CS6533/CS4533 Lecture 12 Slides/Notes

Shadow Blending in HW4 (b); Texture Mapping (Notes, Sample Program)

By Prof. Yi-Jen Chiang
CSE Dept., Tandon School of Engineering
New York University

1

*** Discussing how to modify the ``making decal'' process in HW3 to do shadow blending in HW4 part (b)** (a demo of HW4 is shown first).

* **Texture Mapping**

- Discussing the sample program ``Handout: checker-new.cpp" (complete sample program has been posted at https://cse.engineering.nyu.edu/cs653/Checker.tar.gz.
- Some screenshots of this sample program with annotations are then shown next.
- Showing a demo of this sample program, as well as a demo of HW4.

2

Making Decal (from HW3) ⇒ HW4 (b) for semi-transparent shadow. (changes are shown in green)

(shadow is a decal on top of ground).

0. Always enable z-buffer testing.

1. (Draw ground only to frame buffer)
   Disable writing to z-buffer
   Draw ground (only to frame buffer)

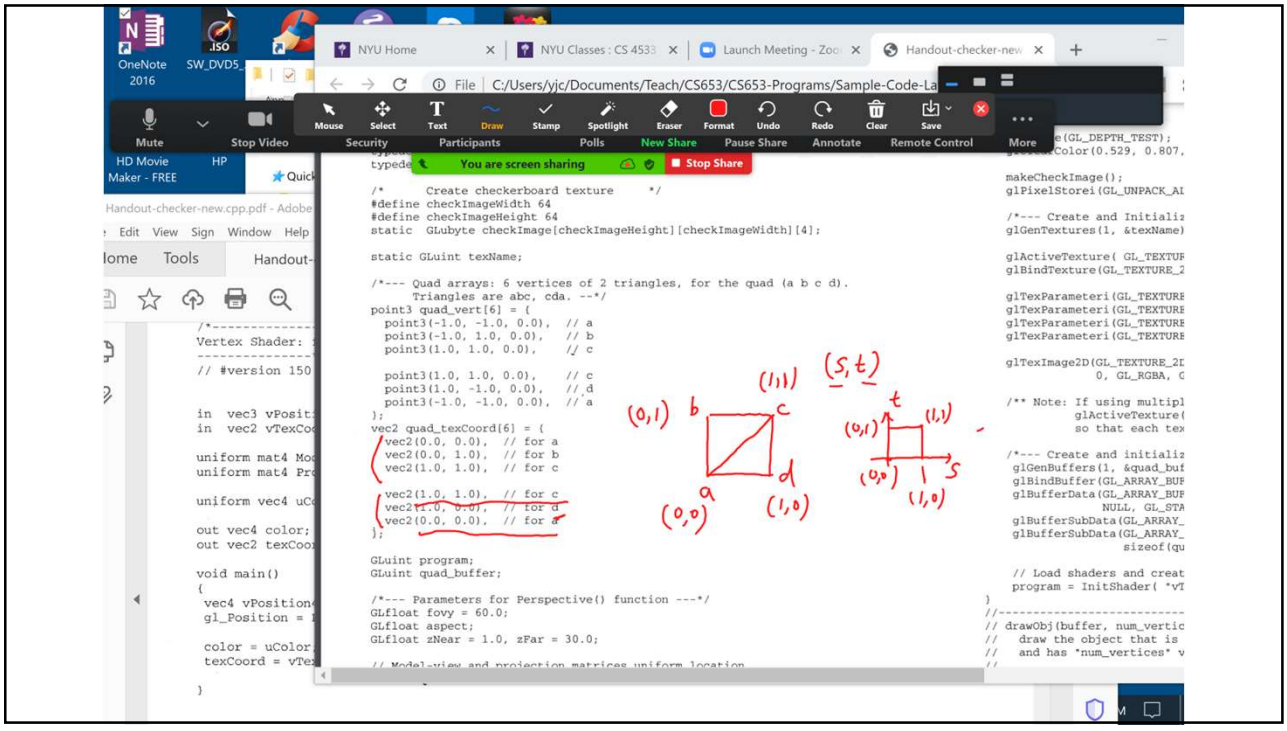
2. ~~Enable writing to z-buffer~~ X
   Draw shadow (to Both buffers. z-buffer, frame buffer
   shadow is NOT blocked by ground. so is drawn on top of ground)
   (only to frame buffer)
   shadow : on top of ground.
   shadow Δ's are NOT in z-buffer ⇒ they are all drawn & blended. ok. ✓ to block each other

3. (Restore ground into z-buffer)
   Enable writing to z-buffer.
   Disable writing to frame buffer
   Draw ground (only to z-buffer)
   optional [Draw shadow (= = =)]

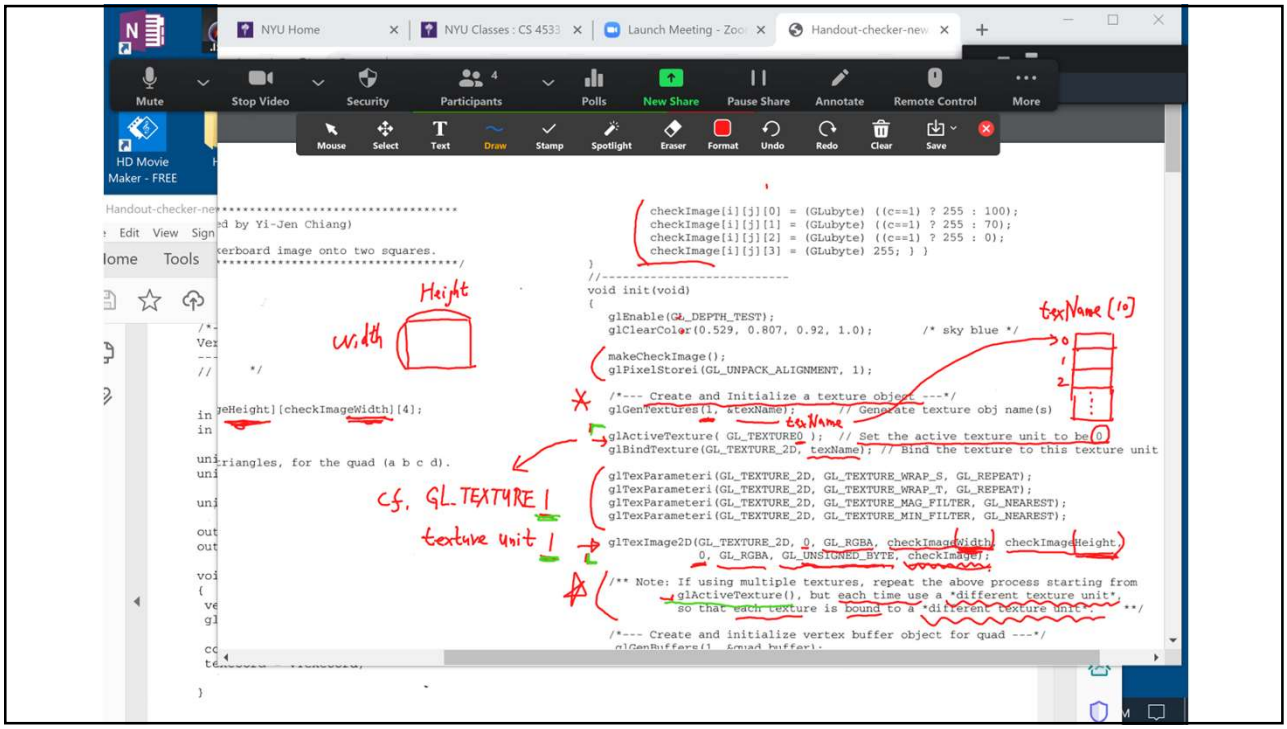
4. Enable writing to frame buffer.
   Resume normal operations.

T : critical section.

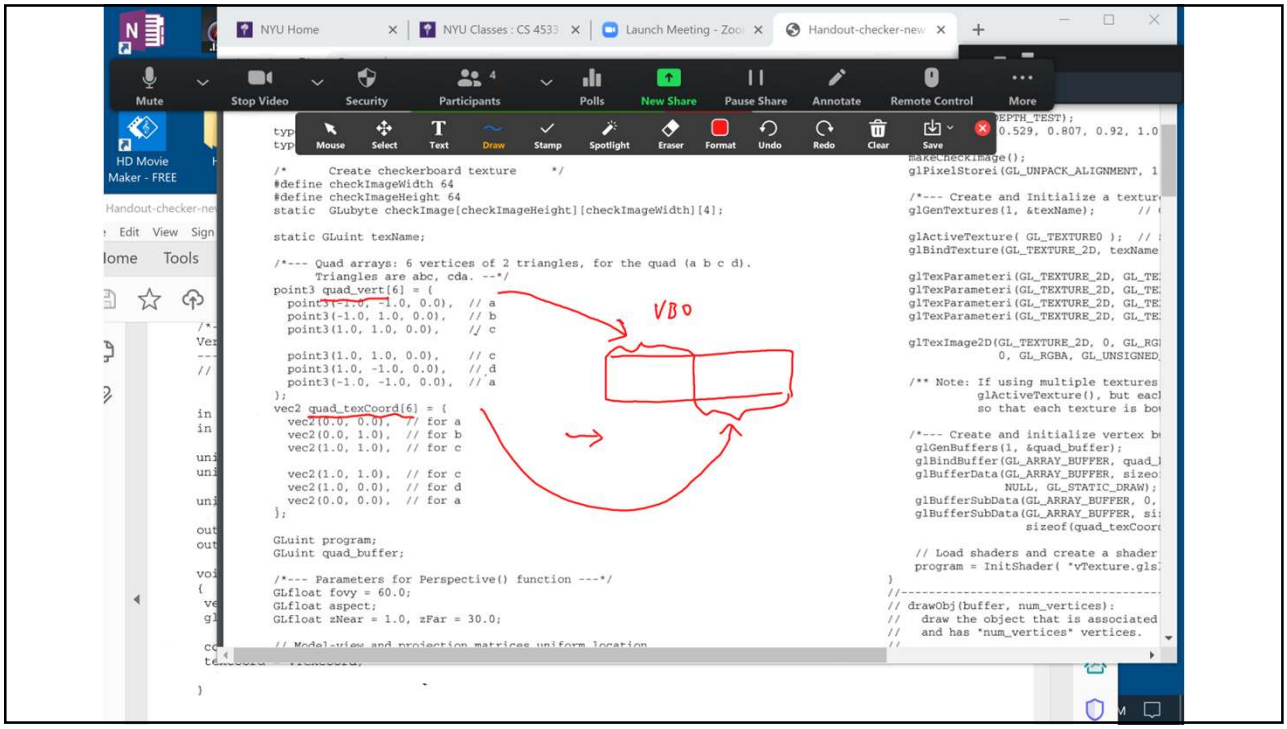
3



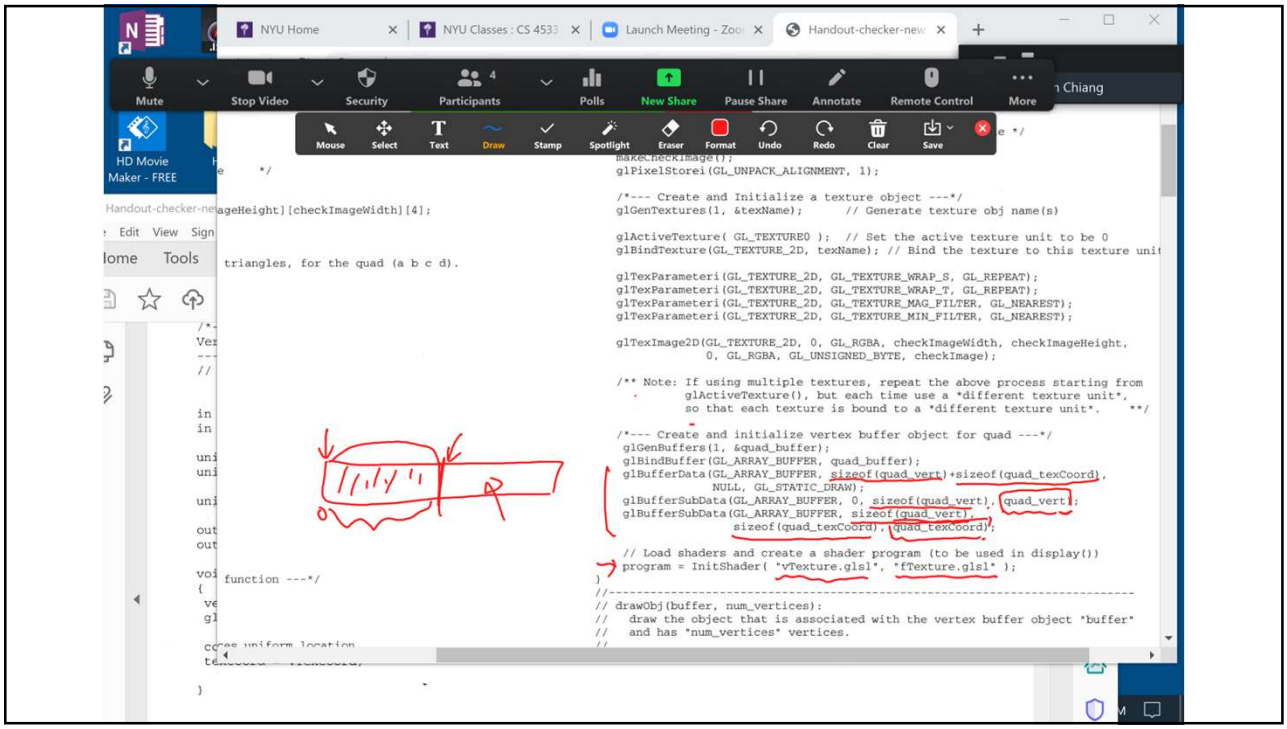
4


NYU Home
NYU Classes : CS 4533
Launch Meeting - Zoo
Handout-checker-new
Mute
Stop Video
Security
Participants
Polls
New Share
Pause Share
Annotate
Remote Control
More
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
HD Movie Maker - FREE
Edit View Sign
Home Tools
Height
width
texName (10)
cf. GL_TEXTURE1
texture unit 1
checkImage[i][j][0] = (GLubyte) ((c==1) ? 255 : 100);
checkImage[i][j][1] = (GLubyte) ((c==1) ? 255 : 70);
checkImage[i][j][2] = (GLubyte) ((c==1) ? 255 : 0);
checkImage[i][j][3] = (GLubyte) 255; } }
void init(void)
glEnable(GL_DEPTH_TEST);
glClearColor(0.529, 0.807, 0.92, 1.0); /* sky blue */
makeCheckImage();
glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
/*--- Create and Initialize a texture object ---*/
glGenTextures(1, &texName); // Generate texture obj name(s)
glActiveTexture(GL_TEXTURE0); // Set the active texture unit to be 0
glBindTexture(GL_TEXTURE_2D, texName); // Bind the texture to this texture unit
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, checkImageWidth, checkImageHeight,
0, GL_RGBA, GL_UNSIGNED_BYTE, checkImage);
/** Note: If using multiple textures, repeat the above process starting from
glActiveTexture(), but each time use a *different texture unit*,
so that each texture is bound to a *different texture unit*. **/
/*--- Create and initialize vertex buffer object for quad ---*/

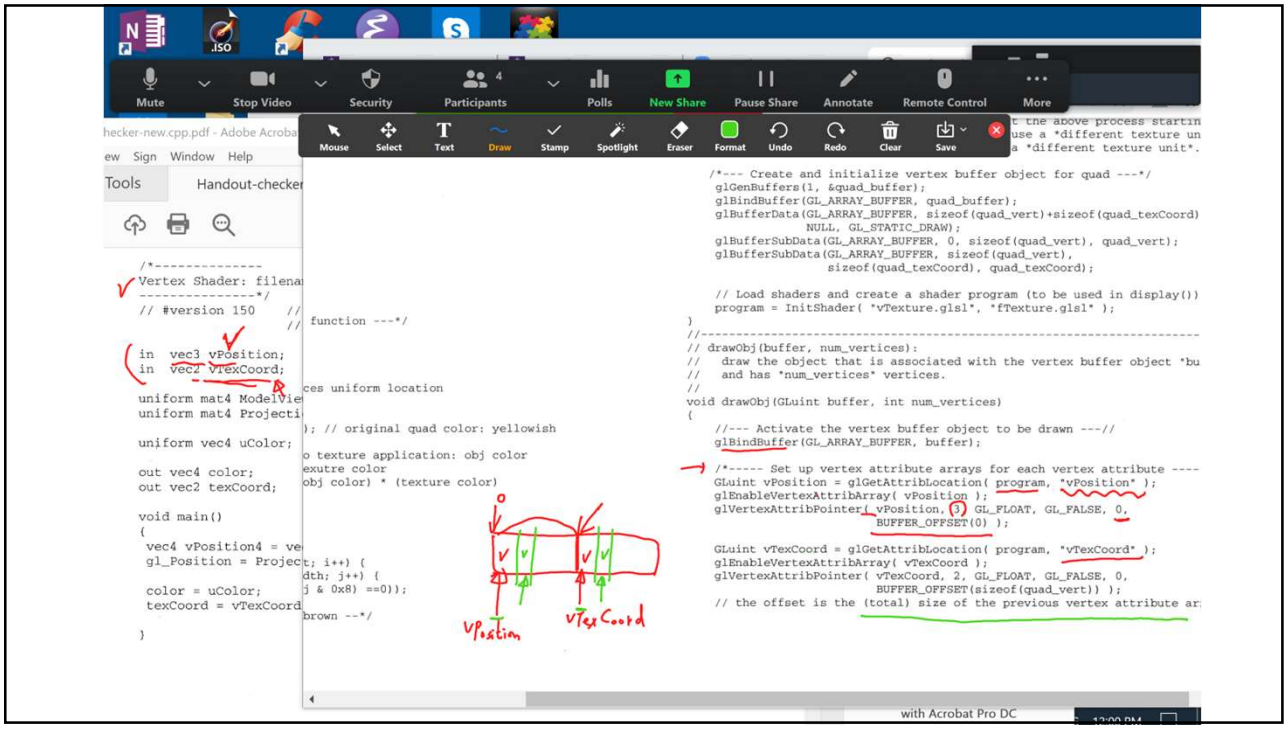
5


NYU Home
NYU Classes : CS 4533
Launch Meeting - Zoo
Handout-checker-new
Mute
Stop Video
Security
Participants
Polls
New Share
Pause Share
Annotate
Remote Control
More
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
HD Movie Maker - FREE
Edit View Sign
Home Tools
VBO
/* Create checkerboard texture */
#define checkImageWidth 64
#define checkImageHeight 64
static GLubyte checkImage[checkImageHeight][checkImageWidth][4];
static GLuint texName;
/*--- Quad arrays: 6 vertices of 2 triangles, for the quad (a b c d).
Triangles are abc, cda. --*/
point3 quad_vert[6] = {
point3(-1.0, -1.0, 0.0), // a
point3(-1.0, 1.0, 0.0), // b
point3(1.0, 1.0, 0.0), // c
point3(1.0, 1.0, 0.0), // c
point3(1.0, -1.0, 0.0), // d
point3(-1.0, -1.0, 0.0), // a
};
vec2 quad_texCoord[6] = {
vec2(0.0, 0.0), // for a
vec2(0.0, 1.0), // for b
vec2(1.0, 1.0), // for c
vec2(1.0, 1.0), // for c
vec2(1.0, 0.0), // for d
vec2(0.0, 0.0), // for a
};
GLuint program;
GLuint quad_buffer;
/*--- Parameters for Perspective() function ---*/
GLfloat fovy = 60.0;
GLfloat aspect;
GLfloat zNear = 1.0, zFar = 30.0;
// drawObj(buffer, num_vertices):
// draw the object that is associated
// and has "num_vertices" vertices.

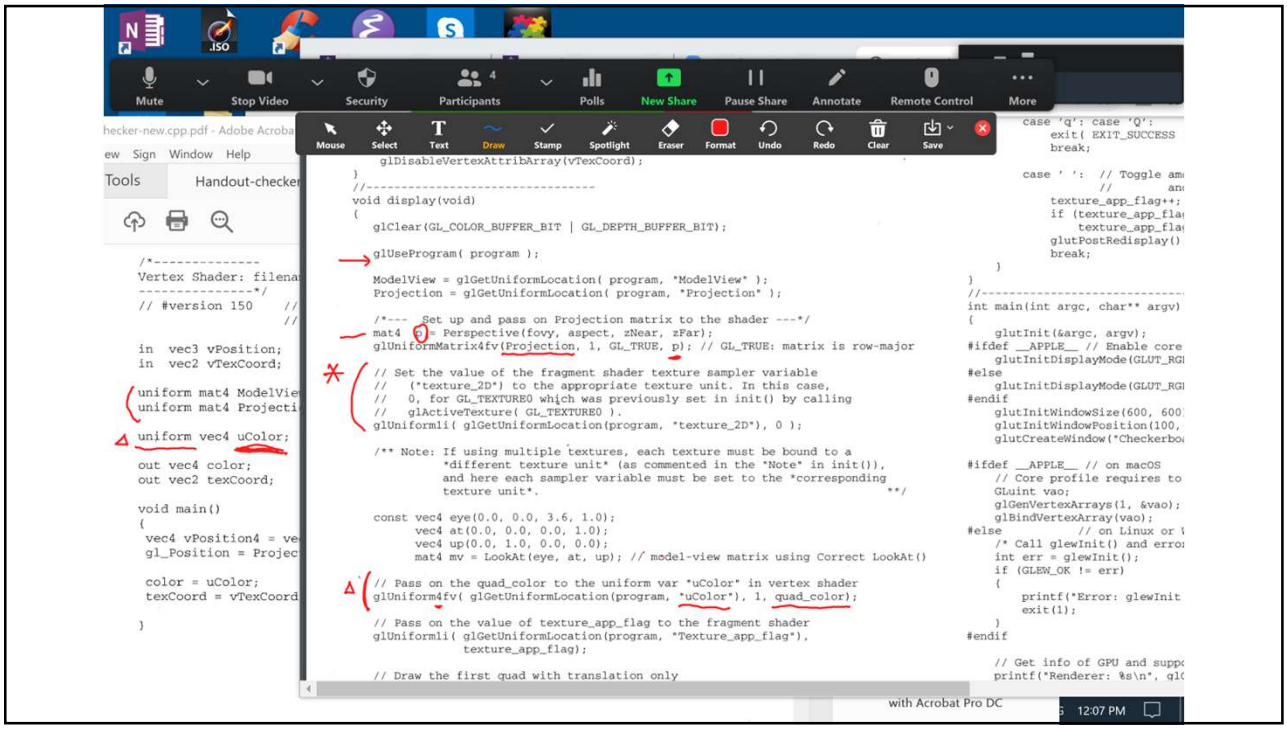
6


NYU Home
NYU Classes : CS 4533
Launch Meeting - Zoo
Handout-checker-new
Mute
Stop Video
Security
Participants
Polls
New Share
Pause Share
Annotate
Remote Control
More
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
HD Movie Maker - FREE
triangles, for the quad (a b c d).
function ---*/
glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
/*--- Create and Initialize a texture object ---*/
glGenTextures(1, &texName); // Generate texture obj name(s)
glActiveTexture(GL_TEXTURE0); // Set the active texture unit to be 0
glBindTexture(GL_TEXTURE_2D, texName); // Bind the texture to this texture unit
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, checkImageWidth, checkImageHeight,
0, GL_RGBA, GL_UNSIGNED_BYTE, checkImage);
/** Note: If using multiple textures, repeat the above process starting from
glActiveTexture(), but each time use a *different texture unit*,
so that each texture is bound to a *different texture unit*. **/
/*--- Create and initialize vertex buffer object for quad ---*/
glGenBuffers(1, &quad_buffer);
glBindBuffer(GL_ARRAY_BUFFER, quad_buffer);
glBufferData(GL_ARRAY_BUFFER, sizeof(quad_vert)+sizeof(quad_texCoord),
NULL, GL_STATIC_DRAW);
glBufferSubData(GL_ARRAY_BUFFER, 0, sizeof(quad_vert), quad_vert);
glBufferSubData(GL_ARRAY_BUFFER, sizeof(quad_vert),
sizeof(quad_texCoord), quad_texCoord);
// Load shaders and create a shader program (to be used in display())
program = InitShader("vTexture.glsl", "fTexture.glsl");
}
//----------
// drawObj(buffer, num_vertices):
// draw the object that is associated with the vertex buffer object "buffer"
// and has "num_vertices" vertices.

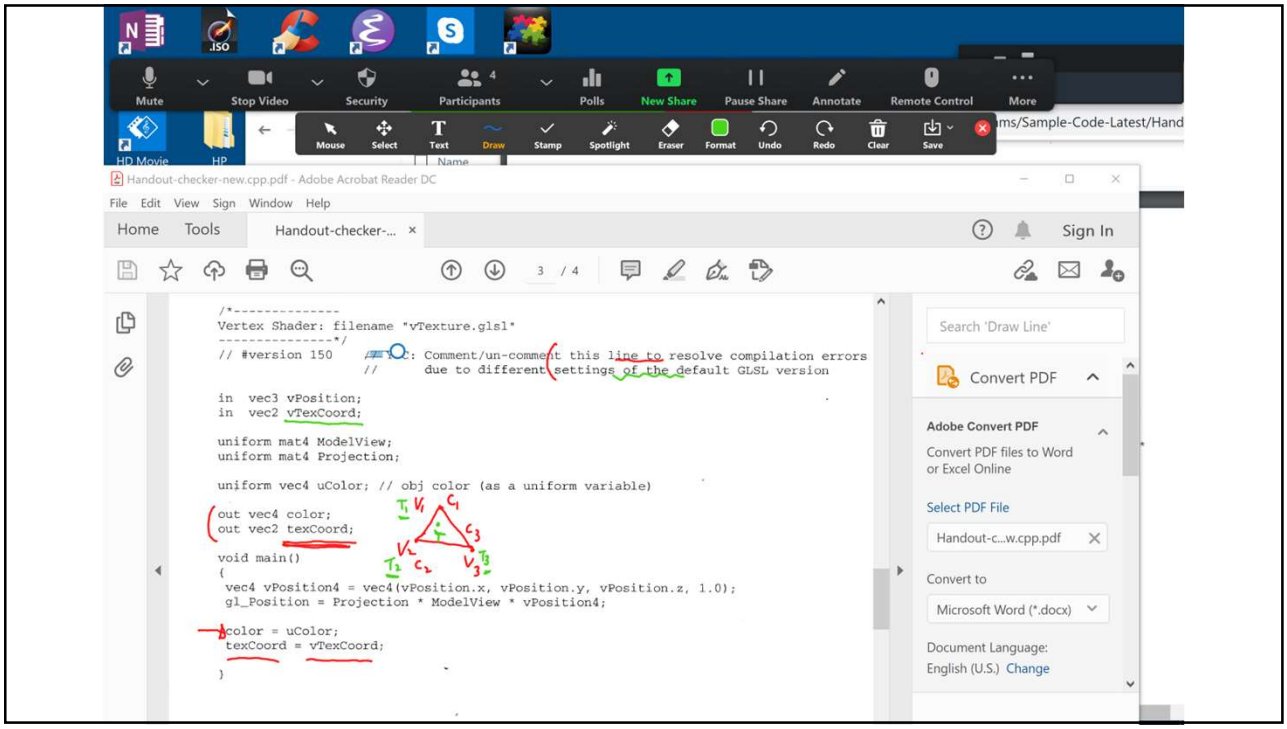
7


Mute
Stop Video
Security
Participants
Polls
New Share
Pause Share
Annotate
Remote Control
More
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
Vertex Shader: filena
// #version 150
in vec3 vPosition;
in vec2 vTexCoord;
uniform mat4 ModelVie
uniform mat4 Projecti
uniform vec4 uColor;
out vec4 color;
out vec2 texCoord;
void main()
{
vec4 vPosition4 = ve
gl_Position = Project
color = uColor;
texCoord = vTexCoord
}
function ---*/
); // original quad color: yellowish
o texture application: obj color
exutre color
obj color) * (texture color)
brown --*/
/*--- Create and initialize vertex buffer object for quad ---*/
glGenBuffers(1, &quad_buffer);
glBindBuffer(GL_ARRAY_BUFFER, quad_buffer);
glBufferData(GL_ARRAY_BUFFER, sizeof(quad_vert)+sizeof(quad_texCoord)
NULL, GL_STATIC_DRAW);
glBufferSubData(GL_ARRAY_BUFFER, 0, sizeof(quad_vert), quad_vert);
glBufferSubData(GL_ARRAY_BUFFER, sizeof(quad_vert),
sizeof(quad_texCoord), quad_texCoord);
// Load shaders and create a shader program (to be used in display())
program = InitShader("vTexture.glsl", "fTexture.glsl");
}
//----------
// drawObj(buffer, num_vertices):
// draw the object that is associated with the vertex buffer object "bu
// and has "num_vertices" vertices.
void drawObj(GLuint buffer, int num_vertices)
{
//--- Activate the vertex buffer object to be drawn ---//
glBindBuffer(GL_ARRAY_BUFFER, buffer);
/*----- Set up vertex attribute arrays for each vertex attribute ----
GLuint vPosition = glGetAttribLocation(program, "vPosition");
glEnableVertexAttribArray(vPosition);
glVertexAttribPointer(vPosition, 3, GL_FLOAT, GL_FALSE, 0,
BUFFER_OFFSET(0));
GLuint vTexCoord = glGetAttribLocation(program, "vTexCoord");
glEnableVertexAttribArray(vTexCoord);
glVertexAttribPointer(vTexCoord, 2, GL_FLOAT, GL_FALSE, 0,
BUFFER_OFFSET(sizeof(quad_vert)));
// the offset is the (total) size of the previous vertex attribute ar
vPosition
vTexCoord
with Acrobat Pro DC

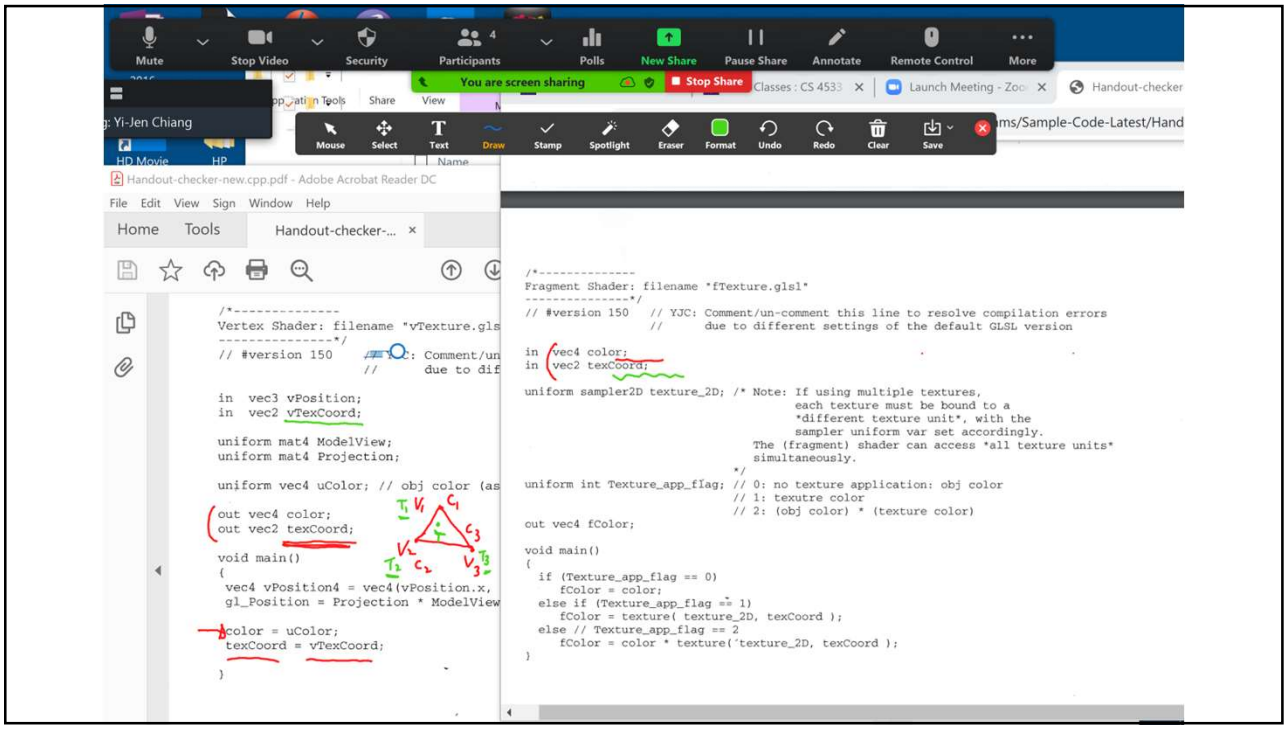
8


Mute
Stop Video
Security
Participants
Polls
New Share
Pause Share
Annotate
Remote Control
More
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
Tools
Handout-checker
/*---------------
Vertex Shader: filena
---------------*/
// #version 150
in vec3 vPosition;
in vec2 vTexCoord;
uniform mat4 ModelVie
uniform mat4 Projecti
uniform vec4 uColor;
out vec4 color;
out vec2 texCoord;
void main()
{
vec4 vPosition4 = ve
gl_Position = Projec
color = uColor;
texCoord = vTexCoord
}
glDisableVertexAttribArray(vTexCoord);
}
//---------------------------------
void display(void)
{
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
glUseProgram(program);
ModelView = glGetUniformLocation(program, "ModelView");
Projection = glGetUniformLocation(program, "Projection");
/*--- Set up and pass on Projection matrix to the shader ---*/
mat4 p = Perspective(fovy, aspect, zNear, zFar);
glUniformMatrix4fv(Projection, 1, GL_TRUE, p); // GL_TRUE: matrix is row-major
// Set the value of the fragment shader texture sampler variable
// ("texture_2D") to the appropriate texture unit. In this case,
// 0, for GL_TEXTURE0 which was previously set in init() by calling
// glActiveTexture(GL_TEXTURE0).
glUniform1i(glGetUniformLocation(program, "texture_2D"), 0);
/** Note: If using multiple textures, each texture must be bound to a
"different texture unit" (as commented in the "Note" in init()),
and here each sampler variable must be set to the "corresponding
texture unit". **/
const vec4 eye(0.0, 0.0, 3.6, 1.0);
vec4 at(0.0, 0.0, 0.0, 1.0);
vec4 up(0.0, 1.0, 0.0, 0.0);
mat4 mv = LookAt(eye, at, up); // model-view matrix using Correct LookAt()
// Pass on the quad_color to the uniform var "uColor" in vertex shader
glUniform4fv(glGetUniformLocation(program, "uColor"), 1, quad_color);
// Pass on the value of texture_app_flag to the fragment shader
glUniform1i(glGetUniformLocation(program, "Texture_app_flag"),
texture_app_flag);
// Draw the first quad with translation only
case 'q': case 'Q':
exit(EXIT_SUCCESS
break;
case ' ': // Toggle am
texture_app_flag++;
if (texture_app_fla
texture_app_fla
glutPostRedisplay()
break;
}
}
//---------------------------------
int main(int argc, char** argv)
{
glutInit(&argc, argv);
#ifdef __APPLE__ // Enable core
glutInitDisplayMode(GLUT_RGB
#else
glutInitDisplayMode(GLUT_RGB
#endif
glutInitWindowSize(600, 600
glutInitWindowPosition(100,
glutCreateWindow("Checkerboa
#ifdef __APPLE__ // on macOS
// Core profile requires to
GLuint vao;
glGenVertexArrays(1, &vao);
glBindVertexArray(vao);
#else // on Linux or
/* Call glewInit() and erro
int err = glewInit();
if (GLEW_OK != err)
{
printf("Error: glewInit
exit(1);
}
#endif
// Get info of GPU and suppo
printf("Renderer: %s\n", gl
with Acrobat Pro DC
12:07 PM

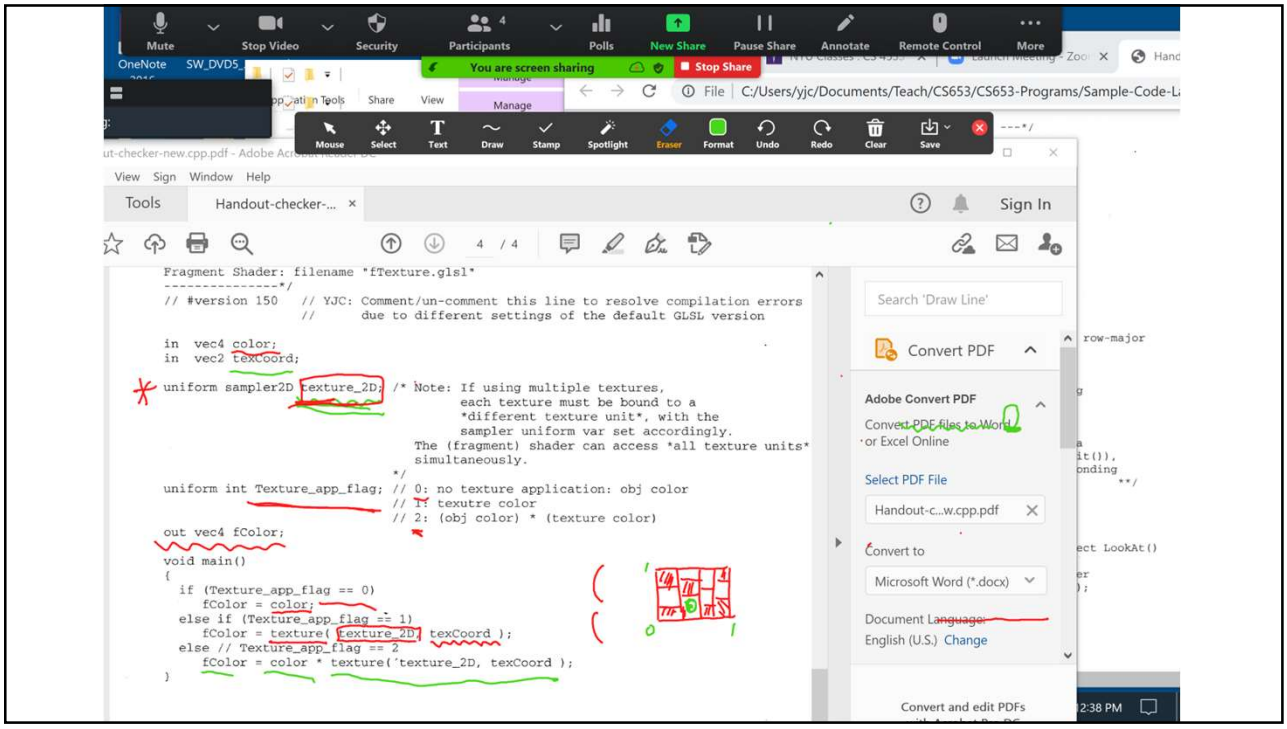
9


Mute
Stop Video
Security
Participants
Polls
New Share
Pause Share
Annotate
Remote Control
More
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
HD Movie
HP
Handout-checker-new.cpp.pdf - Adobe Acrobat Reader DC
File Edit View Sign Window Help
Home Tools
Handout-checker-...
3 / 4
Sign In
/*---------------
Vertex Shader: filename "vTexture.glsl"
---------------*/
// #version 150 // C: Comment/un-comment this line to resolve compilation errors
// due to different settings of the default GLSL version
in vec3 vPosition;
in vec2 vTexCoord;
uniform mat4 ModelView;
uniform mat4 Projection;
uniform vec4 uColor; // obj color (as a uniform variable)
out vec4 color;
out vec2 texCoord;
void main()
{
vec4 vPosition4 = vec4(vPosition.x, vPosition.y, vPosition.z, 1.0);
gl_Position = Projection * ModelView * vPosition4;
color = uColor;
texCoord = vTexCoord;
}
Search 'Draw Line'
Convert PDF
Adobe Convert PDF
Convert PDF files to Word or Excel Online
Select PDF File
Handout-c...w.cpp.pdf
Convert to
Microsoft Word (*.docx)
Document Language:
English (U.S.) Change

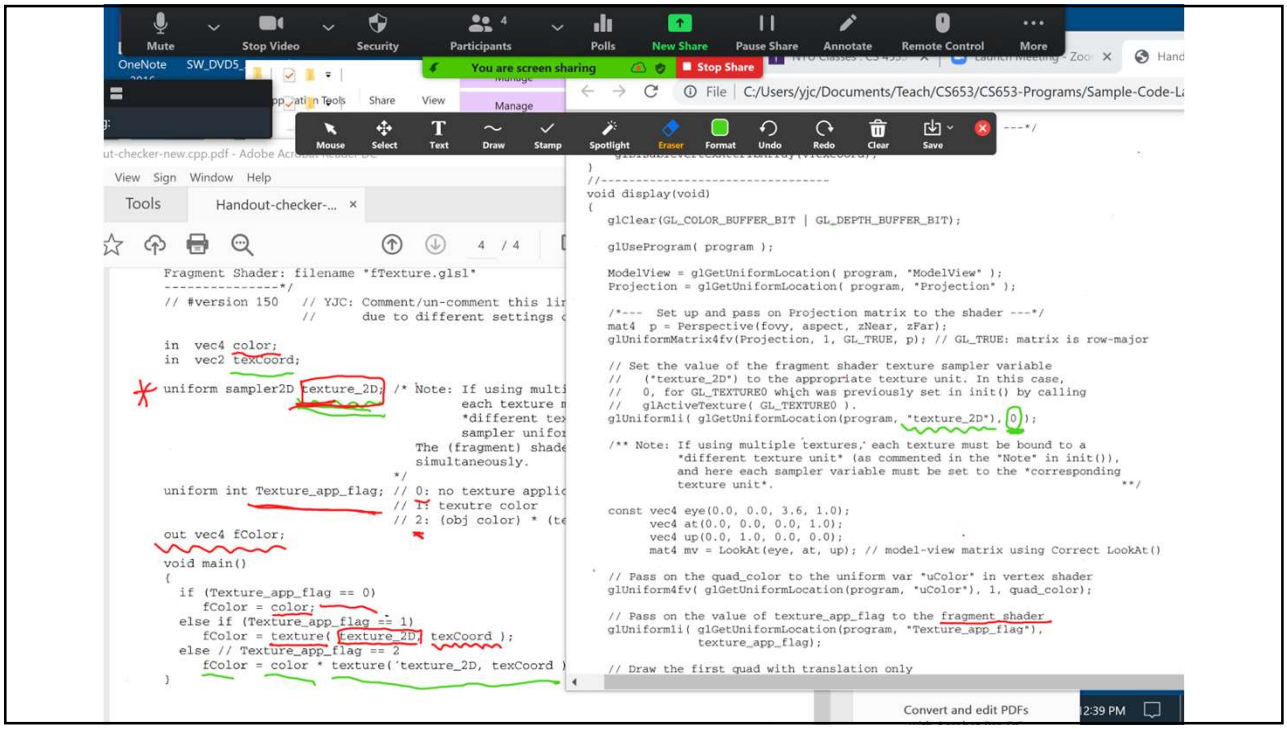
10


Mute
Stop Video
Security
Participants
Polls
New Share
Pause Share
Annotate
Remote Control
More
You are screen sharing
Stop Share
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
Handout-checker-new.cpp.pdf - Adobe Acrobat Reader DC
File Edit View Sign Window Help
Home Tools
Handout-checker-...
/*--------------
Vertex Shader: filename "vTexture.gls
--------------*/
// #version 150
// YJC: Comment/un
due to dif
in vec3 vPosition;
in vec2 vTexCoord;
uniform mat4 ModelView;
uniform mat4 Projection;
uniform vec4 uColor; // obj color (as
out vec4 color;
out vec2 texCoord;
void main()
{
vec4 vPosition4 = vec4(vPosition.x,
gl_Position = Projection * ModelView
color = uColor;
texCoord = vTexCoord;
}
/*--------------
Fragment Shader: filename "fTexture.glsl"
--------------*/
// #version 150 // YJC: Comment/un-comment this line to resolve compilation errors
// due to different settings of the default GLSL version
in vec4 color;
in vec2 texCoord;
uniform sampler2D texture_2D; /* Note: If using multiple textures,
each texture must be bound to a
different texture unit, with the
sampler uniform var set accordingly.
The (fragment) shader can access *all texture units*
simultaneously.
*/
uniform int Texture_app_flag; // 0: no texture application: obj color
// 1: texutre color
// 2: (obj color) * (texture color)
out vec4 fColor;
void main()
{
if (Texture_app_flag == 0)
fColor = color;
else if (Texture_app_flag == 1)
fColor = texture(texture_2D, texCoord);
else // Texture_app_flag == 2
fColor = color * texture(texture_2D, texCoord);
}

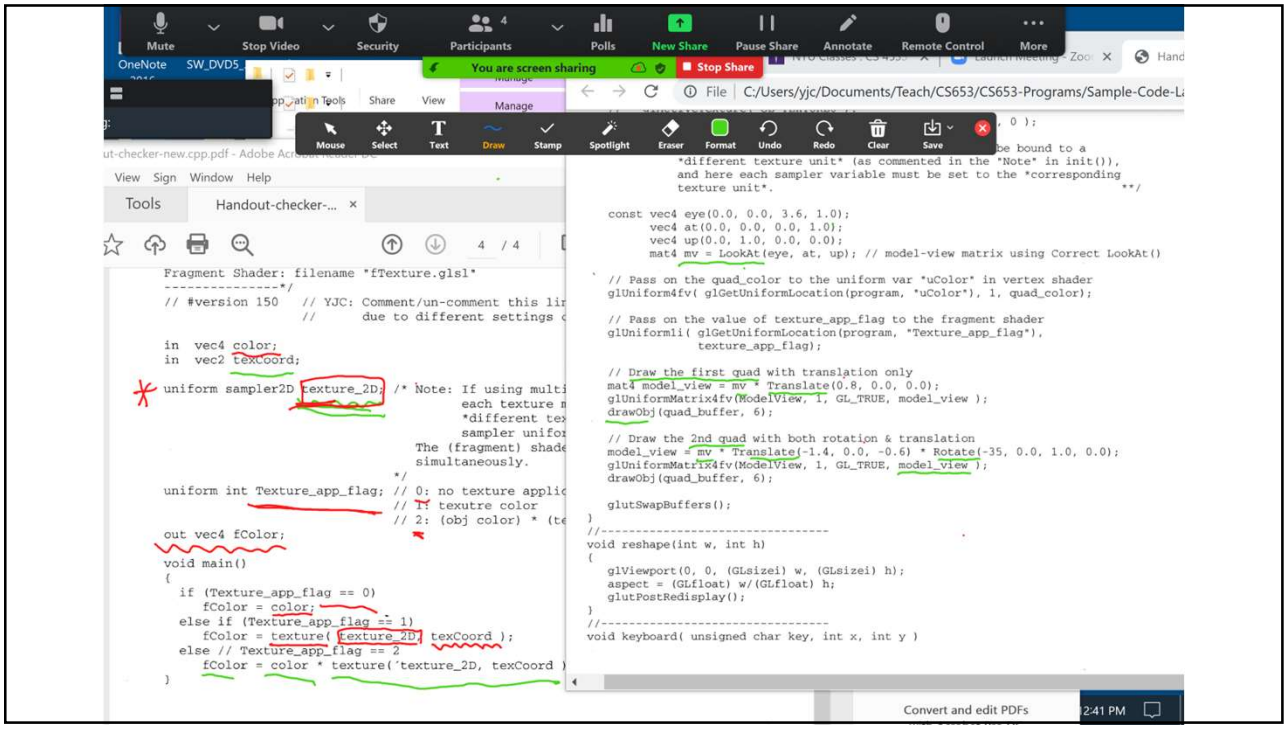
11


Mute
Stop Video
Security
Participants
Polls
New Share
Pause Share
Annotate
Remote Control
More
You are screen sharing
Stop Share
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
Tools
Handout-checker-...
Sign In
Fragment Shader: filename "fTexture.glsl"
--------------*/
// #version 150 // YJC: Comment/un-comment this line to resolve compilation errors
// due to different settings of the default GLSL version
in vec4 color;
in vec2 texCoord;
uniform sampler2D texture_2D; /* Note: If using multiple textures,
each texture must be bound to a
different texture unit, with the
sampler uniform var set accordingly.
The (fragment) shader can access *all texture units*
simultaneously.
*/
uniform int Texture_app_flag; // 0: no texture application: obj color
// 1: texutre color
// 2: (obj color) * (texture color)
out vec4 fColor;
void main()
{
if (Texture_app_flag == 0)
fColor = color;
else if (Texture_app_flag == 1)
fColor = texture(texture_2D, texCoord);
else // Texture_app_flag == 2
fColor = color * texture(texture_2D, texCoord);
}
Search 'Draw Line'
Convert PDF
Adobe Convert PDF
Convert PDF files to Word or Excel Online
Select PDF File
Handout-c...w.cpp.pdf
Convert to
Microsoft Word (*.docx)
Document Language:
English (U.S.) Change
Convert and edit PDFs

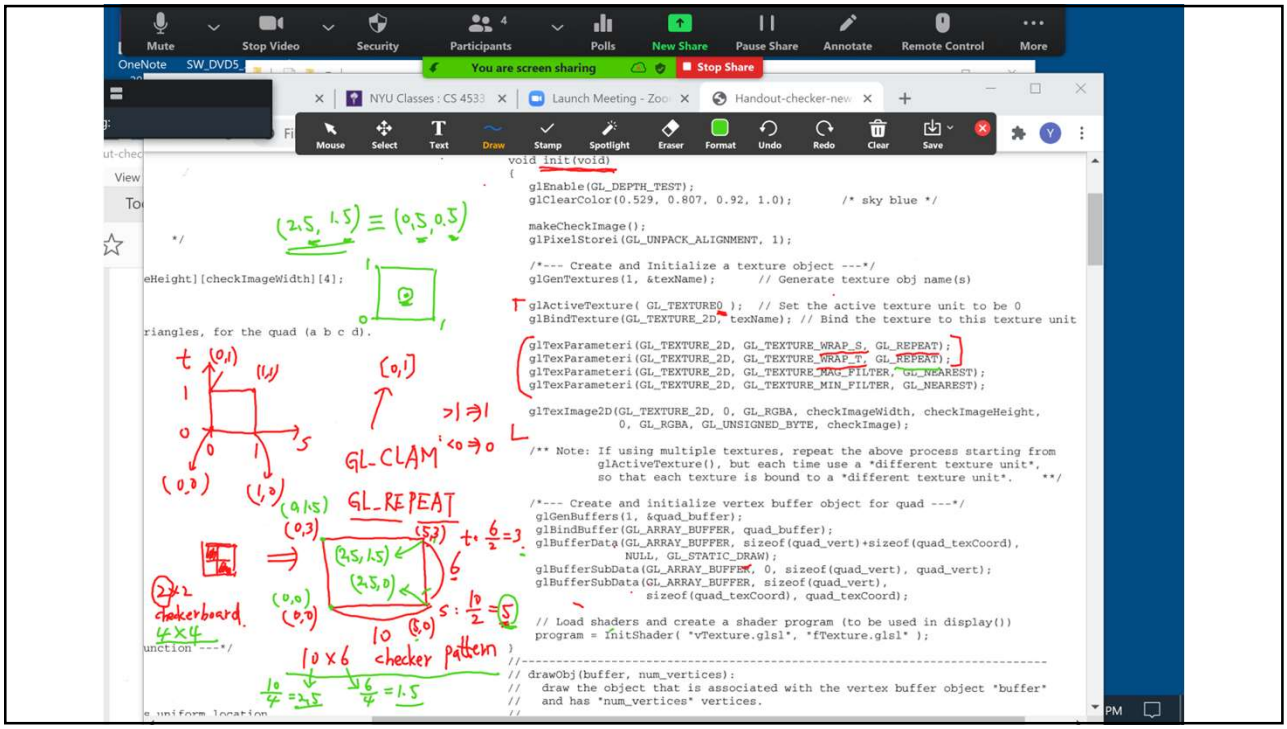
12


Mute
Stop Video
Security
Participants
Polls
New Share
Pause Share
Annotate
Remote Control
More
You are screen sharing
Stop Share
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
File | C:/Users/yjc/Documents/Teach/CS653/CS653-Programs/Sample-Code-L
View Sign Window Help
Tools
Handout-checker-...
4 / 4
Fragment Shader: filename "fTexture.gls1"
// #version 150
in vec4 color;
in vec2 texCoord;
uniform sampler2D texture_2D; /* Note: If using multi
each texture
*different tex
sampler unifo
The (fragment) shade
simultaneously.
*/
uniform int Texture_app_flag; // 0: no texture applic
// 1: texutre color
// 2: (obj color) * (te
out vec4 fColor;
void main()
{
if (Texture_app_flag == 0)
fColor = color;
else if (Texture_app_flag == 1)
fColor = texture(texture_2D, texCoord);
else // Texture_app_flag == 2
fColor = color * texture(texture_2D, texCoord
}
void display(void)
{
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
glUseProgram(program);
ModelView = glGetUniformLocation(program, "ModelView");
Projection = glGetUniformLocation(program, "Projection");
/*--- Set up and pass on Projection matrix to the shader ---*/
mat4 p = Perspective(fovy, aspect, zNear, zFar);
glUniformMatrix4fv(Projection, 1, GL_TRUE, p); // GL_TRUE: matrix is row-major
// Set the value of the fragment shader texture sampler variable
// ("texture_2D") to the appropriate texture unit. In this case,
// 0, for GL_TEXTURE0 which was previously set in init() by calling
// glActiveTexture(GL_TEXTURE0).
glUniform1i(glGetUniformLocation(program, "texture_2D"), 0);
/** Note: If using multiple textures, each texture must be bound to a
different texture unit (as commented in the "Note" in init()),
and here each sampler variable must be set to the *corresponding
texture unit*. **/
const vec4 eye(0.0, 0.0, 3.6, 1.0);
vec4 at(0.0, 0.0, 0.0, 1.0);
vec4 up(0.0, 1.0, 0.0, 0.0);
mat4 mv = LookAt(eye, at, up); // model-view matrix using Correct LookAt()
// Pass on the quad_color to the uniform var "uColor" in vertex shader
glUniform4fv(glGetUniformLocation(program, "uColor"), 1, quad_color);
// Pass on the value of texture_app_flag to the fragment shader
glUniform1i(glGetUniformLocation(program, "Texture_app_flag"),
texture_app_flag);
// Draw the first quad with translation only
Convert and edit PDFs
2:39 PM

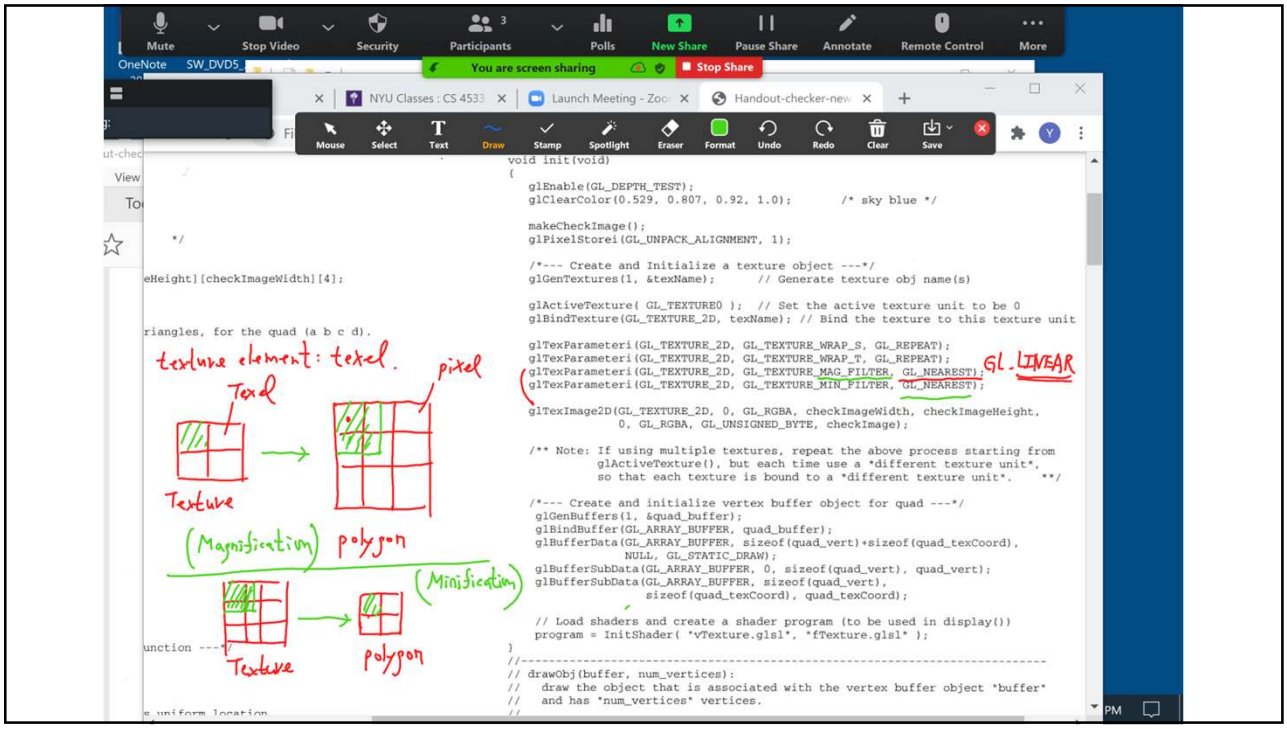
13


Mute
Stop Video
Security
Participants
Polls
New Share
Pause Share
Annotate
Remote Control
More
You are screen sharing
Stop Share
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
File | C:/Users/yjc/Documents/Teach/CS653/CS653-Programs/Sample-Code-L
View Sign Window Help
Tools
Handout-checker-...
4 / 4
Fragment Shader: filename "fTexture.gls1"
// #version 150
in vec4 color;
in vec2 texCoord;
uniform sampler2D texture_2D; /* Note: If using multi
each texture
*different tex
sampler unifo
The (fragment) shade
simultaneously.
*/
uniform int Texture_app_flag; // 0: no texture applic
// 1: texutre color
// 2: (obj color) * (te
out vec4 fColor;
void main()
{
if (Texture_app_flag == 0)
fColor = color;
else if (Texture_app_flag == 1)
fColor = texture(texture_2D, texCoord);
else // Texture_app_flag == 2
fColor = color * texture(texture_2D, texCoord
}
texture unit*. **/
const vec4 eye(0.0, 0.0, 3.6, 1.0);
vec4 at(0.0, 0.0, 0.0, 1.0);
vec4 up(0.0, 1.0, 0.0, 0.0);
mat4 mv = LookAt(eye, at, up); // model-view matrix using Correct LookAt()
// Pass on the quad_color to the uniform var "uColor" in vertex shader
glUniform4fv(glGetUniformLocation(program, "uColor"), 1, quad_color);
// Pass on the value of texture_app_flag to the fragment shader
glUniform1i(glGetUniformLocation(program, "Texture_app_flag"),
texture_app_flag);
// Draw the first quad with translation only
mat4 model_view = mv * Translate(0.8, 0.0, 0.0);
glUniformMatrix4fv(ModelView, 1, GL_TRUE, model_view);
drawObj(quad_buffer, 6);
// Draw the 2nd quad with both rotation & translation
model_view = mv * Translate(-1.4, 0.0, -0.6) * Rotate(-35, 0.0, 1.0, 0.0);
glUniformMatrix4fv(ModelView, 1, GL_TRUE, model_view);
drawObj(quad_buffer, 6);
glutSwapBuffers();
}
void reshape(int w, int h)
{
glViewport(0, 0, (GLsizei) w, (GLsizei) h);
aspect = (GLfloat) w/(GLfloat) h;
glutPostRedisplay();
}
void keyboard(unsigned char key, int x, int y)
Convert and edit PDFs
2:41 PM

14


Mute
Stop Video
Security
Participants
Polls
New Share
Pause Share
Annotate
Remote Control
More
You are screen sharing
Stop Share
NYU Classes : CS 4533
Launch Meeting - Zoo
Handout-checker-new
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
(2.5, 1.5) ≡ (0.5, 0.5)
GL-CLAM
GL_REPEAT
2x2 checkerboard
4x4
10 x 6 checker pattern
riangles, for the quad (a b c d).
unction ---*/
e uniform location
void init(void)
{
glEnable(GL_DEPTH_TEST);
glClearColor(0.529, 0.807, 0.92, 1.0); /* sky blue */
makeCheckImage();
glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
/*--- Create and Initialize a texture object ---*/
glGenTextures(1, &texName); // Generate texture obj name(s)
glActiveTexture(GL_TEXTURE0); // Set the active texture unit to be 0
glBindTexture(GL_TEXTURE_2D, texName); // Bind the texture to this texture unit
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, checkImageWidth, checkImageHeight,
0, GL_RGBA, GL_UNSIGNED_BYTE, checkImage);
/** Note: If using multiple textures, repeat the above process starting from
glActiveTexture(), but each time use a *different texture unit*,
so that each texture is bound to a *different texture unit*. **/
/*--- Create and initialize vertex buffer object for quad ---*/
glGenBuffers(1, &quad_buffer);
glBindBuffer(GL_ARRAY_BUFFER, quad_buffer);
glBufferData(GL_ARRAY_BUFFER, sizeof(quad_vert)+sizeof(quad_texCoord),
NULL, GL_STATIC_DRAW);
glBufferSubData(GL_ARRAY_BUFFER, 0, sizeof(quad_vert), quad_vert);
glBufferSubData(GL_ARRAY_BUFFER, sizeof(quad_vert),
sizeof(quad_texCoord), quad_texCoord);
// Load shaders and create a shader program (to be used in display())
program = InitShader("vTexture.glsl", "fTexture.glsl");
}
//----------
// drawObj(buffer, num_vertices):
// draw the object that is associated with the vertex buffer object "buffer"
// and has "num_vertices" vertices.
//
PM

15


Mute
Stop Video
Security
Participants
Polls
New Share
Pause Share
Annotate
Remote Control
More
You are screen sharing
Stop Share
NYU Classes : CS 4533
Launch Meeting - Zoo
Handout-checker-new
Mouse
Select
Text
Draw
Stamp
Spotlight
Eraser
Format
Undo
Redo
Clear
Save
texture element: texel.
Texel
pixel
Texture
(Magnification)
polygon
(Minification)
Texture
polygon
GL_LINEAR
riangles, for the quad (a b c d).
unction ---*/
e uniform location
void init(void)
{
glEnable(GL_DEPTH_TEST);
glClearColor(0.529, 0.807, 0.92, 1.0); /* sky blue */
makeCheckImage();
glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
/*--- Create and Initialize a texture object ---*/
glGenTextures(1, &texName); // Generate texture obj name(s)
glActiveTexture(GL_TEXTURE0); // Set the active texture unit to be 0
glBindTexture(GL_TEXTURE_2D, texName); // Bind the texture to this texture unit
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, checkImageWidth, checkImageHeight,
0, GL_RGBA, GL_UNSIGNED_BYTE, checkImage);
/** Note: If using multiple textures, repeat the above process starting from
glActiveTexture(), but each time use a *different texture unit*,
so that each texture is bound to a *different texture unit*. **/
/*--- Create and initialize vertex buffer object for quad ---*/
glGenBuffers(1, &quad_buffer);
glBindBuffer(GL_ARRAY_BUFFER, quad_buffer);
glBufferData(GL_ARRAY_BUFFER, sizeof(quad_vert)+sizeof(quad_texCoord),
NULL, GL_STATIC_DRAW);
glBufferSubData(GL_ARRAY_BUFFER, 0, sizeof(quad_vert), quad_vert);
glBufferSubData(GL_ARRAY_BUFFER, sizeof(quad_vert),
sizeof(quad_texCoord), quad_texCoord);
// Load shaders and create a shader program (to be used in display())
program = InitShader("vTexture.glsl", "fTexture.glsl");
}
//----------
// drawObj(buffer, num_vertices):
// draw the object that is associated with the vertex buffer object "buffer"
// and has "num_vertices" vertices.
//
PM

16