                        const MMatrix&, unsigned)
```

```
{
```

```
    MStatus      status;  
    MDataHandle inputData;  
    inputData = block.inputValue(currentFramePlug, &status);  
    int currentFrame = inputData.asLong();  
    // ...
```

Input



```
    // computes displacedVertices ...
```

```
    MArrayDataHandle outputData;  
    outputData = block.outputArrayValue(outputGeom, &status);  
    MDataHandle outMesh = outputData.outputValue(&status);  
    MFnMesh      mesh(outMesh.asMesh(), &status);  
    mesh.setPoints(displacedVertices, Mspace::kWord);
```

Output

```
    return status;
```

```
}
```

Creating Deformer Nodes

- ❑ **Press F7 to compile and link**
 - You have to do many things ...

- ❑ **Load the plug-in**

- ❑ **Create a deformer node, and then connect the plugs**

```
deformer -type dimeCluster;  
  
connect    time1.outTime          dimeCluster1.currentFrame;  
connect    root.worldMatrix[0]    dimeCluster1.matrix[0];
```

- This can be automated by MEL and shelf items!
-

Verts: 28210 0 0
Edges: 84456 0 0
Faces: 56276 0 0
Tris: 56360 0 0

How to change simulation constants?

UVs:

Attribute Editor: dinosaur

List Selected Focus Attributes Help

dinosaur | dinosaurShape | dinosaurShapeOrig | dimeCluster1 | tweak1 | lambert3

dimeCluster: dimeCluster1

Focus Presets

General Properties

Display

Material Properties

Dynamic Simulation

Use Gravity Use Viscosity

Gravity 0.000 0.000 0.000

Damping Factor 0.200

Integration Scheme Implicit Integration

Theta 1.000

Playback Speed Real-time [30 fps]

Num Time Step 0

Start Frame 0

Acceleration Scale -1.500

Disable Evaluation Use Constraint

Simulation Scheme Modal Analysis

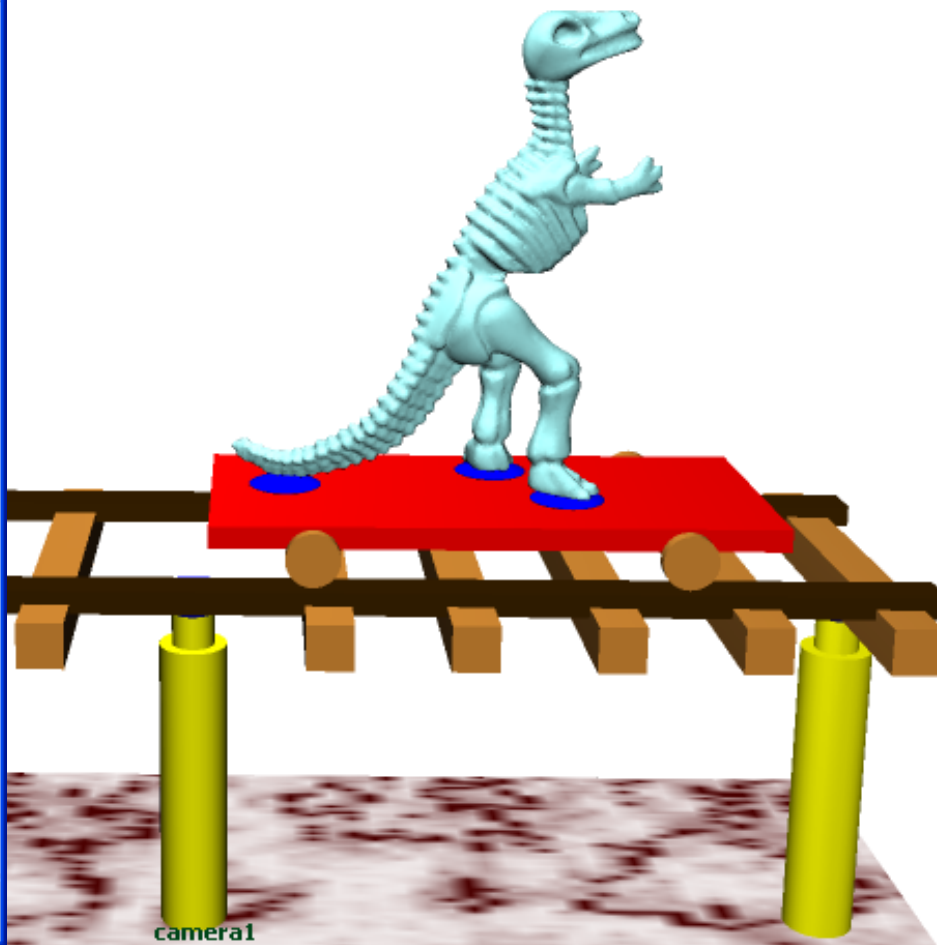
Warping Linear Constraint

Coriolis' Effect

Initial Setting

Extra Attributes

Select Load Attributes Copy Tab Close



0 20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 300.00 300.00

16.00

0.00 0.00 0 300.00 300.00

No Character Set

Result: 1

Customizing Attribute Editor

```
global proc AEdimeClusterTemplate( string $nodeName )
{
    // Put our attributes into a scrolled layout field
    editorTemplate -beginScrollLayout;

    editorTemplate -beginLayout "Dynamic Simulation" -collapse f
    editorTemplate -addControl "useGravity" "checkGravity";
    editorTemplate -addControl "useViscosity";
    editorTemplate -addControl "gravity";
    editorTemplate -addControl "dampingFactor";
    editorTemplate -addSeparator;

    editorTemplate -callCustom "dimeIntegratorNew"
                        "dimeIntegratorReplace"
                        "integrator";

    editorTemplate -suppress "integrator";
    editorTemplate -addControl "theta";

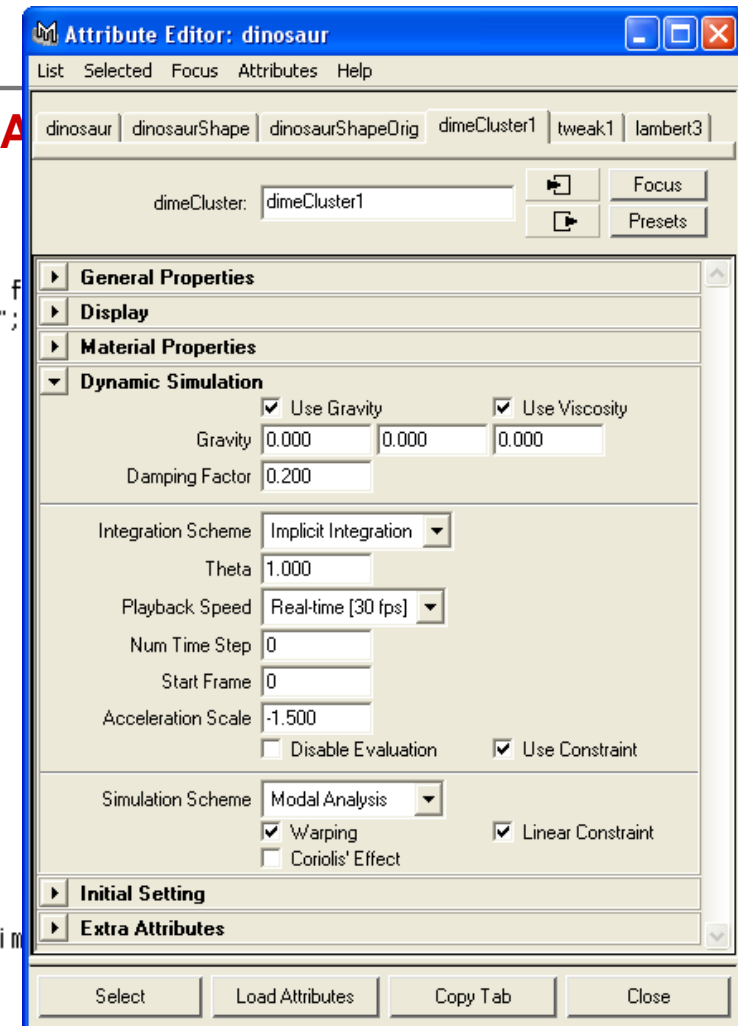
    editorTemplate -callCustom "dimeFPSNew"
                        "dimeFPSReplace"
                        "FPS";

    editorTemplate -suppress "FPS";

    editorTemplate -addControl "numTimeStep";
    editorTemplate -addControl "startFrame";

    editorTemplate -l "Acceleration Scale" -addControl "sim
    editorTemplate -addControl "disableEvaluation";
    editorTemplate -addControl "useConstraint";

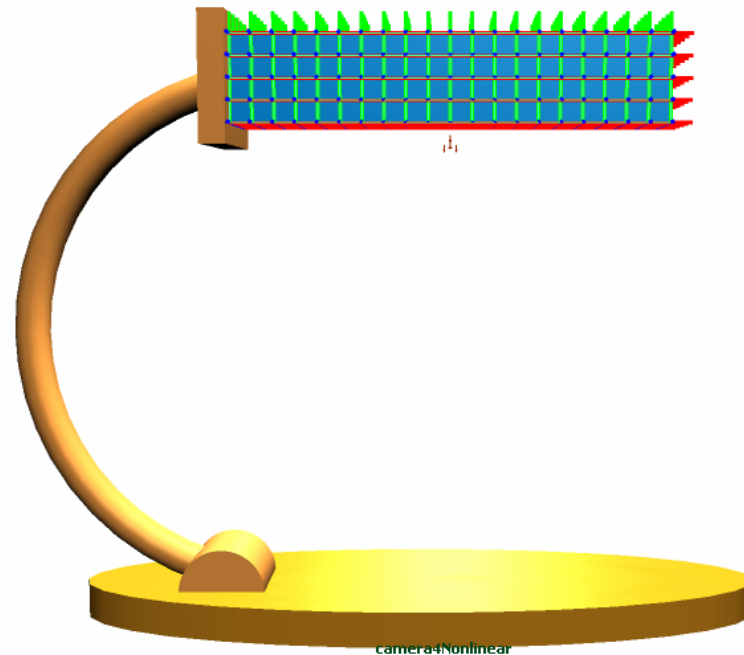
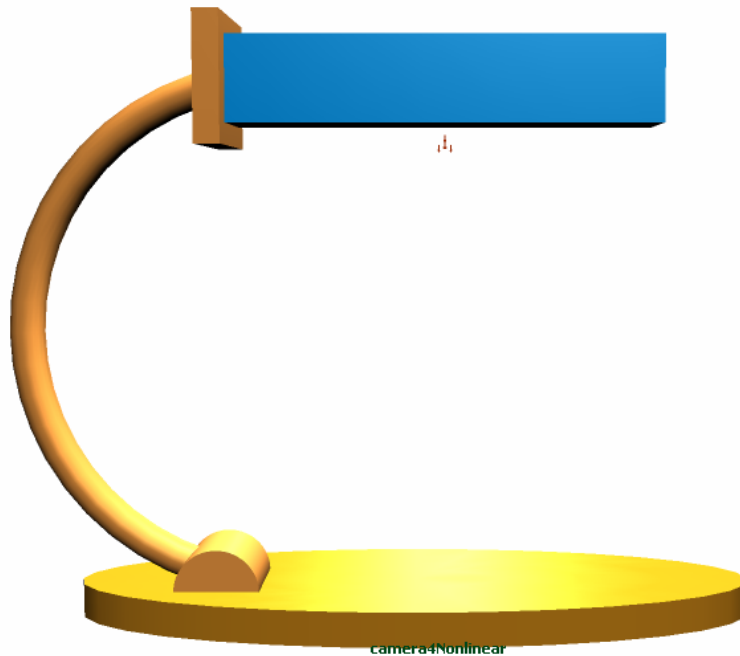
    editorTemplate -suppress "manipulateAll";
```



AEdimeClusterTemplate.mel should be located in **MAYA_SCRIPT_PATH**.

Additional DAG Nodes for Visualization

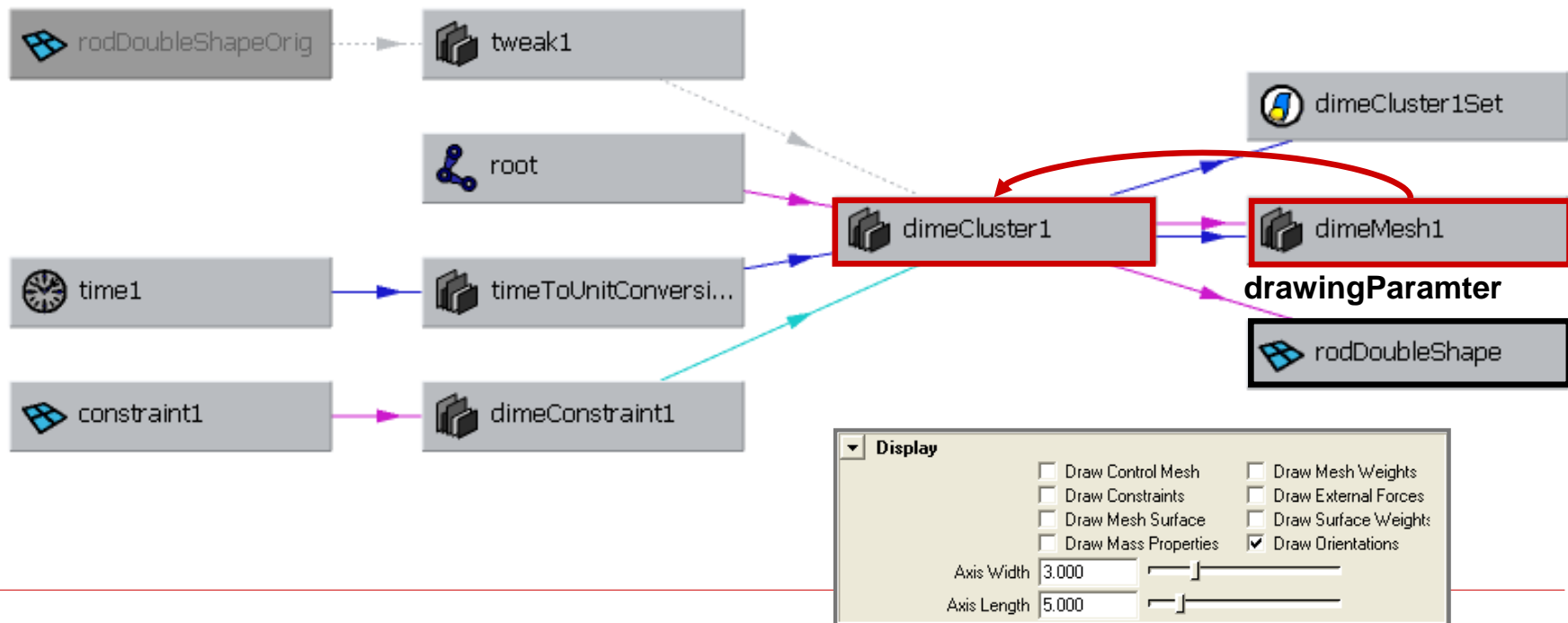
- To visualize useful information
 - We need to develop additional DAG nodes.



With local axes at the mesh nodes

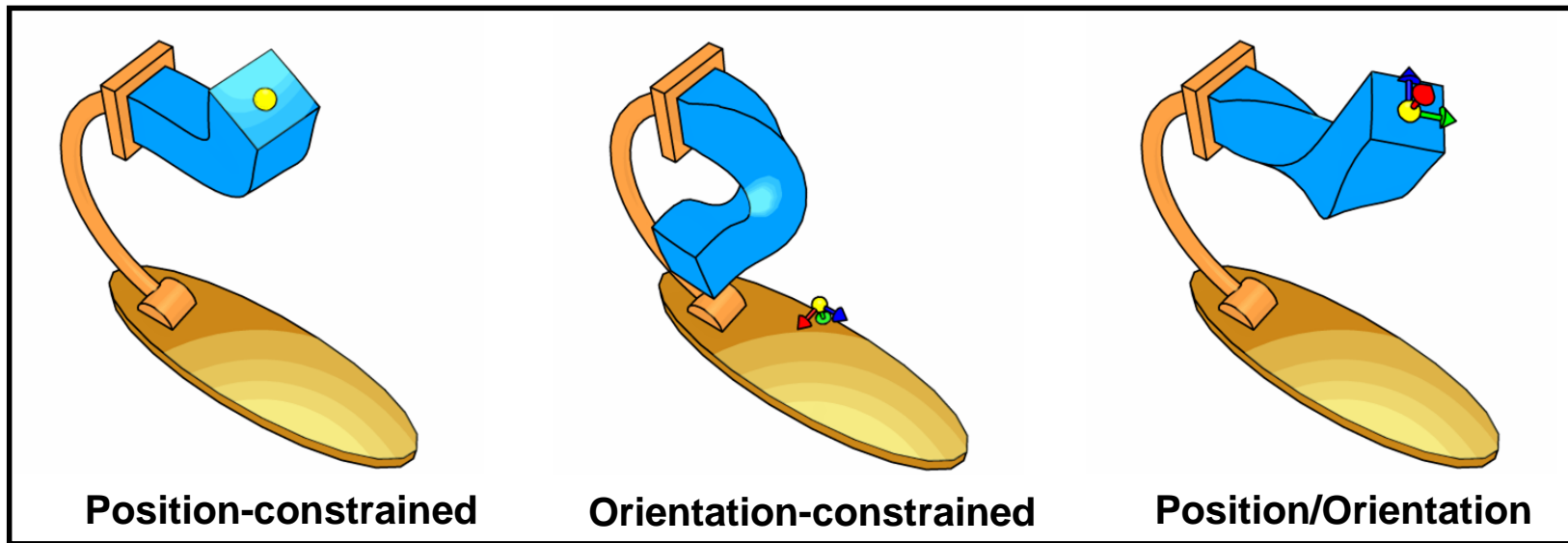
Additional DAG Nodes for Visualization

- **Derived node of MPxSurfaceShape**
 - This node gets connection to the deformer node though backward traversal of the connected plug.
 - (MPxSurfaceShapeUI supports OpenGL for H/W rendering)



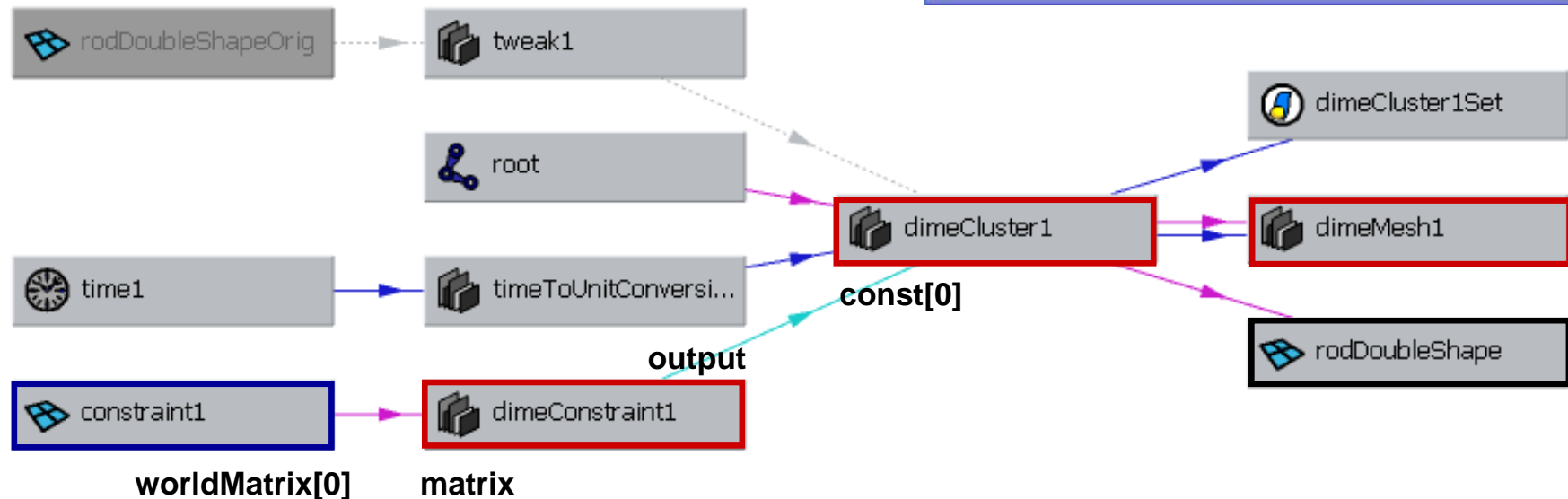
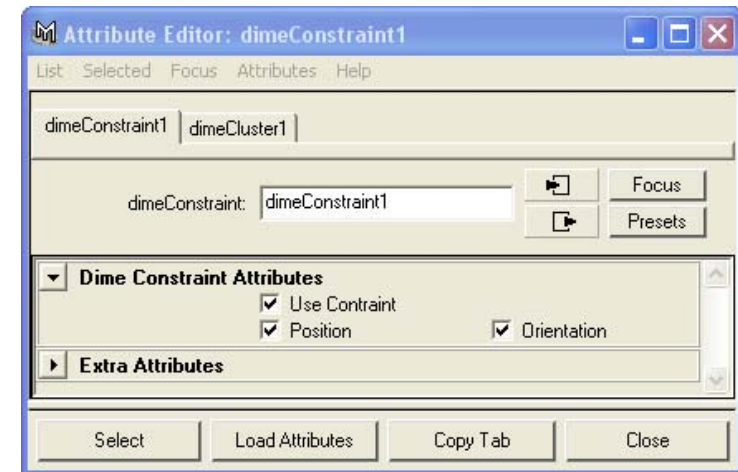
Additional DG Nodes for Constraints

- **To specify the manipulation constraints**
 - We need to develop additional dependency nodes.

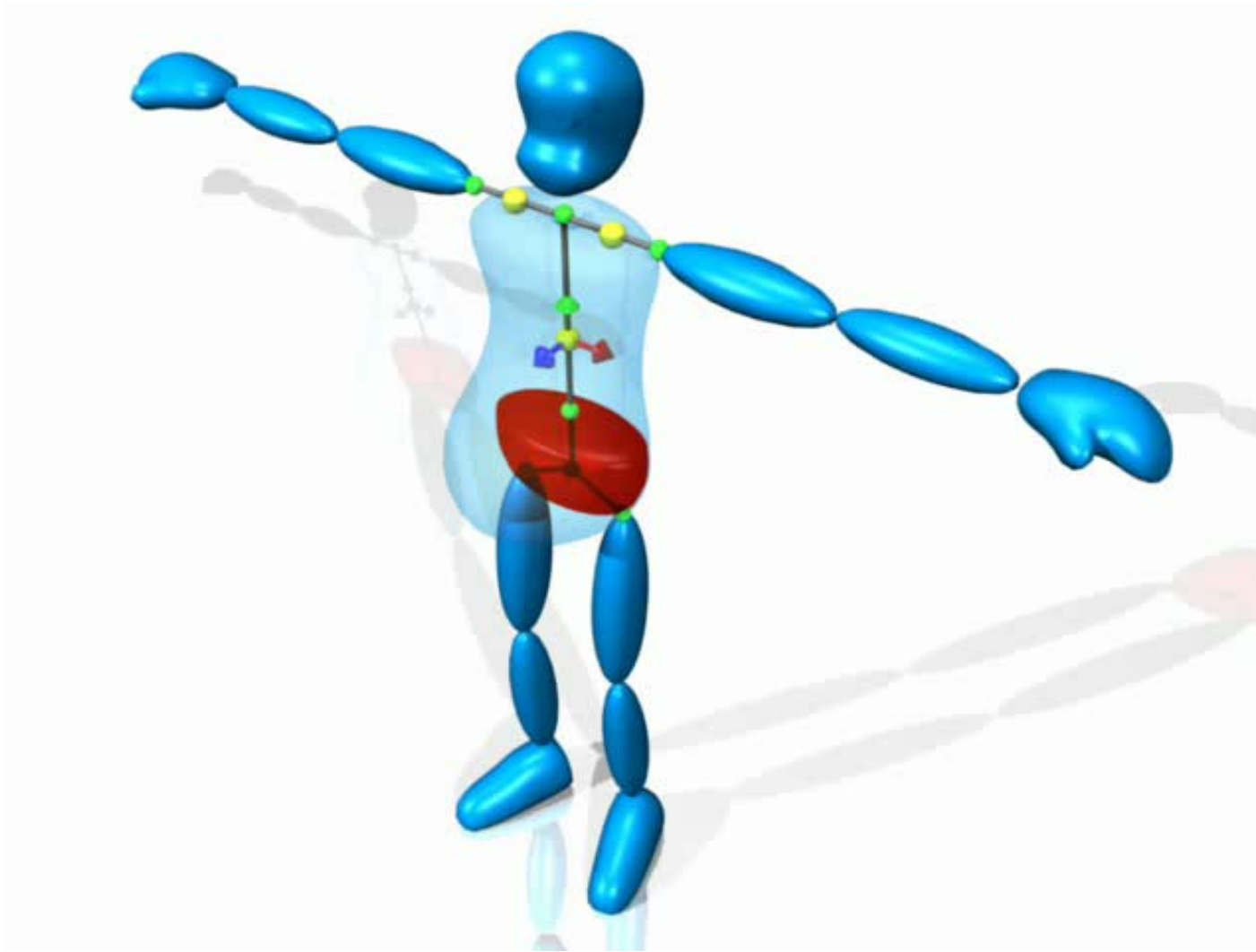


Additional DG Nodes for Constraints

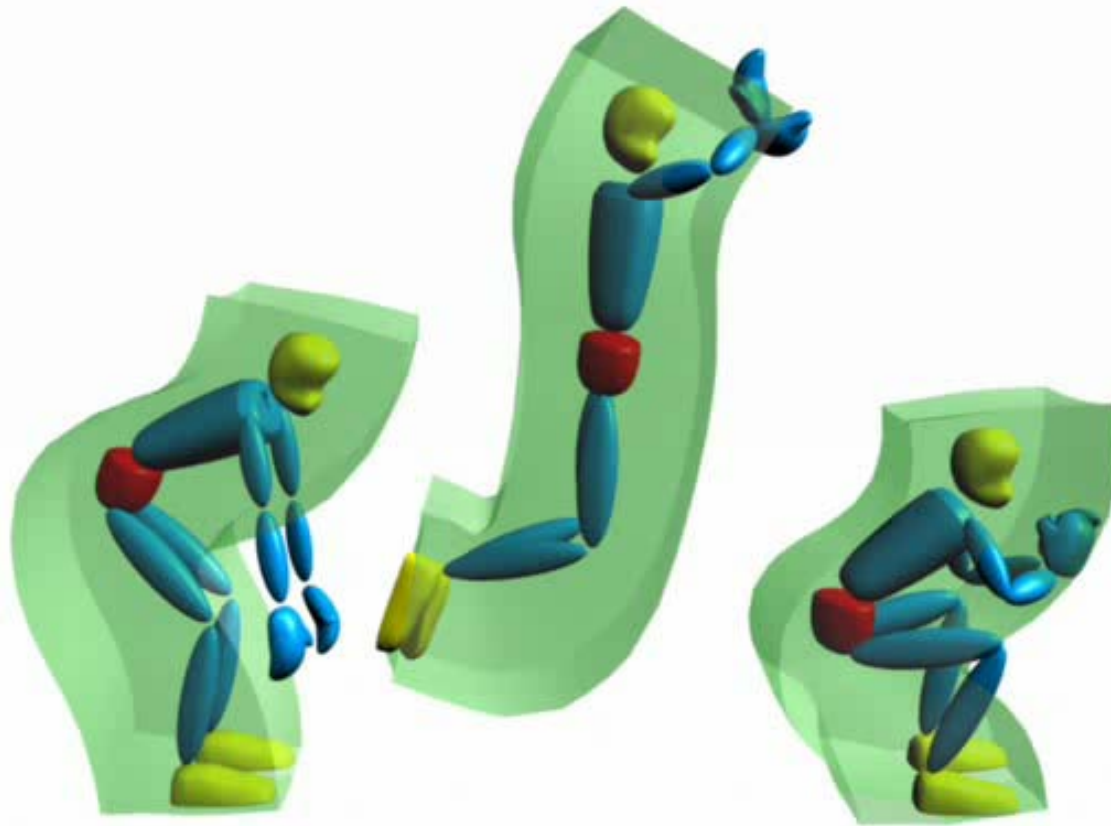
- Derived node of **MPxNode**
 - Simple attributes
 - Simple connections



Additional DG Nodes for Constraints



Additional DG Nodes for Constraints



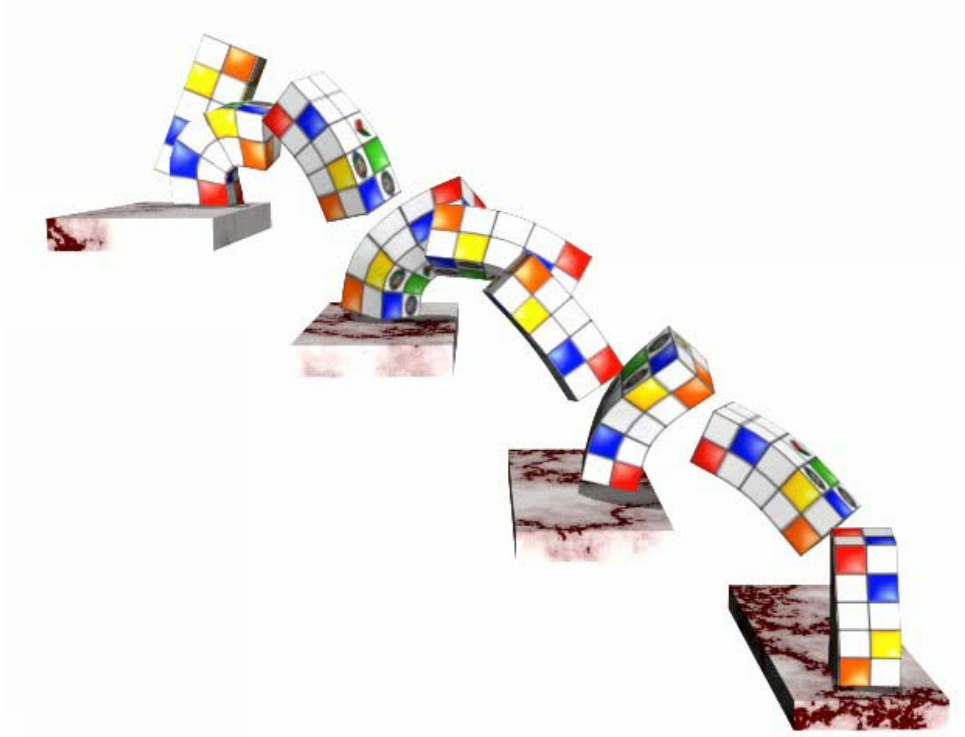
Thin-Shelled Deformable Characters

- ❑ **Degenerate cases: more difficult problems**
 - Thin shell: 2 dimensional solids, i.e. surfaces
- ❑ **The plug-in is similar to the solid deformer.**

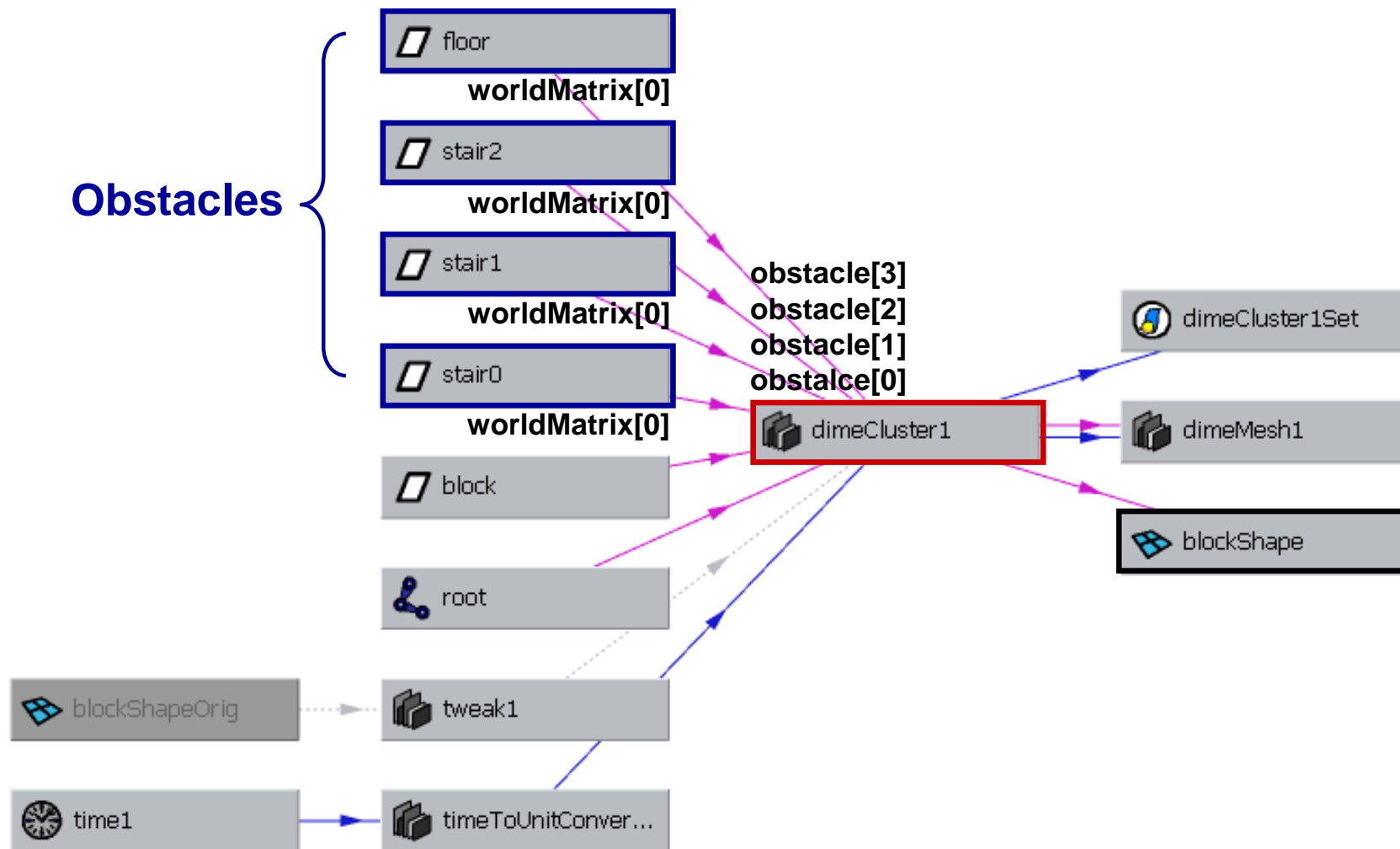


Free-Floating Deformable Characters

- **Modal warping + rigid body simulation**
 - Impulse-based collisions and multiple contacts [Baraff '92, '94]

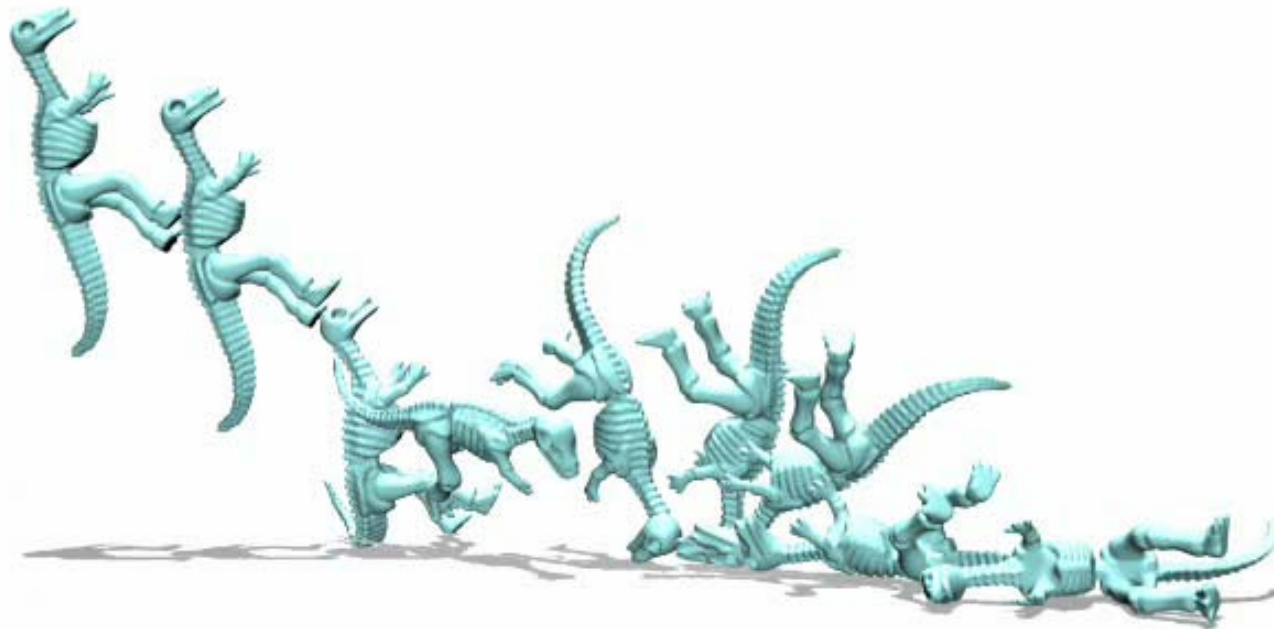


Free-Floating Deformable Characters



Free-Floating Deformable Characters

- **Modal warping + rigid body simulation**
 - Impulse-based collisions and multiple contacts [Baraff '92, '94]



Motion Planning

[ACM TOG 2003]

- **MPxSurfaceShape** for hardware rendering
 - It can be used for visualization though image composition.

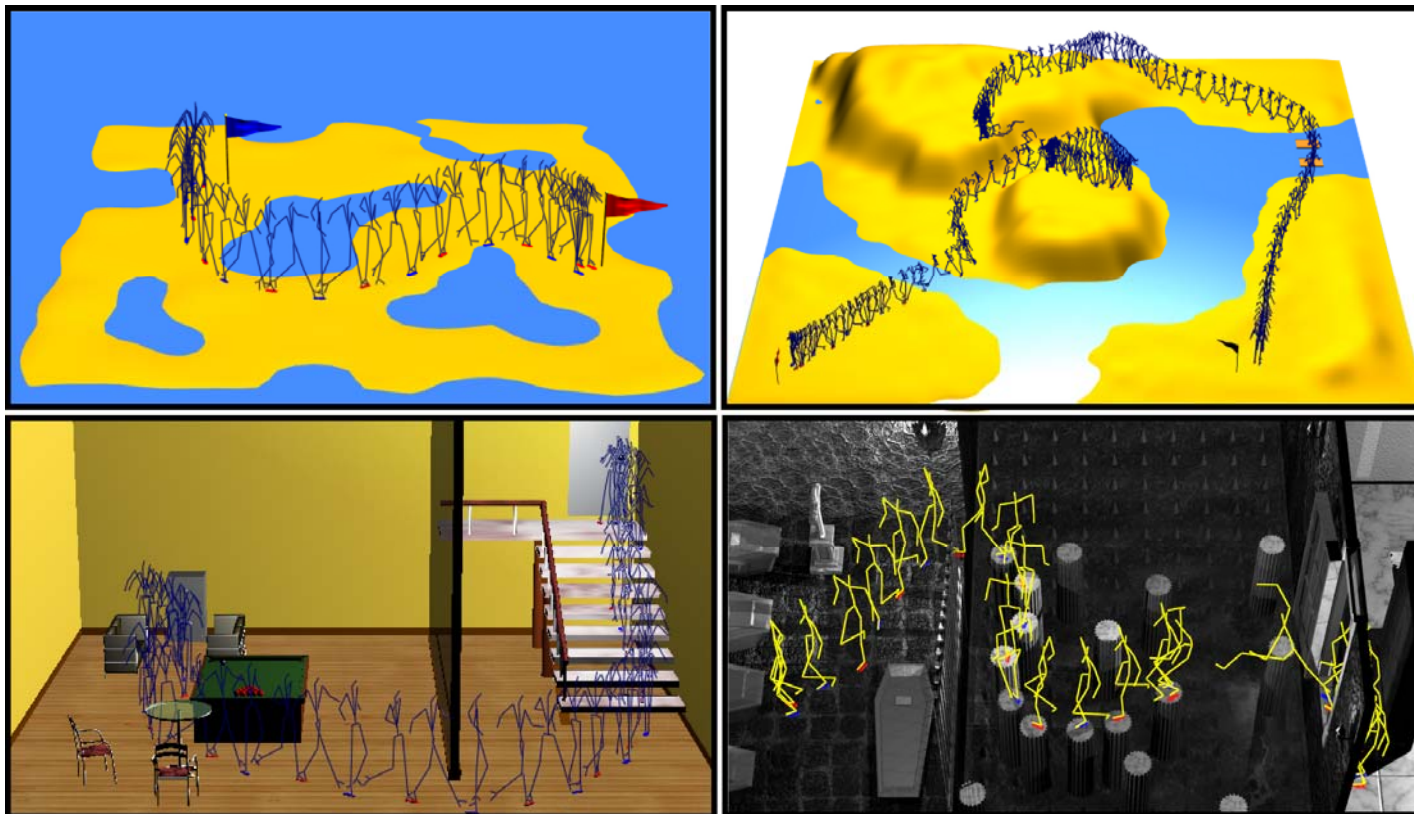


Image Composition

- ❑ Obtain images with depth values
- ❑ Composite images based on depth values

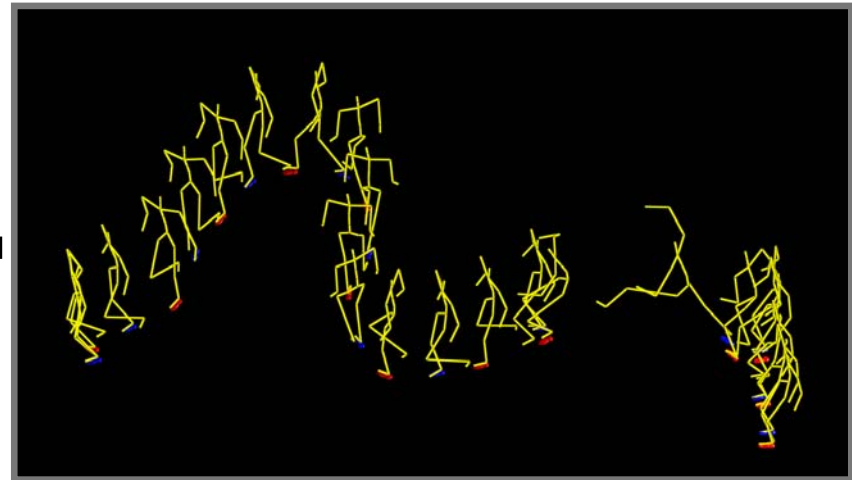
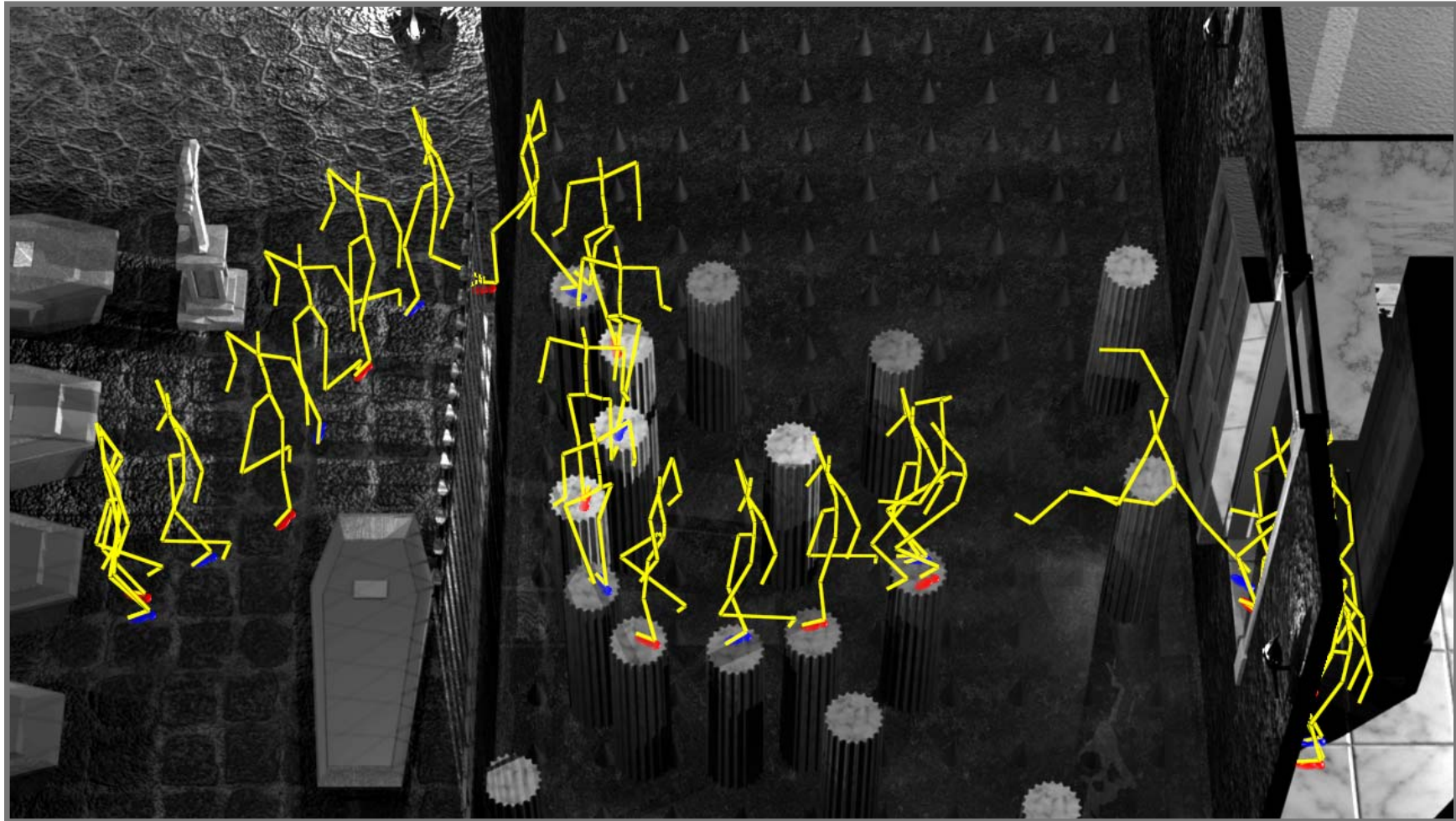
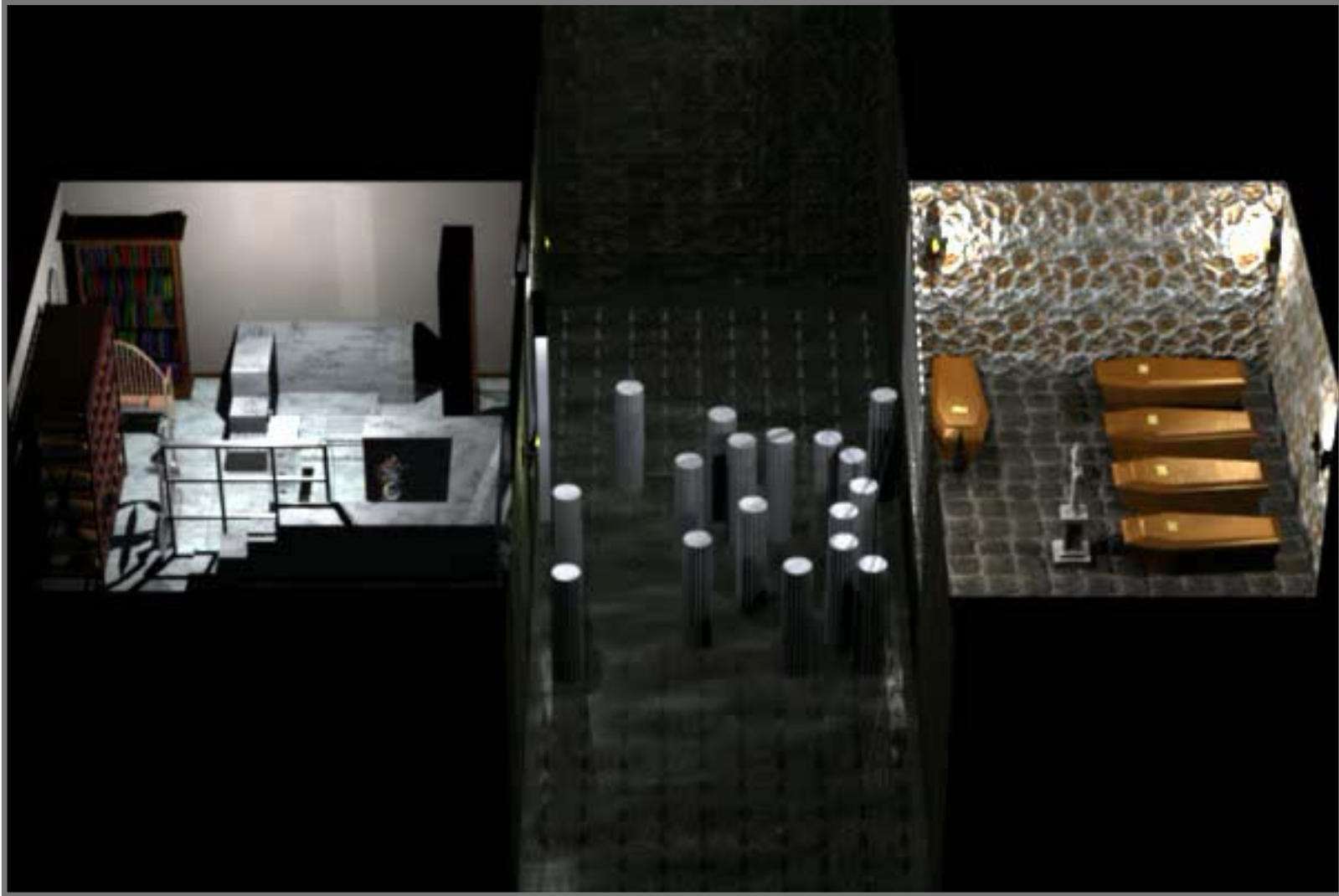


Image Composition

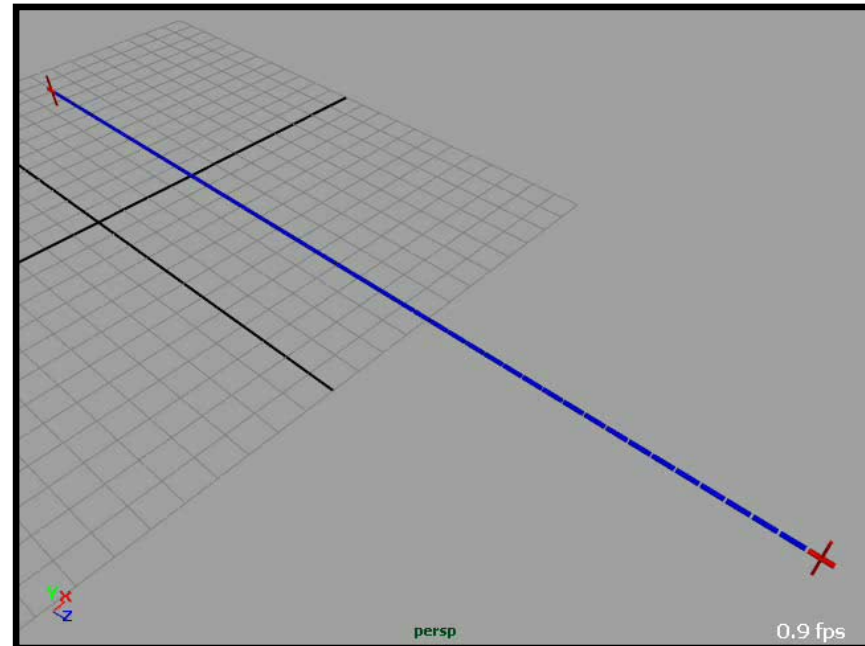
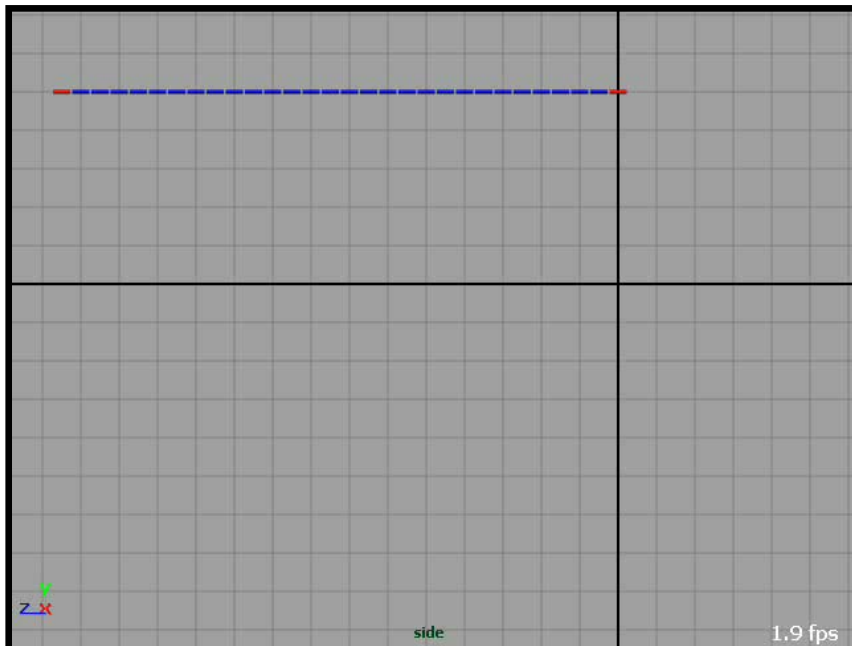


Motion Planning



Dynamic Strand

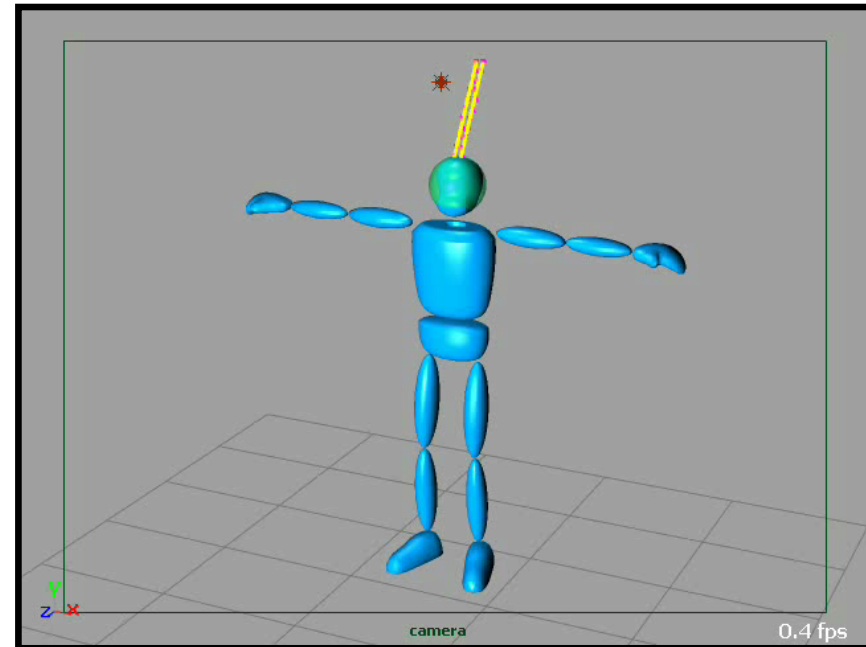
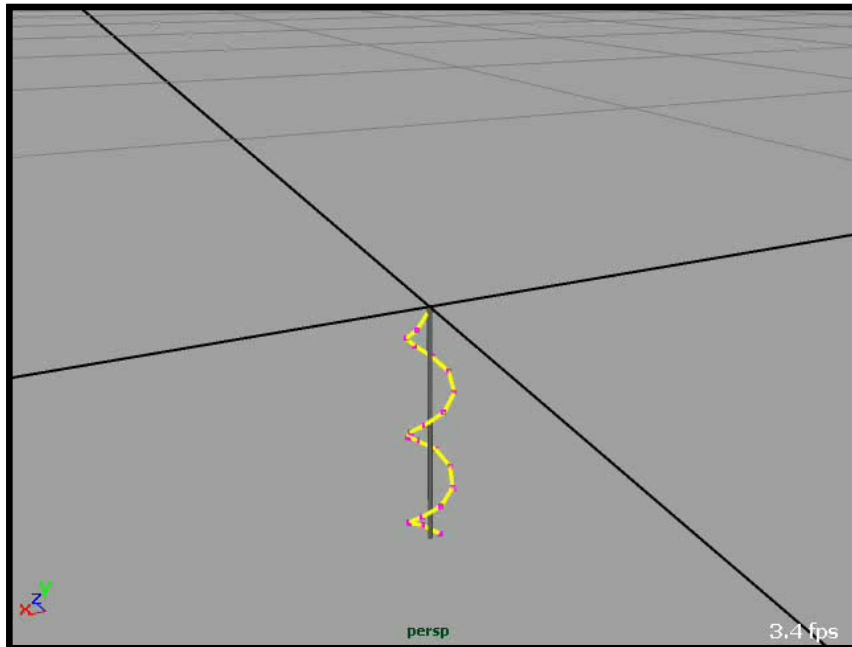
- **Maya can also be used for fast prototyping.**
 - By developing a **command** that adjusts only transform nodes



Movie obtained by playblast

Dynamic Strand

- **Maya can also be used for fast prototyping.**
 - If successful, develop relatively sophisticated **nodes**.



Movie obtained by playblast

Hair Simulation

[SCA 2005]

- **Dynamic strand for hair simulation**
 - The image was obtained using Renderman.



Produced by Byungwon Choe

Summary

<http://cg.kw.ac.kr> mgchoi@kw.ac.kr

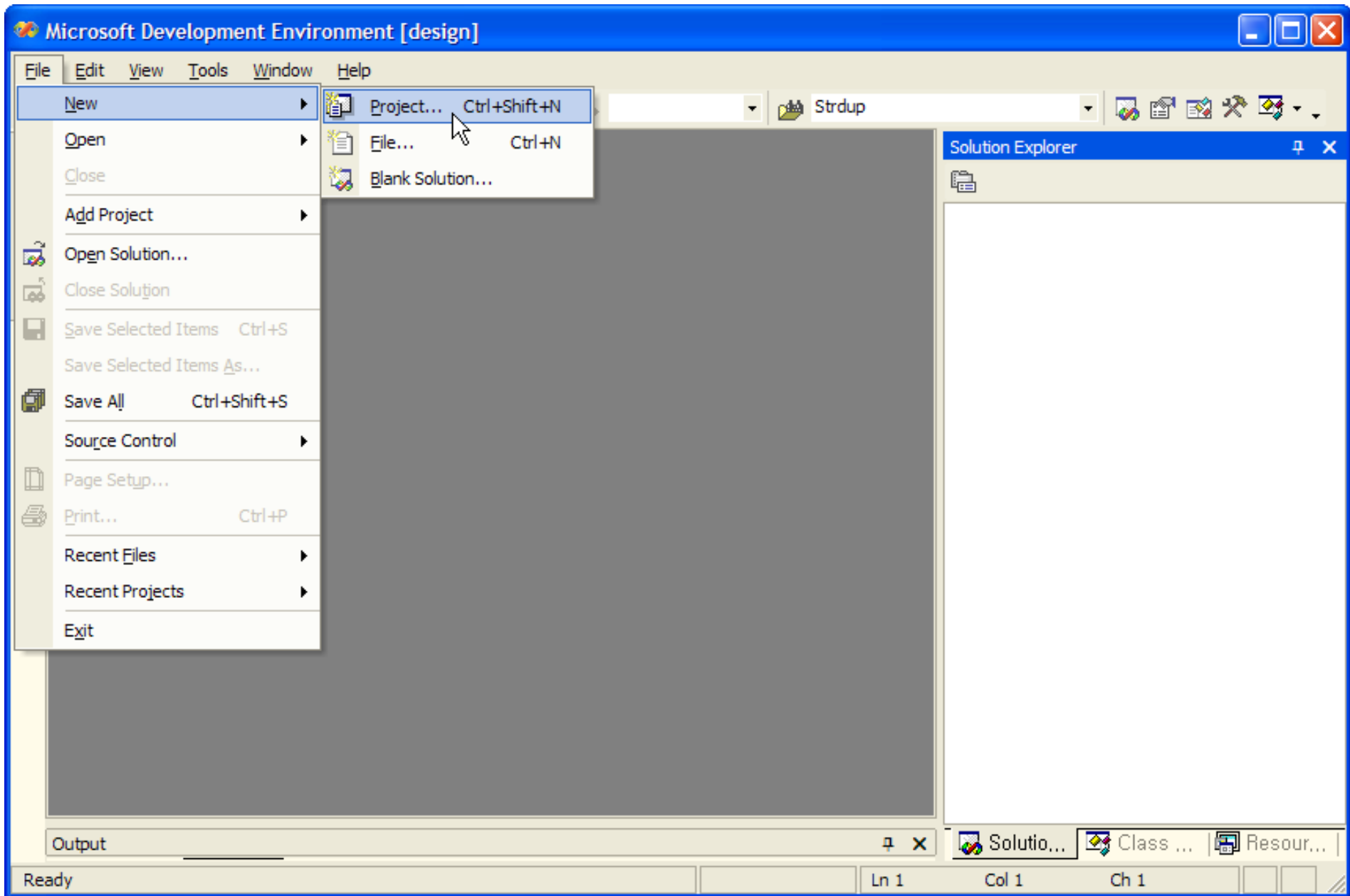
- **Playing with Maya through MEL/API**
 - Nodes/Commands
 - ...

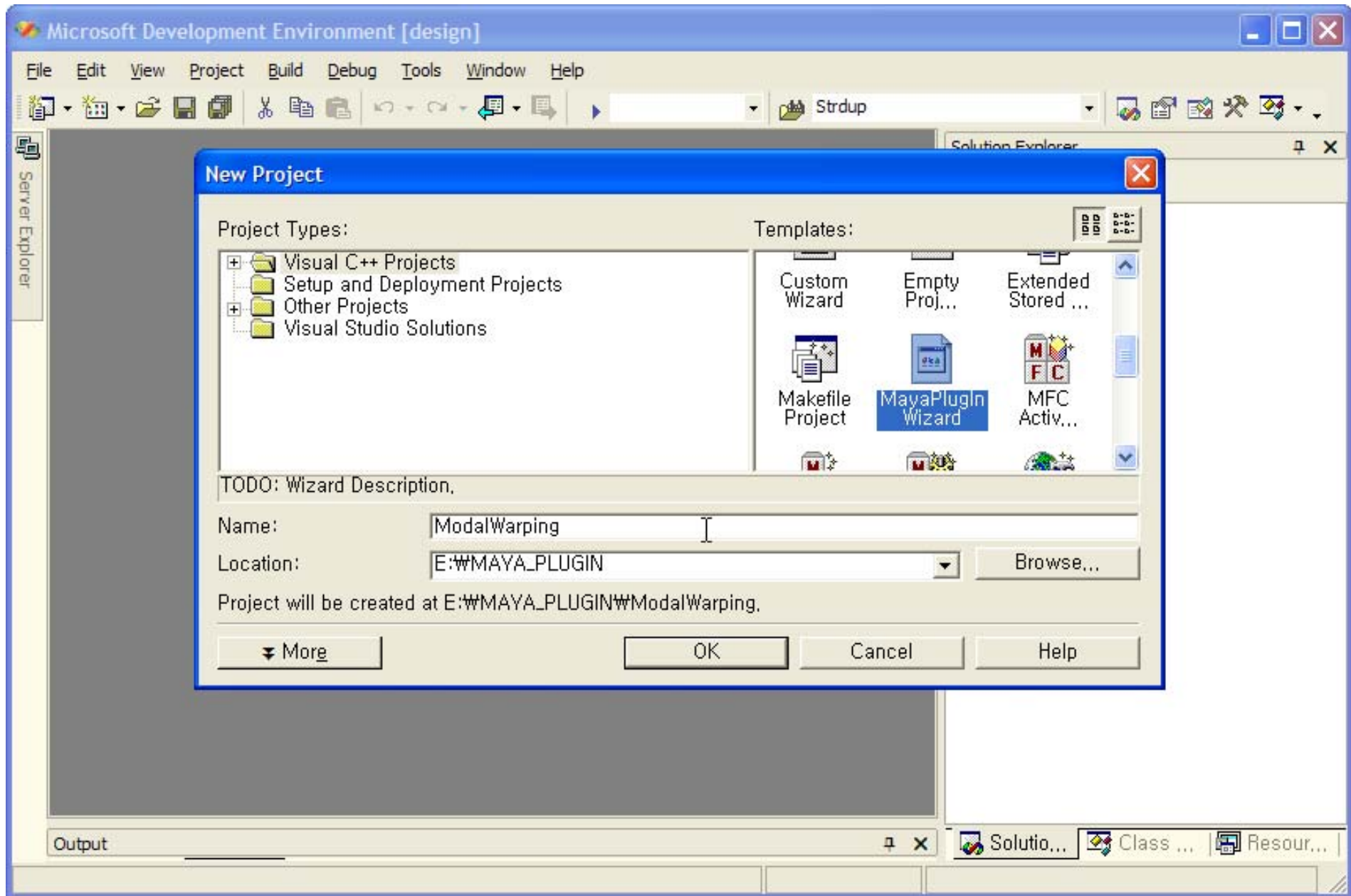
- **Refer to “Maya Developer’s Manual”**
 - C:\Program Files\Alias\Maya6.0\PDF\Reference\API.pdf
 - Developer resources in Maya Help (F1)



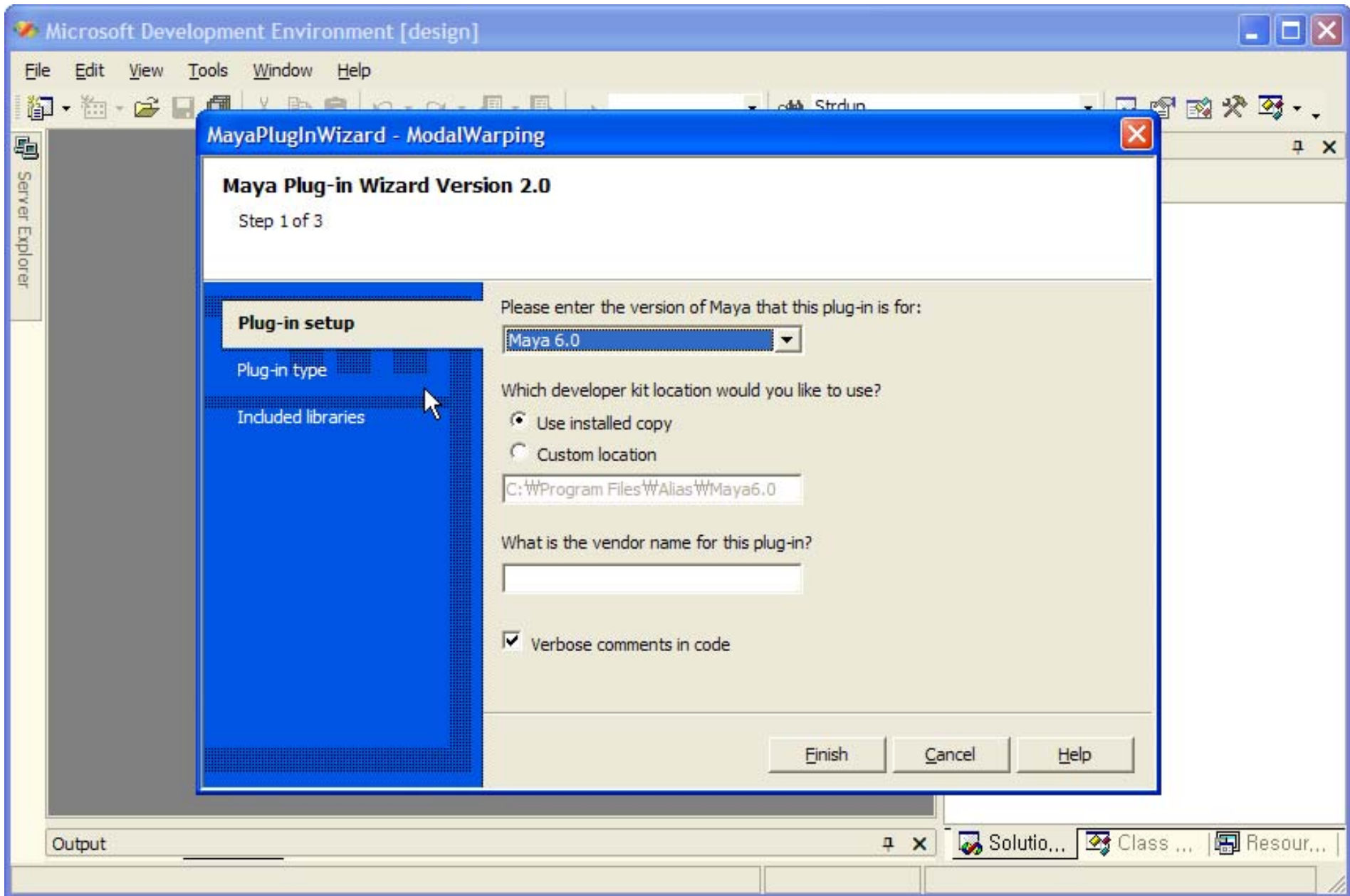
For more video clips, type **Min Gyu Choi** in *Google*

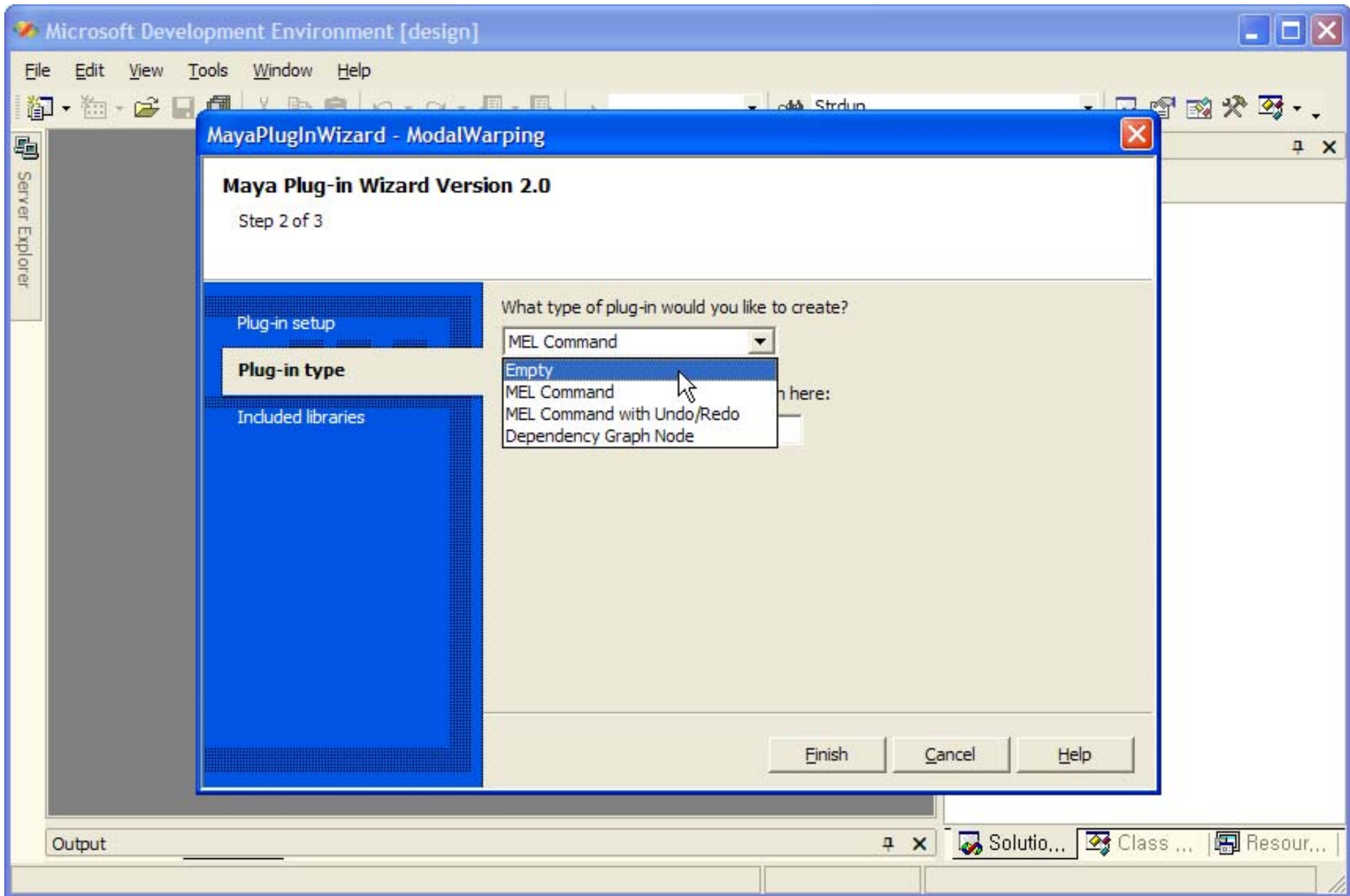
Appendix I

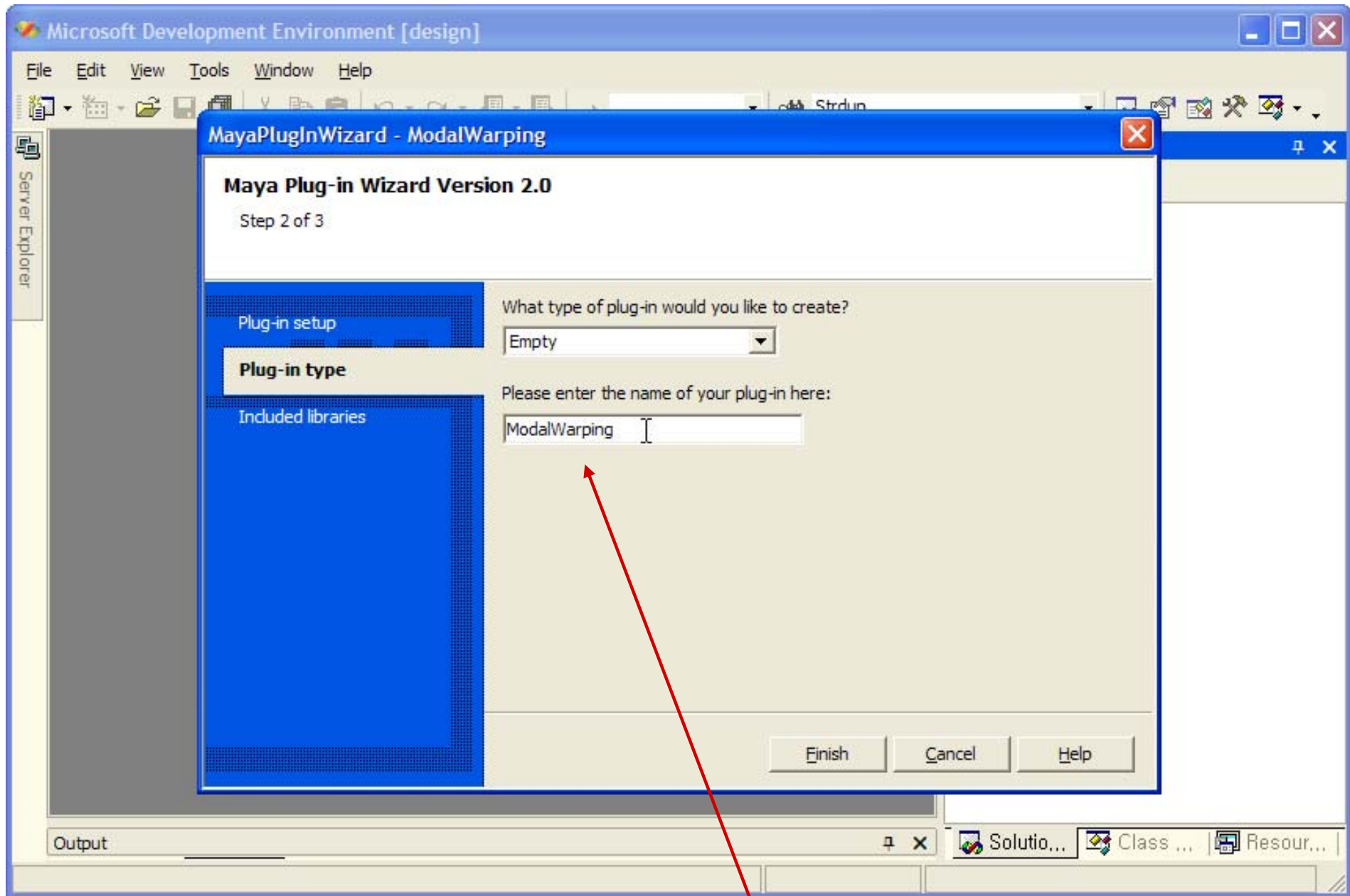




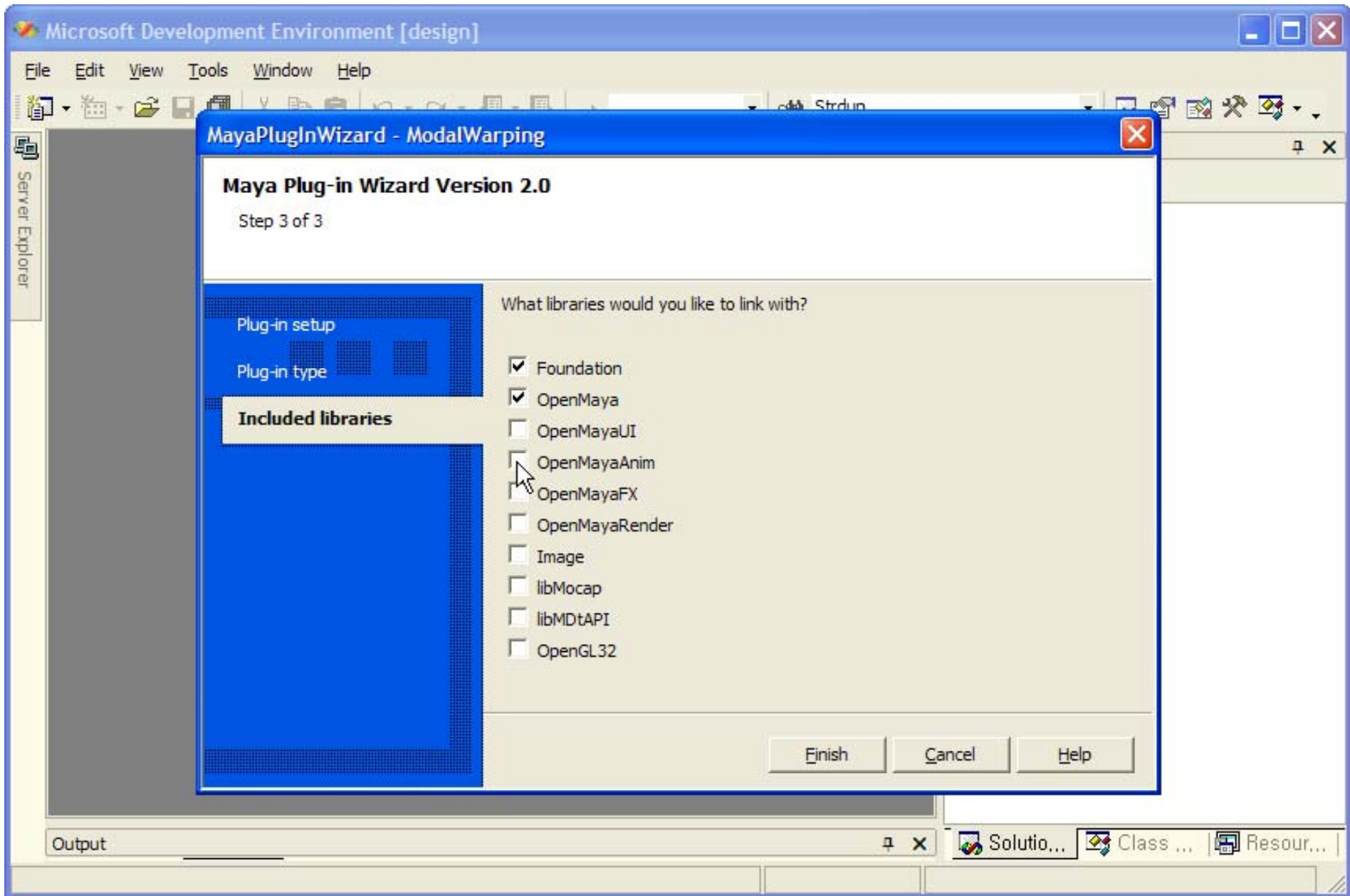
What if there was not such a wizard?

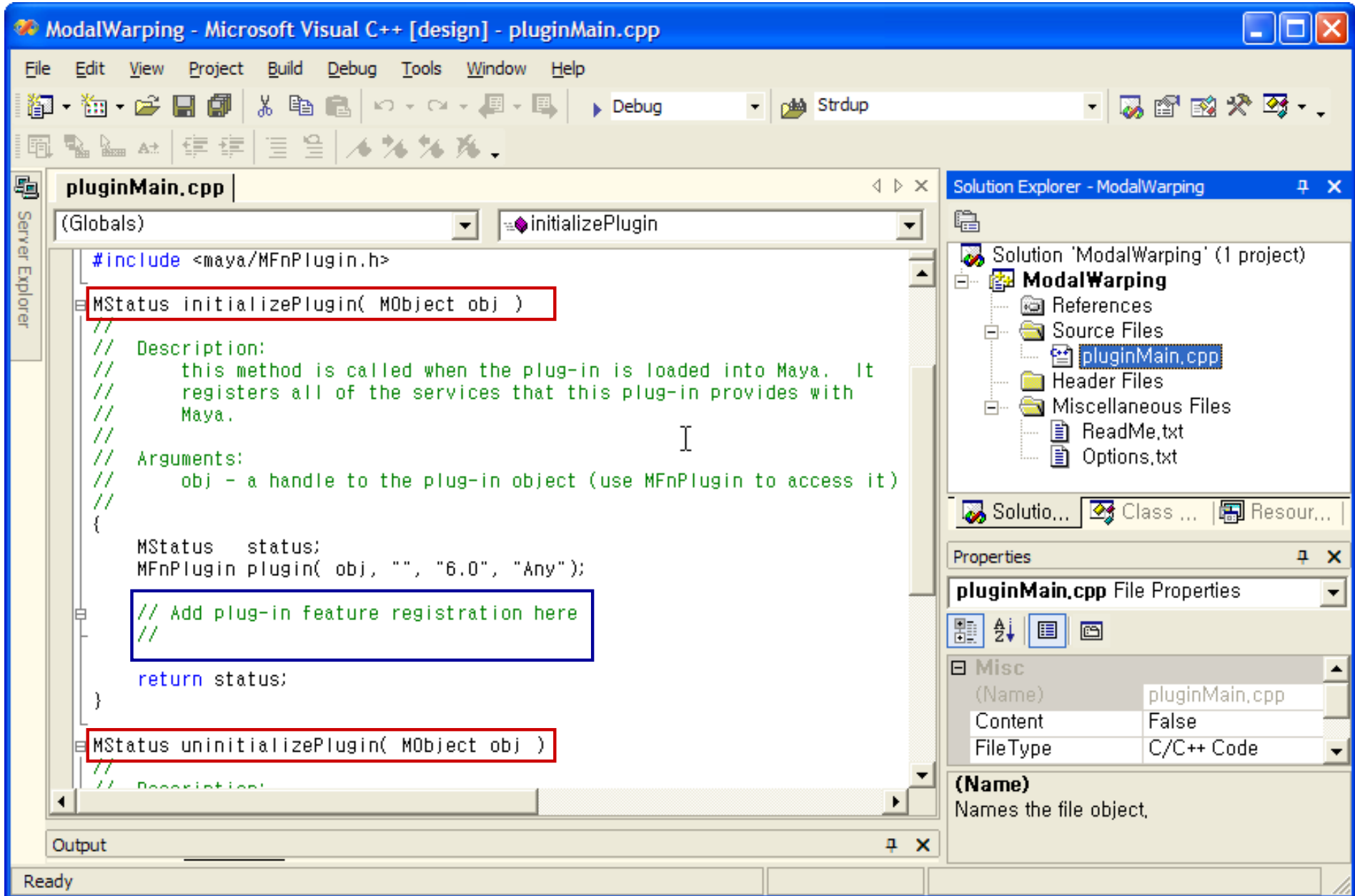






You can achieve **ModalWarping.mll** after compiling/linking.





Maya Plugin Wizard

□ How to install the wizard for Visual Studio .NET?

- Follow the instructions in

- ~\Maya6.0\devkit\pluginwizard\MayaWizardReadme.txt

□ **NOTE**

At this time, the Maya install process will **not** copy the wizard files into the appropriate .NET directories. This process is **manual**.

- Unzip a zip file into your .NET directory

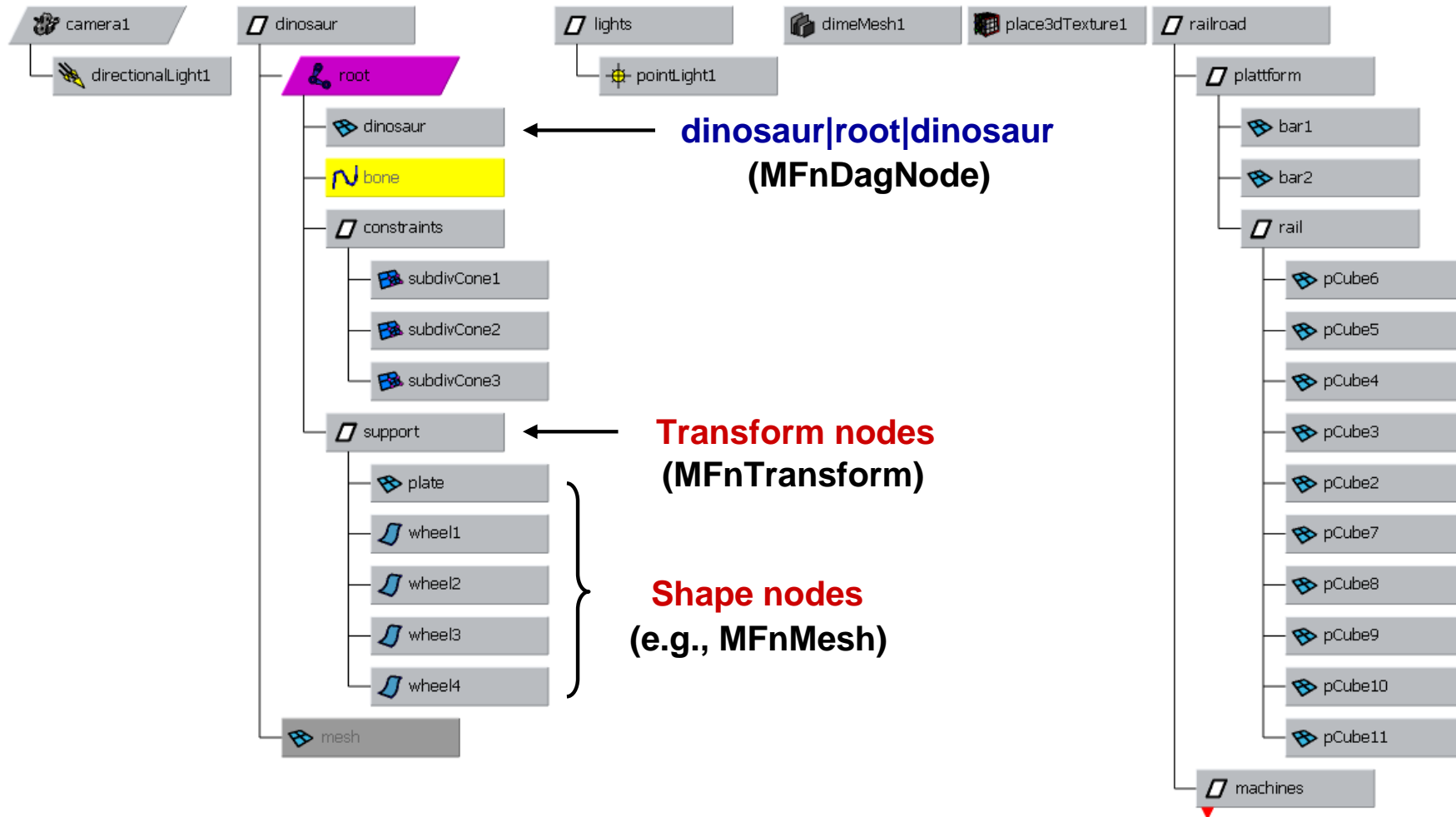
- ~\Maya6.0\devkit\pluginwizard\MayaPluginWizard2.0.zip

Maya Plugin Wizard

- How to install the wizard for Visual Studio .NET?**
 - Unzip the [MayaPluginWizard2.0.zip](#) file
 - Copy the following files to the "[C:\Program Files\Microsoft Visual Studio .NET 2003\Vc7\vcprojects](#)" directory:
 - [MayaPlugInWizard.vmdir](#)
 - [MayaPlugInWizard.vsz](#)
 - Copy the "[MayaPluginWizard](#)" directory to "[C:\Program Files\Microsoft Visual Studio .NET 2003\Vc7\VCWizards](#)"

Appendix III

Directed Acyclic Graph in Maya



A kind of scene graph

Dependency Graph Plug-Ins

□ Twelve parent classes for dependency graph plug-ins

- MPxNode
- MPxLocatorNode
- MPxIkSolverNode
- MPxDeformerNode
- MPxFieldNode
- MPxEmitterNode
- MPxSpringNode
- MPxManipContainer
- MPxSurfaceShape
- MPxObjectSet
- MPxTransform
- MPxHwShaderNode

You can **subclass** new nodes from these!

*n*VIDIA Cg Plug-In
