Final Project Report

Wen Liang
(Wen_Liang@student.uml.edu)

Content

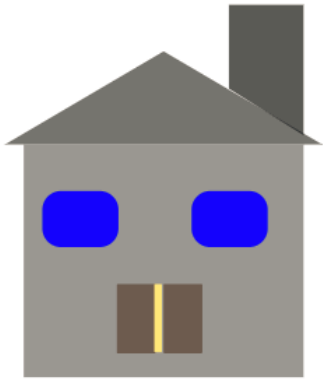
My final project is using WegGL to allow user to edit/change the 3D transformation, texture, projection, multiple view point(including zoom in/zoom out), camera sources, lighting effect, animation of my 3D object etc. And it also allows user to use filter to have different vision effect(e.g. the “emboss” mode of “Filter” can even has bump-mapping-like effect).

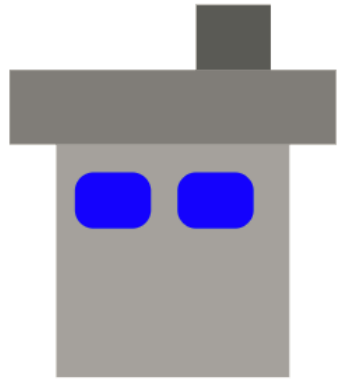
What I did so far ?

- 1. Modeling: draw three 2D “elevation”, and apply 2D transformation
- 2. Transform object: apply 3D transformations to the created object.
- 3. Viewing: view your created object from multiple views.
- 4. Transform camera/viewer
- 5. Generate different projections of the objects
- 6. Edit/Change perspective projection vanishing points (1, 2, 3)
- 7. Create texture for the object
- 8. Add light source to 3D object and allow user to change the shininess
- 9. Animation of my 3D object.
- 10. Additional camera view point.
- 11. different texture option for the user
- 12. Add red light, green light, blue light lighting effect to my 3D Object (Arbitrary combination of light)
- 13. Use the image processing technique and different filter kernel to generate new image effect. (e.g. the “emboss” mode of “Filter” can even has bump-mapping-like effect)

Week1

- **1. Modeling: draw three 2D “elevation”, and apply 2D transformation**





WEEK1

- **1. Modeling: draw three 2D “elevation”, and apply 2D transformation**

Use the svg to draw the 2D “elevation”, and use the svg build in transform API to finish the 2D transformation of the “elevation”

Week2

- 2. Transform object: apply 3D transformations to the created object.
- 3. Viewing: view your created object from multiple views.
- 4. Transform camera/viewer
- 5. Generate different projections of the objects
- 6. Edit/Change perspective projection vanishing points (1, 2, 3)
- 7. Create texture for the object

Week2

Project_week2 Wen Liang(Wen_Liang@student.uml.edu)

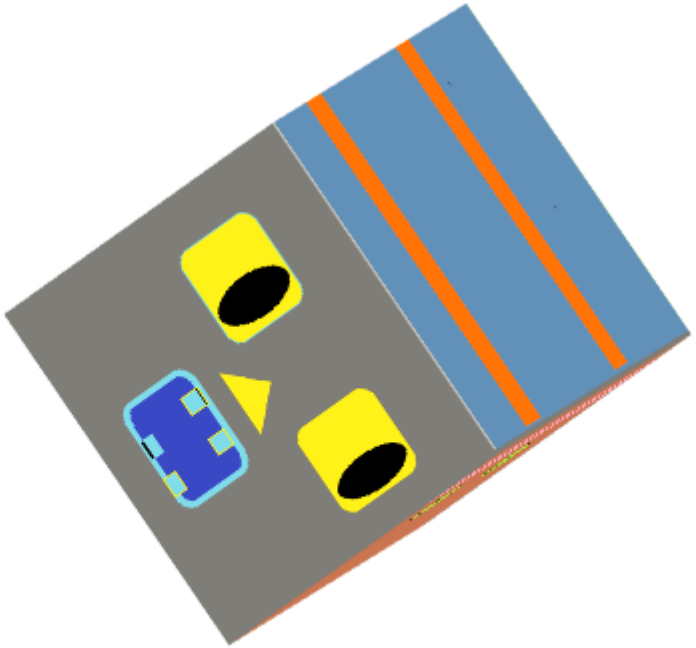
Perspective View: 1 point

x	0
y	260
z	400
angleX	0
angleY	0
angleZ	0
scaleX	1.00
scaleY	1.00
scaleZ	1.00
shearX	0.00
shearY	0.00
fieldOfView	60
cameraAngle	0

Week2

Project_week2 Wen Liang(Wen_Liang@student.uml.edu)

Parallel View: oblique



x	892
y	386
z	0
angleX	20
angleY	29
angleZ	50
scaleX	1.00
scaleY	1.00
scaleZ	1.00
shearX	0.00
shearY	0.00

WEEK2

- 2. **Transform object: apply 3D (Translate/Rotate/Scale/SHear) transformations to the created object.**

Use the transformation matrix to define transform function, and pass the translate/rotate/scale/shear value in the “range slide” to the function and multiply it with the projection matrix to get the final projection matrix.

WEEK2

- **3. Viewing: view your created object from multiple views.**

Change the initial translation/scale/shear/rotation value to control the position and orientation of the 3D object to conform to the definition of the different projection, as well as to transform camera/view point.

WEEK2

- 4. **Transform camera/viewer**

Primarily, I change the camera/view point through the perspective projection matrix and the basic 3D transformation, or define a camera position, calculate camera matrix and inverse the camera matrix to get the view matrix to achieve the purpose of the change of view point.

WEEK2

- 5. **Generate different projections of the objects (refer to class discussions about different projections)**

Change the initial translation/scale/shear/rotation value to control the position and orientation of the 3D object to conform to the definition of the different projection.

WEEK2

- 6. **Edit/Change perspective projection vanishing points (1, 2, 3).**

Change the initial translation/scale/shear/rotation value to control the position and orientation of perspective projection of the 3D object to conform to the definition of the different perspective projection vanishing points.

WEEK2

- 7. **Create texture for the object.**

I draw the image first, then put the image on each side of the 3D Object through WebGL build in texture mapping API

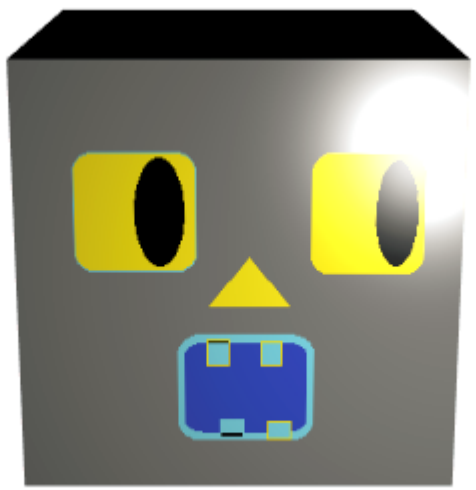
Week 3

- **8. Add light source to 3D object and allow user to change the shininess**
- **9. Animation of my 3D object.**
- **10. Additional camera view point.**

Week 3

Project_week3 Wen Liang(Wen_Liang@student.uml.edu)

Perspective View: 1 point

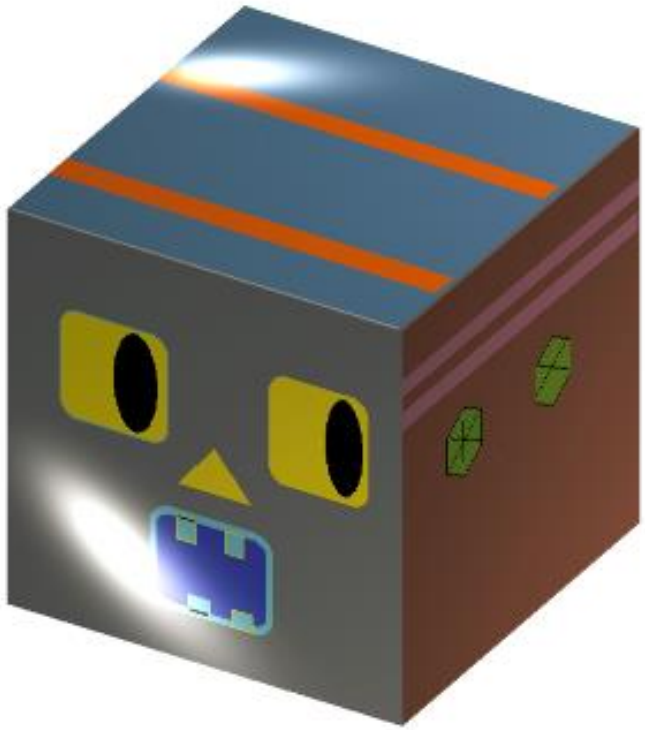


x	0
y	115
z	400
angleX	0
angleY	0
angleZ	0
scaleX	1.00
scaleY	1.00
scaleZ	1.00
shearX	0.00
shearY	0.00
fieldOfView	60
cameraAngle	0
shininess	150

Week 3

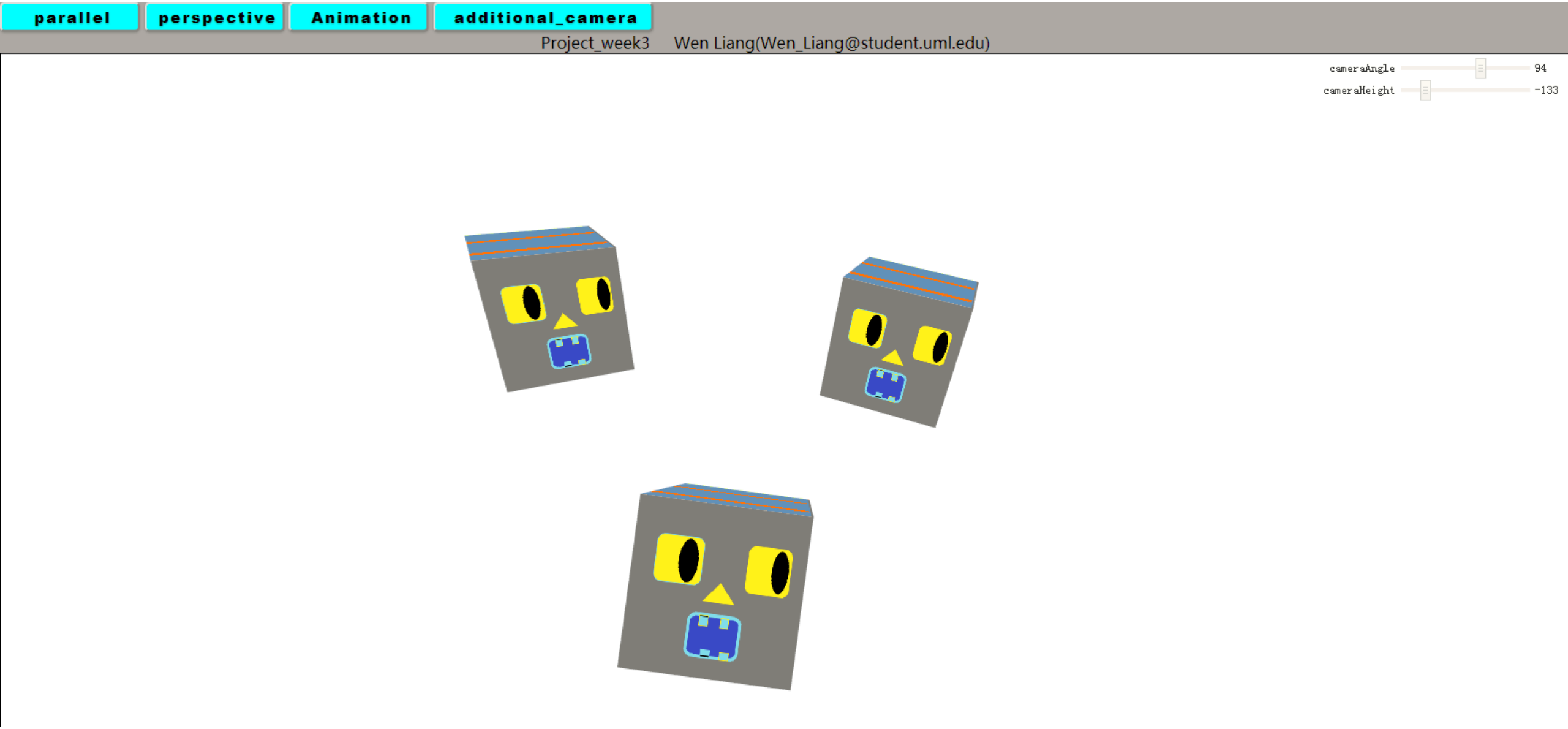
Project_week3 Wen Liang(Wen_Liang@student.uml.edu)

Parallel View: isometric



x	974
y	429
z	0
angleX	346
angleY	31
angleZ	0
scaleX	1.00
scaleY	1.00
scaleZ	1.00
shearX	0.00
shearY	0.00
shininess	150

Week 3


parallel
perspective
Animation
additional_camera
Project_week3 Wen Liang(Wen_Liang@student.uml.edu)
cameraAngle 94
cameraHeight -133

WEEK3

- **8. Add light source to 3D object and allow user to change the shininess of the light.**

Define a light source and then define a view point, then calculate the relationship between the reflection light from the object surface and the view point to calculate the illuminance. Additionally, define a power function to calculate the lightness of light from the light source.

WEEK3

- **Animation of my 3D object.**

Define a rotation speed and rotation angle, then pass them to WebGL build in animation function.

WEEK3

- 10. **Additional camera view point.**

Define a target and compute the viewmatrix when you look at the target at the same time change the camera angle and camera height

Week 4

- **11. different texture option for the user**
- **12. Add red light, green light, blue light lighting effect to (Arbitrary combination of light)**

Week 4

Project_week4 Wen Liang(Wen_Liang@student.uml.edu)

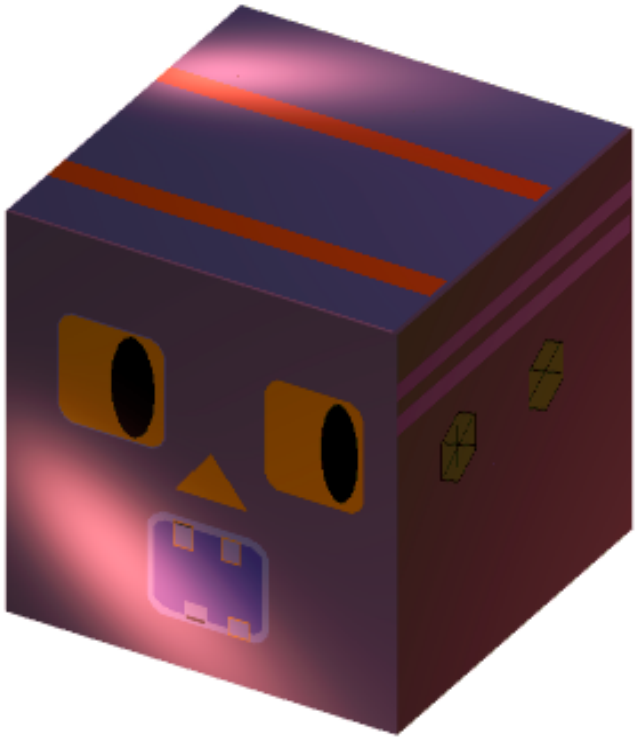
Perspective View: 1 point

Texture: House2

x	0
y	115
z	400
angleX	0
angleY	0
angleZ	0
scaleX	1.00
scaleY	1.00
scaleZ	1.00
shearX	0.00
shearY	0.00
fieldOfView	60
cameraAngle	0
shininess	100
redlight	0.59
greenlight	0.44
bluelight	0.26

Week 4

Project_week4 Wen Liang(Wen_Liang@student.uml.edu)



Parallel View: isometric

Texture: House1

x	798
y	429
z	0
angleX	346
angleY	31
angleZ	0
scaleX	1.00
scaleY	1.00
scaleZ	1.00
shearX	0.00
shearY	0.00
shininess	50
redlight	0.69
greenlight	0.42
bluelight	0.65

Week 4


parallel
perspective
Animation
additional_camera
Project_week4 Wen Liang(Wen_Liang@student.uml.edu)
Texture: House2
cameraAngle 94
cameraHeight -160

WEEK4

- 11. **Add another texture to my 3D object**

Create another texture image first, then put part of the texture image on each side of the 3D Object through WebGL build-in texture mapping API

WEEK4

- 12. **Add red light, green light, blue light lighting effect to my 3D Object, so that user could construct arbitrary combination of these three light themselves**

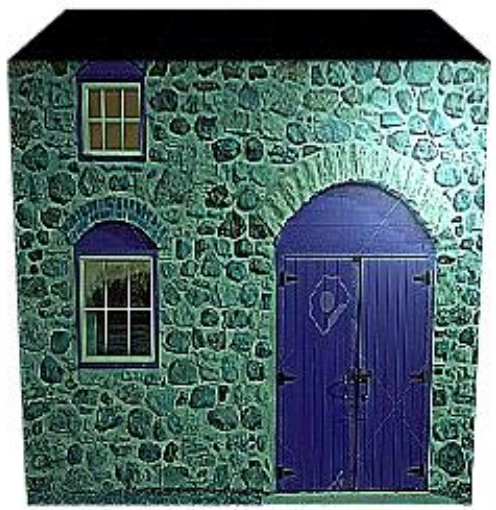
Define the color of the light using the WebGL build-in function, then associate the value of the slide to the value of the (R,G,B)color channel of the light.

Week 5

- **13. Use the image processing technique and different filter kernel to generate new image effect. . (like sharpen, blur, edge detection etc, the “emboss” mode can even has bump-mapping-like effect)**

Week 5

Project_week5 Wen Liang(Wen_Liang@student.uml.edu)



Perspective View: 1 point

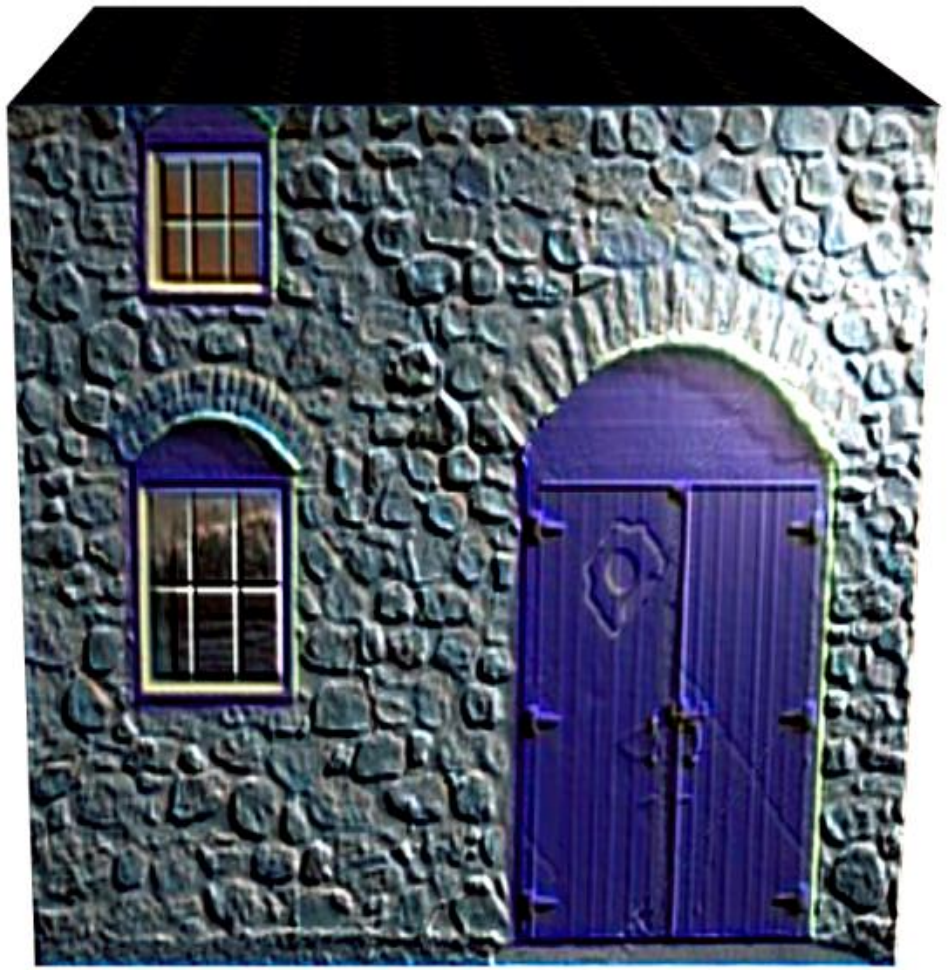
Texture: House2

Filter: unsharp

x	0
y	115
z	400
angleX	0
angleY	0
angleZ	0
scaleX	1.00
scaleY	1.00
scaleZ	1.00
shearX	0.00
shearY	0.00
fieldOfView	60
cameraAngle	0
shininess	100
redlight	0.80
greenlight	1.00
bluelight	0.80

Week 5

Project_week5 Wen Liang(Wen_Liang@student.uml.edu)



Perspective View: 1 point

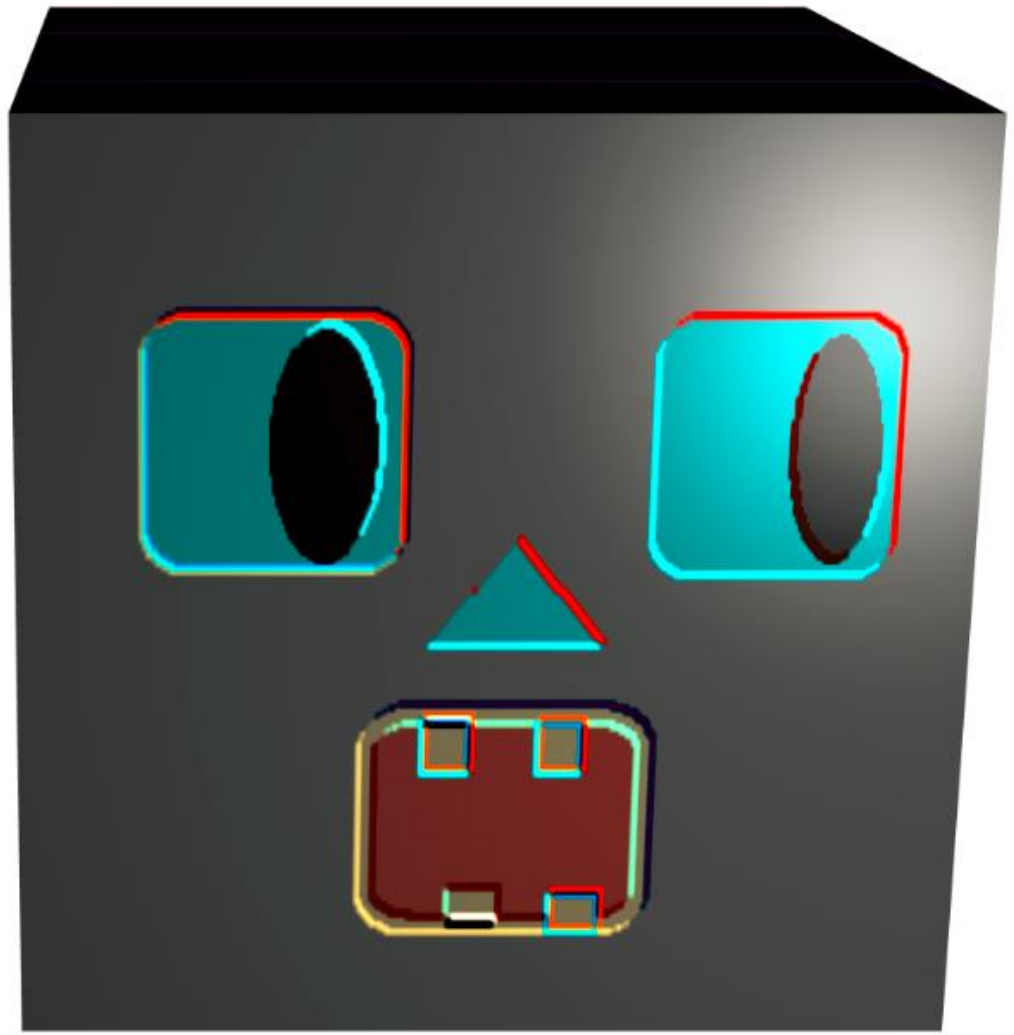
Texture: House2

Filter: emboss

x	23
y	115
z	400
angleX	0
angleY	0
angleZ	0
scaleX	1.00
scaleY	1.00
scaleZ	1.00
shearX	0.00
shearY	0.00
fieldOfView	28
cameraAngle	0
shininess	100
redlight	0.80
greenlight	0.80
bluelight	0.80

Week 5

Project_week5 Wen Liang(Wen_Liang@student.uml.edu)



Perspective View: 1 point

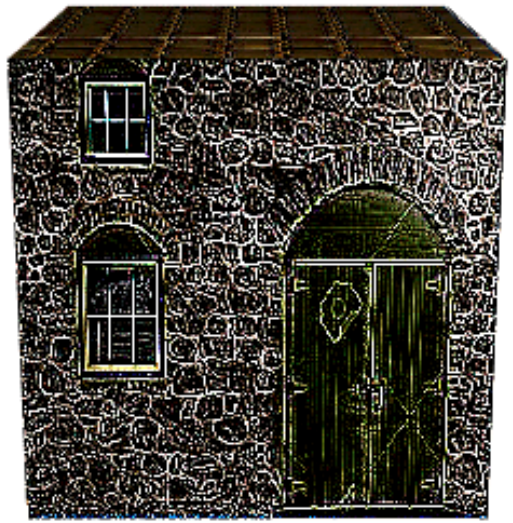
Texture: House1

Filter: emboss

x	66
y	115
z	400
angleX	0
angleY	0
angleZ	0
scaleX	1.00
scaleY	1.00
scaleZ	1.00
shearX	0.00
shearY	0.00
fieldOfView	26
cameraAngle	0
shininess	100
redlight	0.80
greenlight	0.80
bluelight	0.80

Week 5

Project_week5 Wen Liang(Wen_Liang@student.uml.edu)



Perspective View: 1 point

Texture: House2

Filter: edgeDet

x	0
y	115
z	400
angleX	0
angleY	0
angleZ	0
scaleX	1.00
scaleY	1.00
scaleZ	1.00
shearX	0.00
shearY	0.00
fieldOfView	60
cameraAngle	0
shininess	100
redlight	0.80
greenlight	0.80
bluelight	0.80

Week 5

Project_week5 Wen Liang(Wen_Liang@student.uml.edu)

Parallel View: isometric

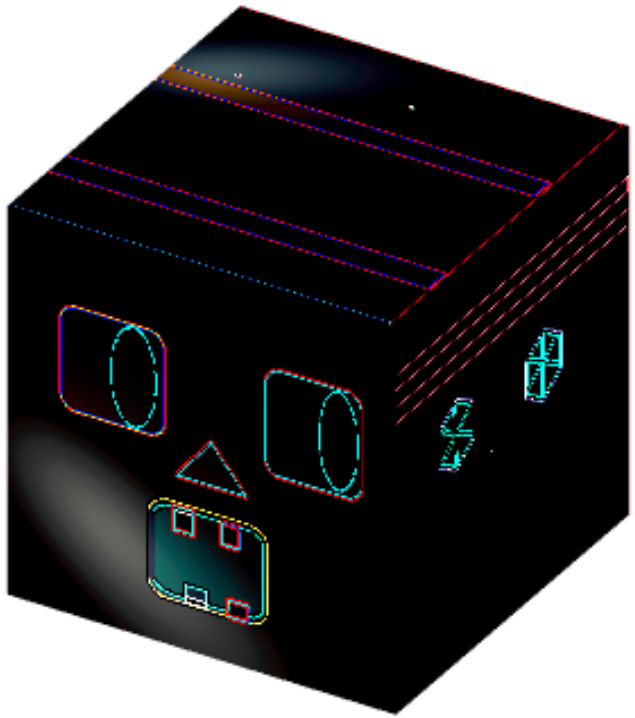
Texture: House1

Filter: normal

x	960
y	429
z	0
angleX	346
angleY	31
angleZ	0
scaleX	1.00
scaleY	1.00
scaleZ	1.00
shearX	0.00
shearY	0.00
shininess	50
redlight	0.80
greenlight	0.80
bluelight	0.80

Week 5

Project_week5 Wen Liang(Wen_Liang@student.uml.edu)



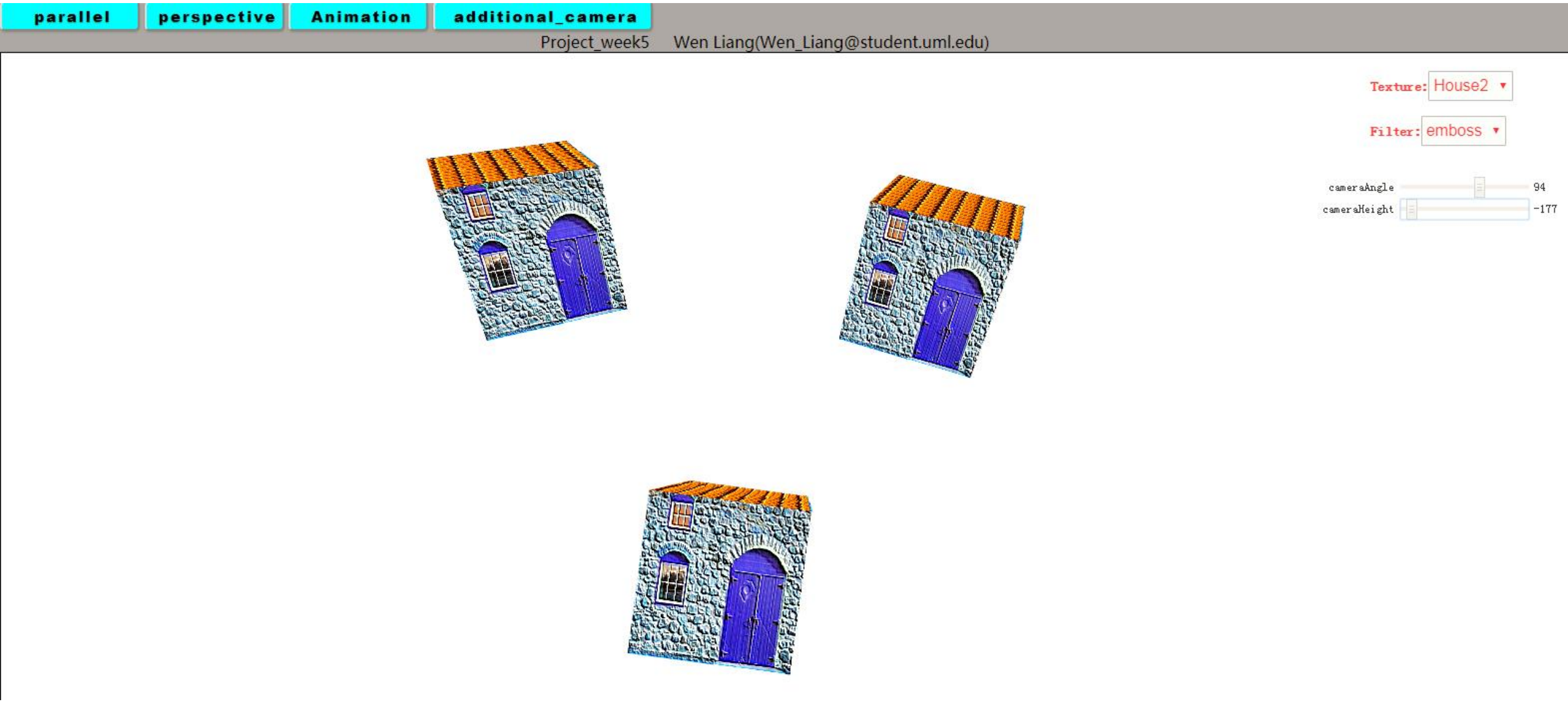
Parallel View: isometric

Texture: House1

Filter: edgeDet

x	865
y	429
z	0
angleX	346
angleY	31
angleZ	0
scaleX	1.00
scaleY	1.00
scaleZ	1.00
shearX	0.00
shearY	0.00
shininess	50
redlight	0.80
greenlight	0.80
bluelight	0.80

Week 5


parallel
perspective
Animation
additional_camera
Project_week5 Wen Liang(Wen_Liang@student.uml.edu)
Texture: House2
Filter: emboss
cameraAngle 94
cameraHeight -177

WEEK5

- 13. **Use the image processing technique and different filter kernel to generate new image effect. like sharpen, blur, edge detection etc, the “emboss” mode can even has bump-mapping-like effect)**

Use the WebGL build-in API to define different filter kernel matrix to process the original image and calculate the weighted sum of the multiplication result between each pixel in the original image and the filter to generate new image effect.

Week 6

- I write up this final project report and debug.

Summary:

- 1. Modeling: draw three 2D “elevation”, and apply 2D transformation
- 2. Transform object: apply 3D transformations to the created object.
- 3. Viewing: view your created object from multiple views.
- 4. Transform camera/viewer
- 5. Generate different projections of the objects
- 6. Edit/Change perspective projection vanishing points (1, 2, 3)
- 7. Create texture for the object
- 8. Add light source to 3D object and allow user to change the shininess
- 9. Animation of my 3D object.
- 10. Additional camera view point.
- 11. different texture option for the user
- 12. Add red light, green light, blue light lighting effect to my 3D Object
- 13. Use the image processing technique and different filter kernel to generate new image effect. (e.g. the “emboss” mode of “Filter” can even has bump-mapping-like effect)

Reference

- WebGL Tutorial: https://webglfundamentals.org/

Thank you!