

CS 380

Introduction to Computer Graphics

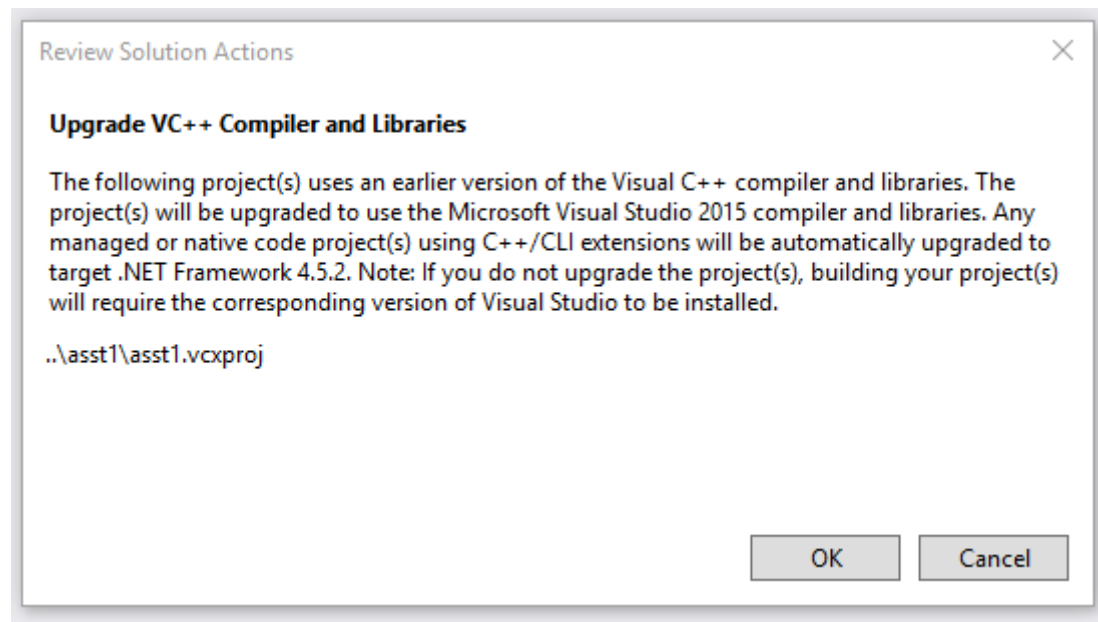
LAB (1) : OpenGL Tutorial (2)

2018. 03. 12

Reference : Foundations of 3D Computer Graphics, Steven J. Gortler

- Set up a project for OpenGL programs.
- Understand texture mapping.
- Practice with shader code.

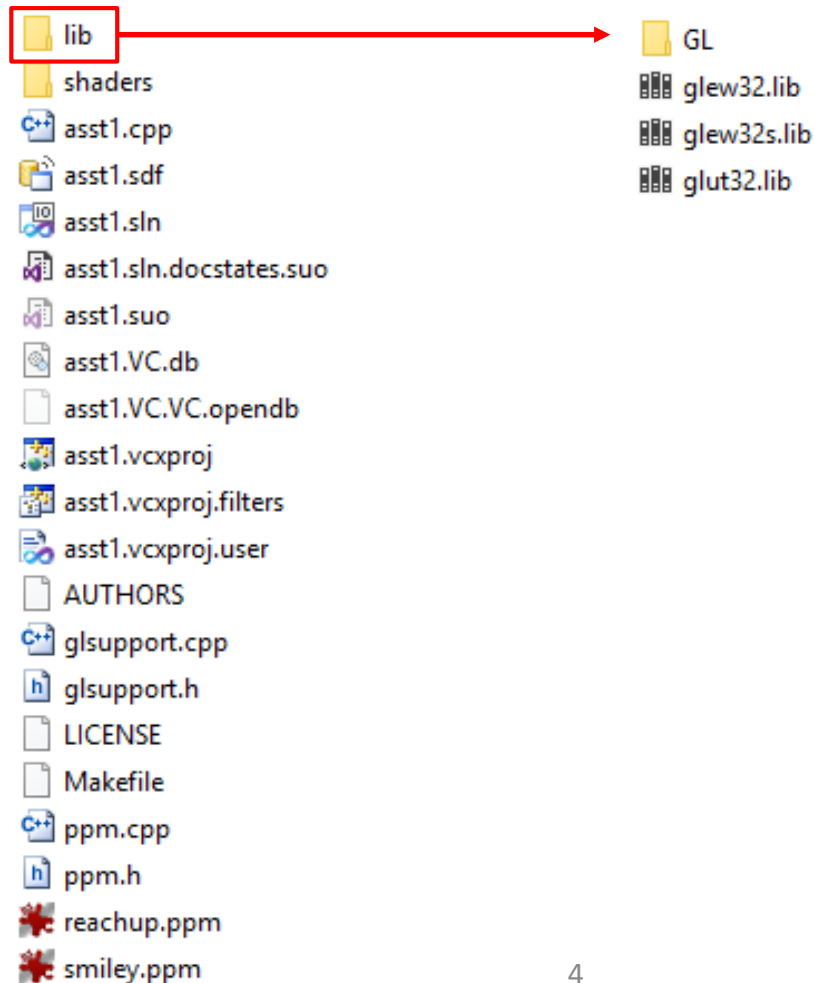
Upgrade Visual Studio Compiler



Click OK

Library Setup

- Your project should include lib folder contains glew, glut



Include Header Files

asst1 Property Pages

Configuration: Debug Platform: Active(Win32) Configuration Manager...

- Configuration Properties
 - General
 - Debugging
 - VC++ Directories
 - C/C++
 - General**
 - Optimization
 - Preprocessor
 - Code Generation
 - Language
 - Precompiled Headers
 - Output Files
 - Browse Information
 - Advanced
 - All Options
 - Command Line
 - Linker
 - General
 - Input
 - Manifest File
 - Debugging
 - System
 - Optimization
 - Embedded IDL
 - Windows Metadata
 - Advanced

Additional Include Directories	lib
Additional #using Directories	
Debug Information Format	Program Database for Edit And Continue (/ZI)
Common Language Runtime Support	
Consume Windows Runtime Extension	
Suppress Startup Banner	Yes (/nologo)
Warning Level	Level3 (/W3)
Treat Warnings As Errors	No (/WX-)
Warning Version	
SDL checks	
Multi-processor Compilation	

Additional Include Directories
Specifies one or more directories to add to the include path; separate with semi-colons if more than one. (/I[path])

OK Cancel Apply

Include Library Directory

The screenshot shows the 'asst1 Property Pages' dialog box. The 'Configuration' is set to 'Debug' and the 'Platform' is 'Active(Win32)'. The left-hand tree view shows the 'Linker > General' tab selected. The main area displays a list of linker properties:

Output File	\$\$(OutDir)\$(TargetName)\$(TargetExt)
Show Progress	Not Set
Version	
Enable Incremental Linking	Yes (/INCREMENTAL)
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	No
Register Output	No
Per-user Redirection	No
Additional Library Directories	lib
Link Library Dependencies	Yes
Use Library Dependency Inputs	No
Link Status	
Prevent Dll Binding	
Treat Linker Warning As Errors	
Force File Output	
Create Hot Patchable Image	
Specify Section Attributes	

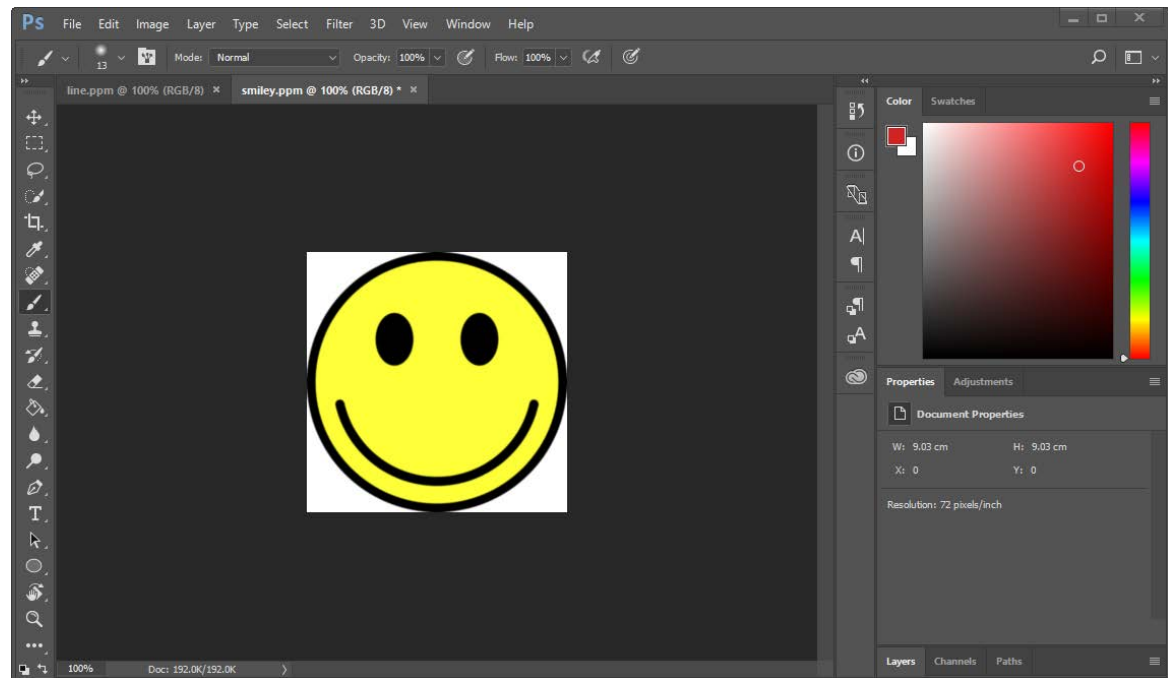
At the bottom, the 'Output File' section is expanded, showing the text: 'The /OUT option overrides the default name and location of the program that the linker creates.'

Add Library Dependencies

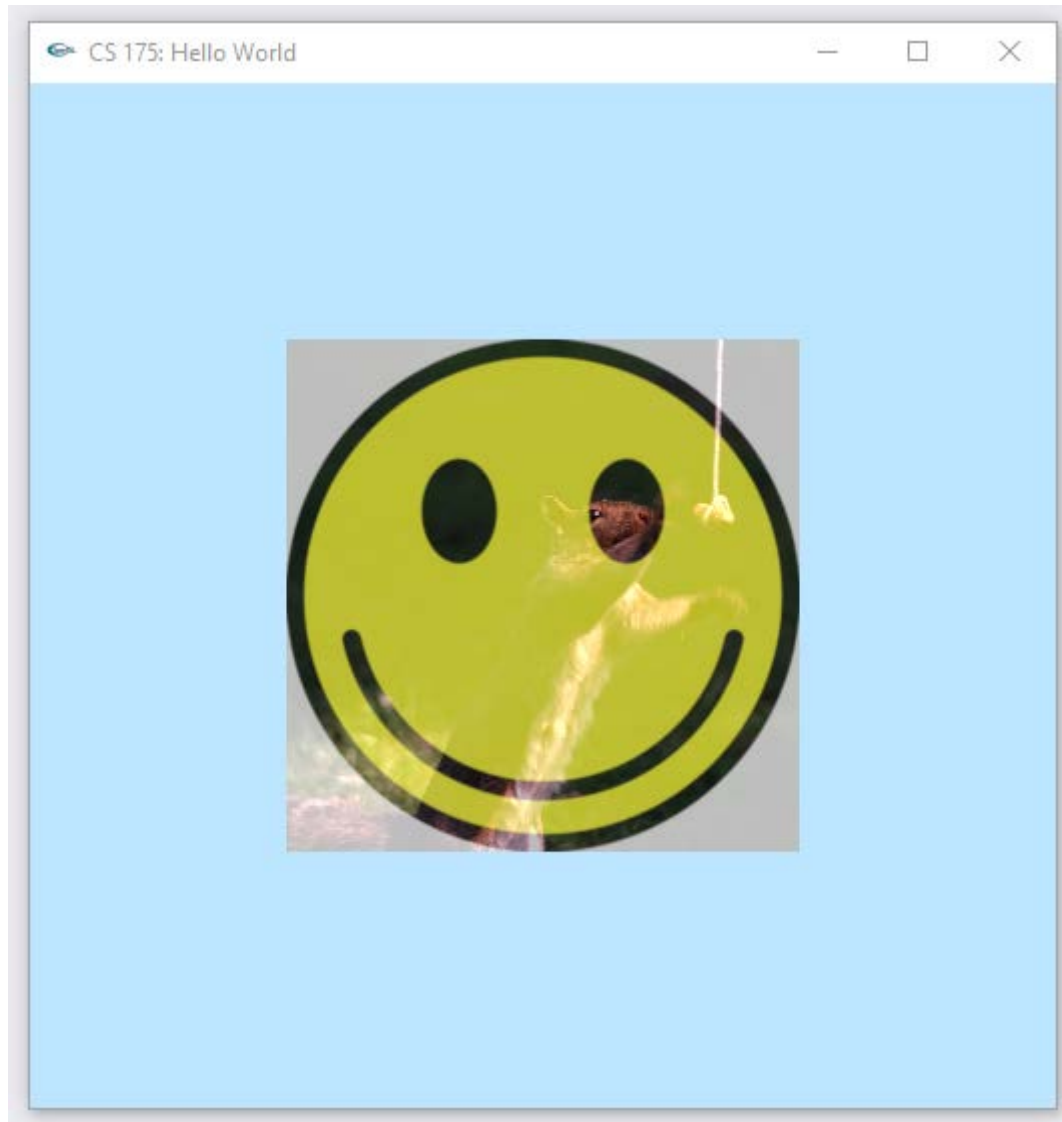
The screenshot shows the 'asst1 Property Pages' dialog box. The 'Configuration' is set to 'Debug' and the 'Platform' is 'Active(Win32)'. The 'Linker' section is expanded, and the 'Input' tab is selected. In the 'Additional Dependencies' field, the text 'glew32.lib;glut32.lib;%(AdditionalDependencies)' is entered and highlighted with a red box. The 'Additional Dependencies' section at the bottom of the dialog explains that this field specifies additional items to add to the link command line, such as 'kernel32.lib'. The 'OK', 'Cancel', and 'Apply' buttons are visible at the bottom right.

About PPM File

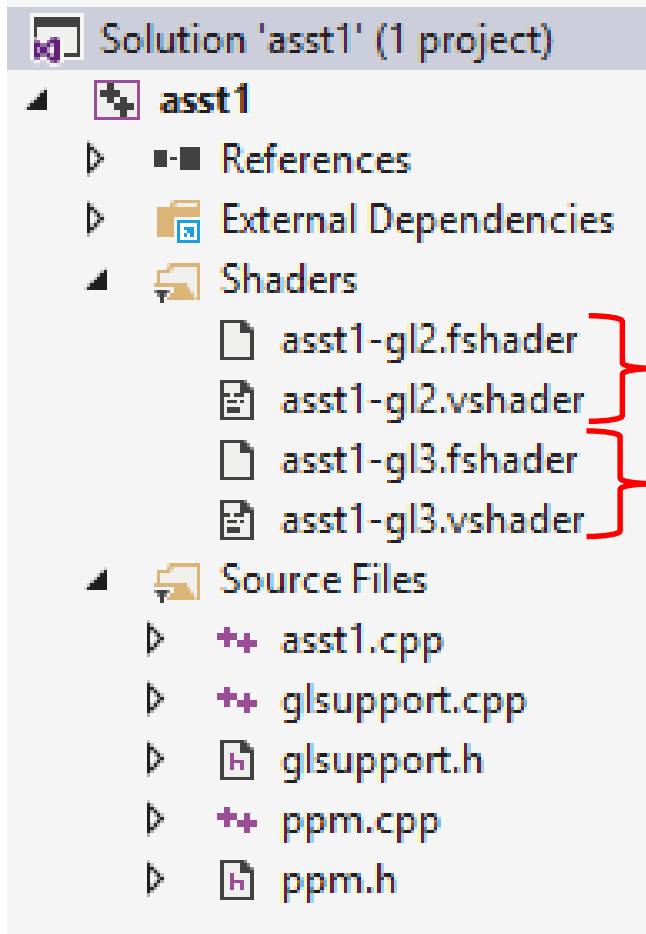
- Image file format such as jpg, png.
- You can open **ppm** file in Photoshop or Gimp.
- The skeleton code includes simple ppm reader (`ppm.cpp`)



First Screen



gl2 & gl3 Shaders



`static const bool g_Gl2Compatible = true;`

`static const bool g_Gl2Compatible = false;`

- Read image file

```
ppmRead(ppmFilename, texWidth, texHeight, pixData);
```

- Load an image to GPU

```
glGenTextures(1, &handle_);
```

Texture ID (int)

```
glActiveTexture(GL_TEXTURE0);
```

```
glBindTexture(GL_TEXTURE_2D, texHandle);
```

```
glTexImage2D(GL_TEXTURE_2D, 0, g_GL2Compatible ? GL_RGB : GL_SRGB, texWidth, texHeight,  
            0, GL_RGB, GL_UNSIGNED_BYTE, &pixData[0]);
```

- Setup texture parameters

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
```

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
```

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
```

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
```

- Find location of uniform variables in the c++ code.

```
h_uVertexScale = safe_glGetUniformLocation(h, "uVertexScale");  
h_uTexUnit0 = safe_glGetUniformLocation(h, "uTexUnit0");  
h_uTexUnit1 = safe_glGetUniformLocation(h, "uTexUnit1");
```

- Set the values of uniform variables.

```
safe_glUniform1i(curSS.h_uTexUnit0, 0);  
safe_glUniform1i(curSS.h_uTexUnit1, 1);  
safe_glUniform1f(curSS.h_uVertexScale, g_objScale);
```

Pass texture ID for texture uniform value

- Accept the texture with sampler2D in shader.

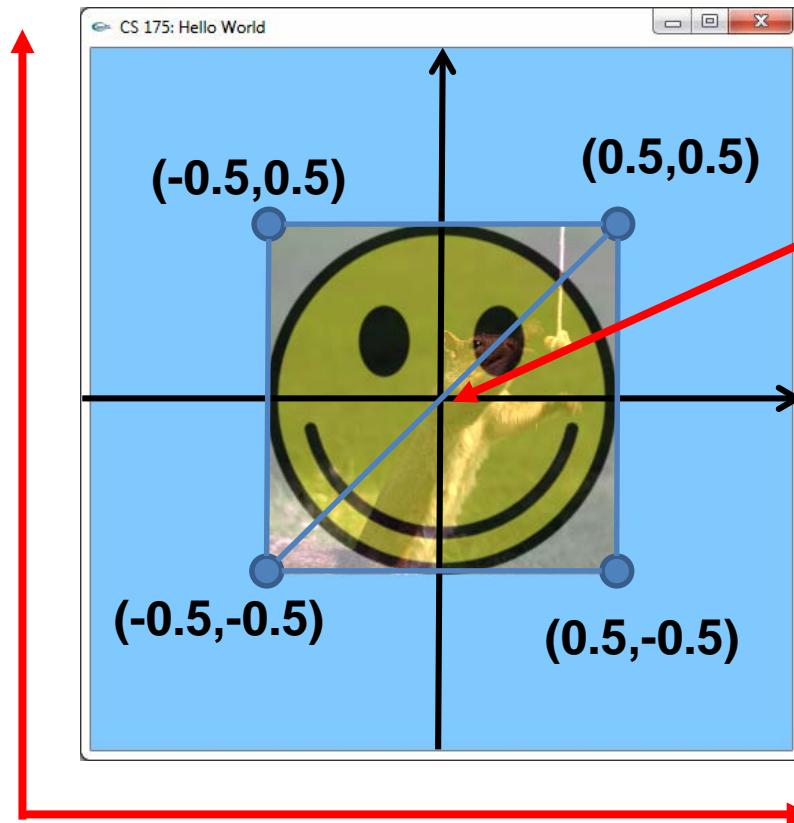
```
uniform sampler2D uTexUnit0, uTexUnit1;
```

Notice that we pass integer value to uTexUnit0,
but shader accept the value as sampler2D

- glUniform1i: set integer value
- glUniform1f: set float value
- glUniform3i : set 3d integer vector
- glUniform3f : set 3d float vector
- glUniformMatrix3fv: set 3×3 float matrix
- All functions are listed in
 - <https://www.khronos.org/registry/OpenGL-Refpages/gl4/html/glUniform.xhtml>

Homework 1

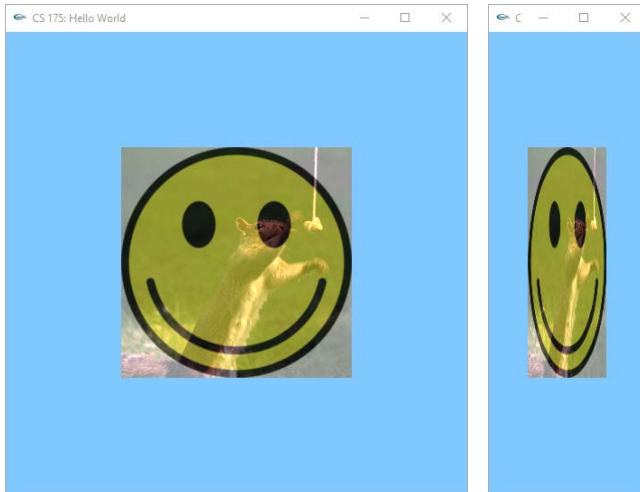
- Geometric overview



Center point of screen: $(0, 0)$

Total size of screen: 2×2

Detect changing window size



Call reshape function

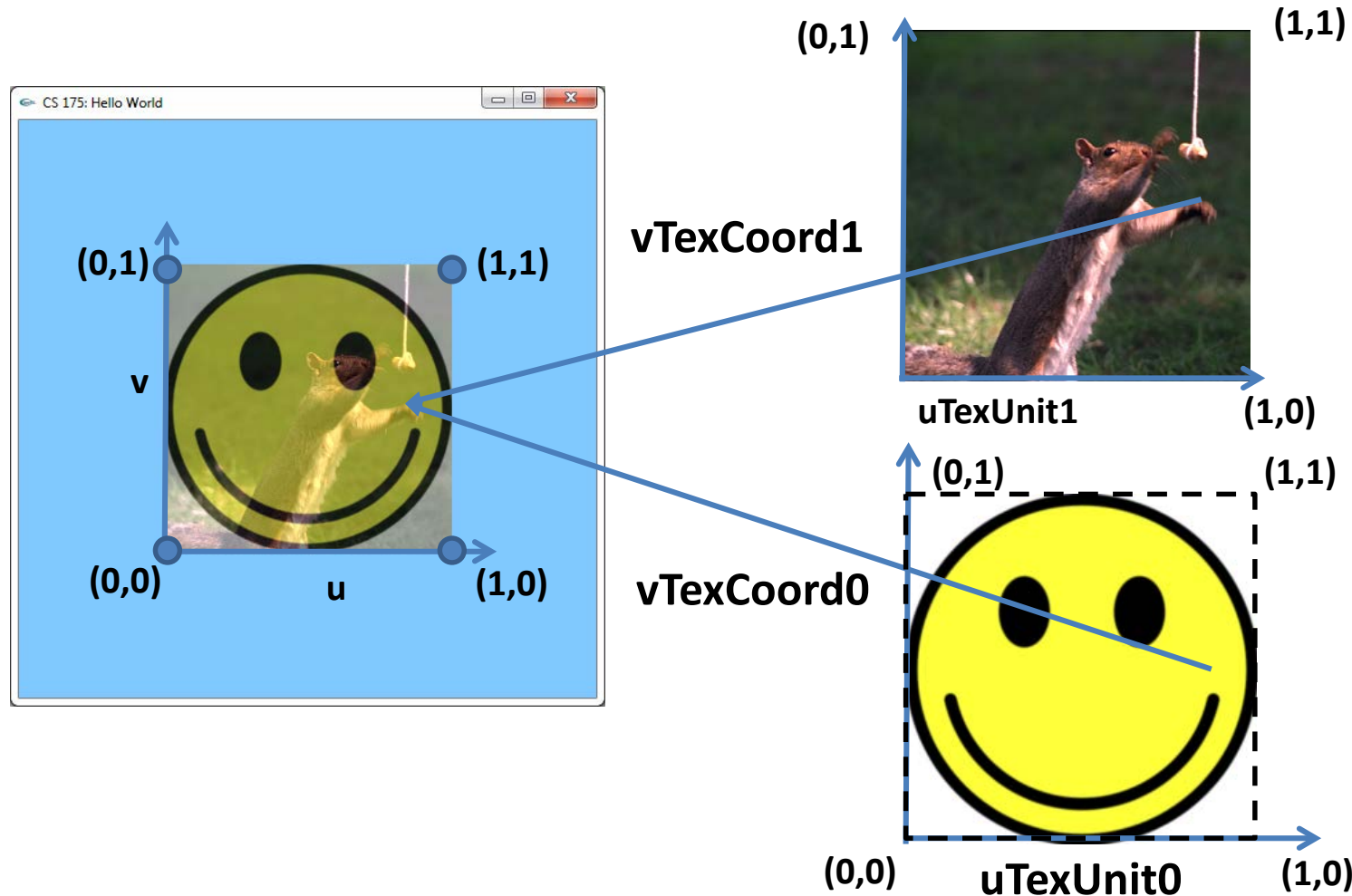
```
/// Whenever a window is resized, a "resize" event is  
/// generated and glut is told to call this reshape  
/// callback function to handle it appropriately.
```

```
static void reshape(int w, int h) {  
    g_width = w;  
    g_height = h;  
    glViewport(0, 0, w, h);  
    glutPostRedisplay();  
}
```

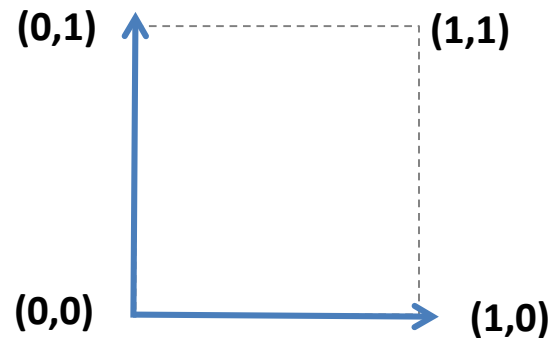
- glViewport sets the width and height of the viewing box.
- glutPostRedisplay() calls the display callback function.

Texture mapping

- `texture(uTexUnitX, vTexCoordX) → Color`



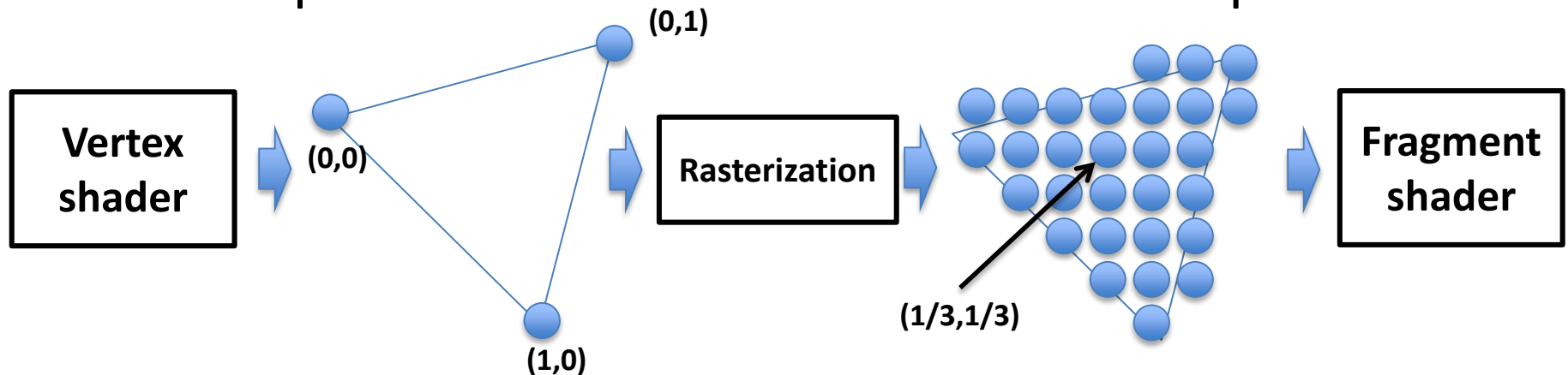
- `vTexCoord (vec2)`
 - A texture coordinate of each vertex or fragment



-
- `uTexUnit (sampler2d)`
 - An image we pass to the shaders.



- In a vertex shader
 - Set a texture coordinate for each vertex
- In a fragment shader
 - As an input of the fragment shader, we get an interpolated texture coordinate for each pixel.



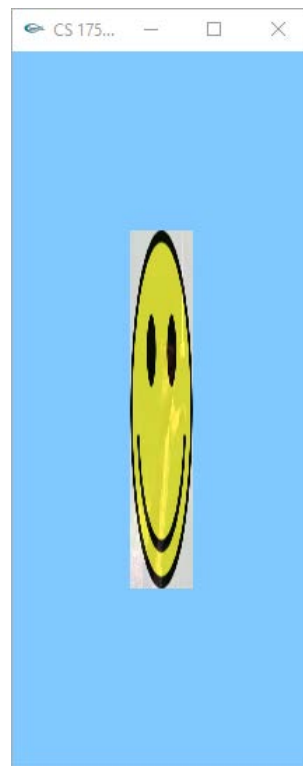
Expected Result

- The smile image should maintain half size of shorter direction (horizontal, vertical).

Correct



Wrong cases



Constraints & Tip for HW1

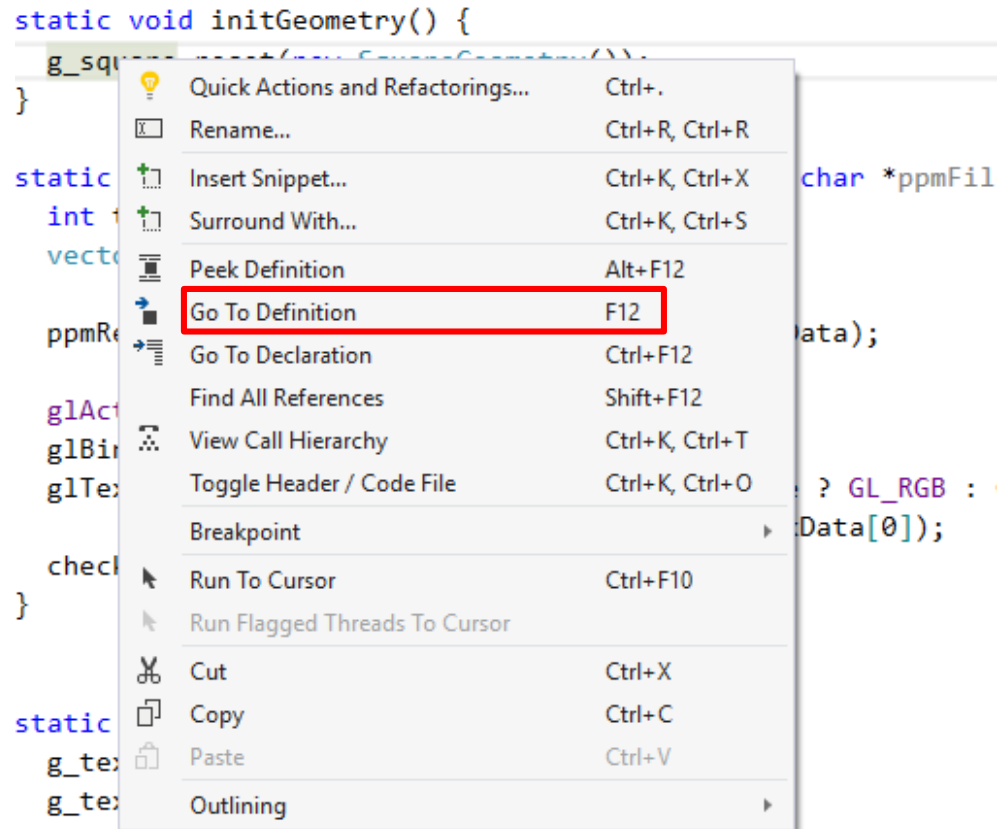
- Your solution cannot change the vertex coordinate
- Your solution cannot modify the call to `glViewport`
 - Do not change the following line

```
glViewport(0, 0, w, h);
```

- Modify shader codes
- Modify uniform variables in your C code

Visual Studio Tips

- If you click mouse right button, you can see **Go To Definition** which leads you to where the variable or function defined.



- Homework due
 - 3/21 (Wed) 23:59
- Submission
 - Zip your code folder and send it to sjjeon@vclab.kaist.ac.kr
 - Mail title: [CS380] HW1 20161234 Name
 - Zip file name: hw1_20161234_Name.zip
- You can submit today if you finish the task.