CIS 515: COMPUTER GRAPHICS
LAB – 7
UNIVERSITY OF MICHIGAN – DEARBORN
FALL 2024

By,
VISHVENDRA REDDY BHOOMIDI
bhoomidi@umich.edu

Task 1 Texture Mapping to a Cube

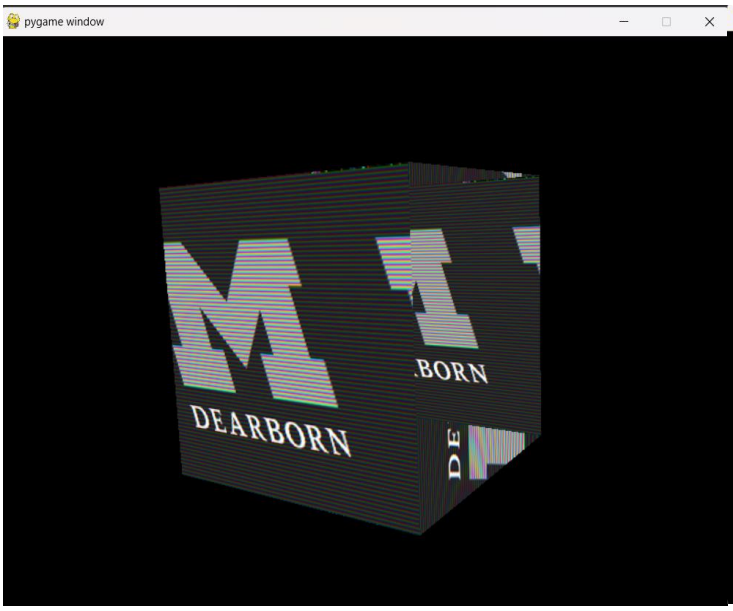
1) Complete mapping of the texture onto each face of the cube

```
import pygame
from OpenGL.GL import *
from OpenGL.GLU import *

# Cube vertices and surfaces
vertices = [
    [-1, -1, -1],
    [1, -1, -1],
    [1, 1, -1],
    [-1, 1, -1],
    [-1, -1, 1],
    [1, -1, 1],
    [1, 1, 1],
    [-1, 1, 1]
]
surfaces = [
    [0, 1, 2, 3],
    [4, 5, 6, 7],
    [0, 4, 7, 3],
    [1, 5, 6, 2],
    [3, 2, 6, 7],
    [0, 1, 5, 4]
]



texture_coords = [
    [0, 0],
    [0.1, 0],
    [0.1, 0.1],
```

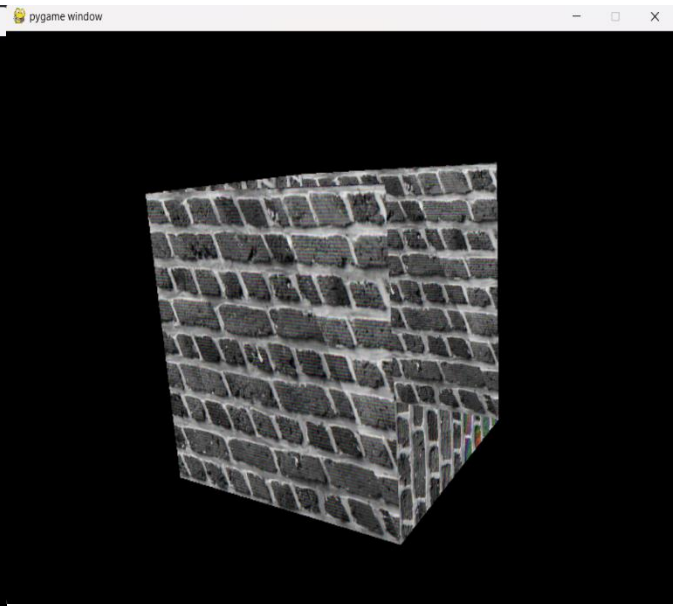
```
    [0, 0.1]
]

def load_texture(image_path):
    texture_surface = pygame.image.load(image_path)
    texture_data = pygame.image.tostring(texture_surface, "RGB", True)
    width, height = texture_surface.get_size()
    texture = glGenTextures(1)
    glBindTexture(GL_TEXTURE_2D, texture)
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, width, height, 0, GL_RGB, GL_UNSIGNED_BYTE,
texture_data)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR)
    return texture



def draw_cube(texture):
    glBindTexture(GL_TEXTURE_2D, texture)
    glBegin(GL_QUADS)
    for surface in surfaces:
        for i, vertex in enumerate(surface):
            glTexCoord2fv(texture_coords[i])
            glVertex3fv(vertices[vertex])
    glEnd()



def main():
    pygame.init()
    display = (800, 600)
    pygame.display.set_mode(display, pygame.DOUBLEBUF | pygame.OPENGL)
    gluPerspective(45, display[0] / display[1], 0.1, 50.0)
    glTranslatef(0.0, 0.0, -5)
```

```
    texture = load_texture("./texture1.png")

    glEnable(GL_TEXTURE_2D)

    clock = pygame.time.Clock()
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()
        glRotatef(1, 3, 1, 1)
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
        draw_cube(texture)
        pygame.display.flip()
        clock.tick(60)
if __name__ == "__main__":
    main()
```

Texture1.png



Texture2.png



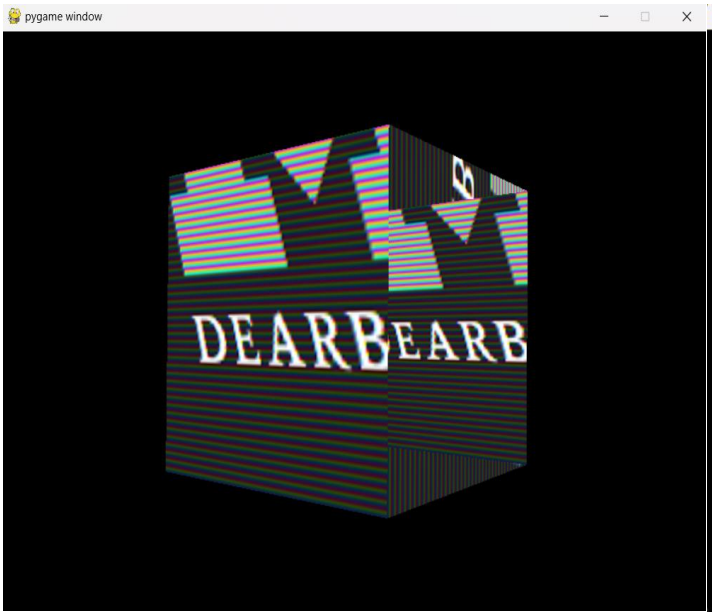
2) Map four copies of the texture onto each face of the cube

```
import pygame
from OpenGL.GL import *
from OpenGL.GLU import *

# Cube vertices and surfaces
vertices = [
    [-1, -1, -1],
    [1, -1, -1],
    [1, 1, -1],
    [-1, 1, -1],
    [-1, -1, 1],
    [1, -1, 1],
    [1, 1, 1],
    [-1, 1, 1]
]
surfaces = [
    [0, 1, 2, 3],
    [4, 5, 6, 7],
    [0, 4, 7, 3],
    [1, 5, 6, 2],
    [3, 2, 6, 7],
    [0, 1, 5, 4]
]



texture_coords = [
    [0, 0],
    [0.5, 0],
    [0.5, 0.5],
    [0, 0.5]
```

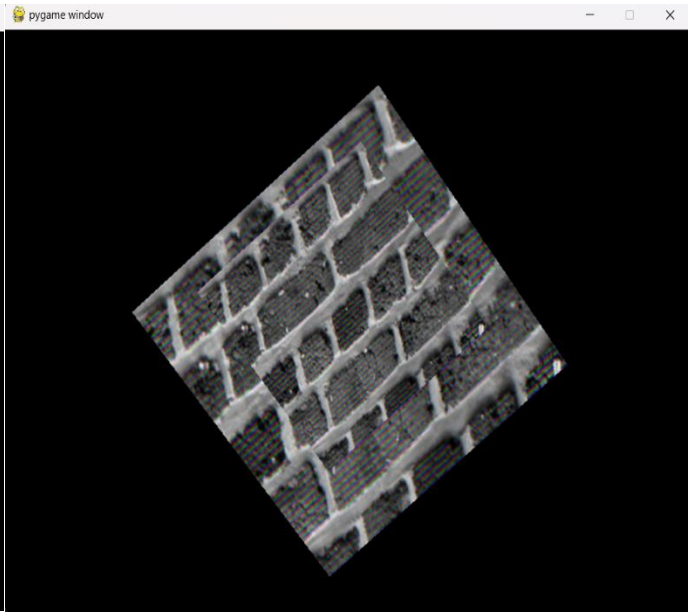
```
]

def load_texture(image_path):
    texture_surface = pygame.image.load(image_path)
    texture_data = pygame.image.tostring(texture_surface, "RGB", True)
    width, height = texture_surface.get_size()
    texture = glGenTextures(1)
    glBindTexture(GL_TEXTURE_2D, texture)
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, width, height, 0, GL_RGB, GL_UNSIGNED_BYTE,
texture_data)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR)
    return texture



def draw_cube(texture):
    glBindTexture(GL_TEXTURE_2D, texture)
    glBegin(GL_QUADS)
    for surface in surfaces:
        for i, vertex in enumerate(surface):
            glTexCoord2fv(texture_coords[i])
            glVertex3fv(vertices[vertex])
    glEnd()



def main():
    pygame.init()
    display = (800, 600)
    pygame.display.set_mode(display, pygame.DOUBLEBUF | pygame.OPENGL)
    gluPerspective(45, display[0] / display[1], 0.1, 50.0)
    glTranslatef(0.0, 0.0, -5)
```

```
    texture = load_texture("./texture1.png")

    glEnable(GL_TEXTURE_2D)

    clock = pygame.time.Clock()
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()

        glRotatef(1, 3, 1, 1)
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
        draw_cube(texture)
        pygame.display.flip()
        clock.tick(60)
if __name__ == "__main__":
    main()
```

Texture1.png



Texture2.png



3) Map a quarter of the texture onto each face of the cube

```
import pygame
from OpenGL.GL import *
from OpenGL.GLU import *

# Cube vertices and surfaces
vertices = [
    [-1, -1, -1],
    [1, -1, -1],
    [1, 1, -1],
    [-1, 1, -1],
    [-1, -1, 1],
    [1, -1, 1],
    [1, 1, 1],
    [-1, 1, 1]
]
surfaces = [
    [0, 1, 2, 3],
    [4, 5, 6, 7],
    [0, 4, 7, 3],
    [1, 5, 6, 2],
    [3, 2, 6, 7],
    [0, 1, 5, 4]
]



texture_coords = [
    [0, 0],
    [0.25, 0],
    [0.25, 0.25],
    [0, 0.25]
]
```

```
def load_texture(image_path):
    texture_surface = pygame.image.load(image_path)
    texture_data = pygame.image.tostring(texture_surface, "RGB", True)
    width, height = texture_surface.get_size()
    texture = glGenTextures(1)
    glBindTexture(GL_TEXTURE_2D, texture)
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, width, height, 0, GL_RGB, GL_UNSIGNED_BYTE, texture_data)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR)
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR)
    return texture


def draw_cube(texture):
    glBindTexture(GL_TEXTURE_2D, texture)
    glBegin(GL_QUADS)
    for surface in surfaces:
        for i, vertex in enumerate(surface):
            glTexCoord2fv(texture_coords[i])
            glVertex3fv(vertices[vertex])
    glEnd()


def main():
    pygame.init()
    display = (800, 600)
    pygame.display.set_mode(display, pygame.DOUBLEBUF | pygame.OPENGL)
    gluPerspective(45, display[0] / display[1], 0.1, 50.0)
    glTranslatef(0.0, 0.0, -5)

    texture = load_texture("./texture1.png")
```

```
    glEnable(GL_TEXTURE_2D)
    clock = pygame.time.Clock()
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()
        glRotatef(1, 3, 1, 1)
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
        draw_cube(texture)
        pygame.display.flip()
        clock.tick(60)

if __name__ == "__main__":
    main()
```

Texture1.png

Texture2.png