

Scrum Update II (YANNIX)

☰ Tags	Scrum Update
🕒 Create	@March 7, 2022 7:02 PM
🔍 Nickname	
🕒 Last edited time	@March 14, 2022 10:48 AM
🔍 Site	
➦ Student	
🔗 URL	
☰ Scrum Update	[Last week] - ทำการศึกษาและทำความเข้าใจ การเขียนโค้ดและ การใช้งานของ OpenGL - ออกแบบ UI ของตัวโปรเจกแบบคร่าวๆ เพื่อเป็นแนวทางในการทำงานต่อไป [This week] - ศึกษาและทดลองใช้ Program ที่สามารถเขียนรูปทรงบนรูปภาพเช่น Silhouette ว่าควรมี Function อะไรบ้างเพื่อที่จะนำไปปรับแก้ UI ที่ออกแบบไป - ศึกษา Model view projection matrix ของ OpenGL - ทำความเข้าใจ FLTK และลองพิมพ์ Code เพื่อใช้งาน [Blocks] - ในสัปดาห์หน้า จะมีการสอบวิชา STA302 จึงจำเป็นที่จะต้องแบ่งเวลาเพื่อที่จะไปเตรียมตัวสอบวิชานั้น
➦ Week	
➦ Scores	
▼ Status	

Resource:

OpenGL: <https://www.youtube.com/playlist?list=PLn3eTxaOtL2PDnEVNwOgZFm5xYPr4dUoR>

My Note:

จากที่สัปดาห์ที่ผ่านมาผมได้ไปทำงานอยู่ 2 ส่วน ซึ่งก็จะมี เรื่องของการศึกษา OpenGL จาก Resource ที่อยู่ในหัวข้อก่อนหน้านี้และ ได้ทำการออกแบบ UI ร่วมกับคนในกลุ่มเพื่อนำไปเป็น

แนวทางในการเขียนโค้ด ซึ่งผมจะขอเริ่มจาก Note ของ OpenGL ก่อนนะครับ

OpenGL

ในหัวข้อนี้ผมได้ทำการเขียนโค้ดและทำการ Comment code ในแต่ละบรรทัดว่ามีไว้ใช้ทำอะไร ซึ่งโค้ดจะอยู่บน Github: <https://github.com/use555555/Onboard/tree/main/Basic>


ซึ่งก็จะมี folder Example ที่จะเป็นโค้ดในตัวอย่างที่ผมได้ทำการ comment ภายในจะมี Transformation ซึ่งจะเป็นของวิดีโอที่ 5 และ Triangle ที่จะเป็นหัวข้อในการวาดสามเหลี่ยมและการใส่ Texture

นอกจากนั้นใน folder Program1 และ 2 จะเป็นส่วนที่ผมนำ Example มาปรับใช้ให้สามารถวาดรูปหลายเหลี่ยมได้และ สามารถปรับความจางบนรูปได้ ซึ่ง Program1 จะไม่มี texture แต่ Program 2 จะมี texture


ในหัวข้อนี้จะขอสรุปในส่วนหลักๆ ที่ได้จากในวิดีโอ ถ้าต้องการเพิ่มเติมขอให้ไปดูในส่วนที่ผม comment ในโค้ดนะครับ

Onboard/Basic at main · use555555/Onboard

Contribute to use555555/Onboard development by creating an account on GitHub.

 <https://github.com/use555555/Onboard/tree/main/Basic>

use555555/
Onboard




1
Contributor

0
Issues

0
Stars

0
Forks



จากที่ผมได้ไปศึกษามาผมได้ทำการดู 5 วิดีโอ แรกของ playlist ซึ่งก็จะมีหัวข้อดังนี้

ในวิดีโอที่ 1 จะเป็นIntro และ Setup เพื่อที่จะเตรียมตัวในการใช้ OpenGL ในวิดีโอนี้ก็จะมี การเขียนโค้ดเริ่มต้นที่จะเป็นการแสดงผลของที่จะมีการเขียนในวิดีโอต่อไป

ในวิดีโอที่ 2 นี้จะเป็นการทำให้โปรแกรมสามารถวาดรูป สามเหลี่ยมได้บนหน้าจอที่ได้ทำการเตรียมไว้ ในวิดีโอที่ 1 ซึ่งในวิดีโอนี้จะมีการเขียนโค้ดกำหนดค่าที่ไว้สำหรับเป็นข้อมูลของ สามเหลี่ยม ซึ่งจะอยู่ใน class Triangle

```
self.vertices = (  
    -0.5, -0.5, 0.0, 1.0, 0.0, 0.0,  
    0.5, -0.5, 0.0, 0.0, 1.0, 0.0,  
    0.0, 0.5, 0.0, 0.0, 0.0, 1.0  
) # OpenGL use normalize device coordinate
```

จากรูปข้างต้นในแต่ละบรรทัดก็จะเป็นข้อมูลของจุดยอดของสามเหลี่ยมซึ่งจะเป็นข้อมูล Position และ Color จะเป็นในรูปแบบ (x, y, z, r, g, b) ซึ่งจะเป็นข้อมูลรวมทำให้หลังจากนั้นจำเป็นต้องแยกข้อมูลออกจากกันเพื่อนำไปใช้ในการวาดซึ่งก็จะเริ่มจาก

```
self.vao = glGenVertexArrays(1) # Add attribute pointer to Vertex object to tell the meaning of the Vertex data
glBindVertexArray(self.vao)

self.vbo = glGenBuffers(1) # vertex buffer = Basic storage container
glBindBuffer(GL_ARRAY_BUFFER, self.vbo) # use to bind buffer
glBufferData(GL_ARRAY_BUFFER, self.vertices.nbytes, self.vertices, GL_STATIC_DRAW)
# Function that use to ship vertices to GPU
# glBufferData(Where to load data, how many byte, data, how we plan to use)
# Static_draw set data once use many times/ Dynamic_draw set and read data as much as you want
```

จะเป็นการเตรียมพื้นที่และ Buffer เพื่อใช้เป็นที่รองรับข้อมูลและ ใช้ในการส่งข้อมูลไปใน GPU จากที่มีพื้นที่แล้วก็ต้องทำการนำข้อมูลมาใส่และ จำแนกชนิดข้อมูลจาก

```
# Describe how the attribute layout
# glVertexAttribPointer(index, number of data, normalise number, stride from the data
# each vertex have 6 number and each number have 4 bytes, offset in bytes)
# Position
glEnableVertexAttribArray(0)
glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 24, ctypes.c_void_p(0))

# Color
glEnableVertexAttribArray(1)
glVertexAttribPointer(1, 3, GL_FLOAT, GL_FALSE, 24, ctypes.c_void_p(12))
```

หลังจากแบ่งข้อมูลแล้วก็จะนำข้อมูลส่งให้ Shader เพื่อใช้ในการวาดรูป ซึ่ง Shader เองก็มีอยู่ 2 ส่วนก็จะมี

1. Vertex shader ที่ใช้ในการ set ข้อมูลของ position บนหน้าจอและดูเรื่องการทำ Transformation
2. Fragment shader ที่ใช้ในการคำนวณสีของแต่ละ pixel

เมื่อมี Shader ที่ใช้ในการวาดรูปแล้วก็จะนำมาวาดใน mainloop ของโค้ด

ในวิดีโอที่ 3 จะเป็นการอธิบายในส่วนของ stride ใน glVertexAttribPointer ว่าการหาค่านั้นสามารถหาคามาจากไหน ซึ่งจากในโค้ดที่ข้อมูลของ Vertices ที่มีค่า

```
self.vertices = (
    -0.5, -0.5, 0.0, 1.0, 0.0, 0.0,
    0.5, -0.5, 0.0, 0.0, 1.0, 0.0,
    0.0, 0.5, 0.0, 0.0, 0.0, 1.0
) # OpenGL use normalize device coordinate
```

จะเห็นได้ว่าใน 1 บรรทัดหรือ 1 Vertex นั้นมีข้อมูลอยู่ 6 ข้อมูลซึ่งจะเป็นข้อมูล Position และ Color จะเป็นในรูปแบบ (x, y, z, r, g, b) ในการแบ่งเพื่อส่งข้อมูลให้ Shader ก็จะมี 3 ข้อมูลแรกเป็น Position และ 3 ข้อมูลหลังเป็น Color และจากโค้ดก็จะมีการจัดให้ข้อมูลแต่ละข้อมูลอยู่ในรูป float32 แสดงว่าข้อมูล 1 ข้อมูล จะมี 32 bit หรือ 4 byte ซึ่งใน glVertexAttribPointer ที่จะทำให้การแบ่งข้อมูลก็ต้องทำการเลื่อน Pointer ที่จะเป็นจุดเริ่มต้นในการเก็บข้อมูล จากที่แต่ละ Vertex มีอยู่ 6 ข้อมูลเพื่อที่จะทำการเลื่อน Pointer ไปให้ตรงกับตัวแรกของข้อมูลที่จะเก็บในแต่ละครั้งจำเป็นที่จะต้องเลื่อนไปที่ละ Vertex และ stride ที่ต้องใส่ไปใน glVertexAttribPointer จะรับเป็นจำนวน Byte ที่ทำการเลื่อน จากที่ข้อมูลมี 6 ข้อมูล และ ข้อมูลหนึ่งมี 4 byte ทำให้ต้องใส่ไปเป็น 24 byte เพื่อใช้ในการเลื่อน

ในวิธีโอที 4 จะเป็นการอธิบายการนำ Texture มาใส่บน shape ที่ได้วาดไปซึ่งก็จะมีส่วนของตัวแปร s และ t เพิ่มมาใน vertices

```
self.vertices = (
    -0.5, -0.5, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0,
    0.5, -0.5, 0.0, 0.0, 1.0, 0.0, 1.0, 1.0,
    0.0, 0.5, 0.0, 0.0, 0.0, 1.0, 0.5, 0.0
) # OpenGL use normalize device coordinate
```

จะเป็นข้อมูล Position, Color และ Texture positionจะเป็นในรูปแบบ (x, y, z, r, g, b, s, t) ซึ่ง

s จะเป็นตัวแปรบอก position ของ texture ว่าอยู่ซ้ายหรือขวาของ shape

t จะเป็นตัวแปรบอก position ของ texture ว่าอยู่บนหรือล่างของ shape

หลังจากที่มีการเพิ่มของข้อมูลก็ต้องมาปรับในเรื่องของ Stride และ shader เพื่อให้รองรับค่า s, t ได้

นอกจากนั้นก็จำเป็นที่จะต้องโหลด texture ที่ต้องการใช้มาในโปรแกรมซึ่งจะใช้เป็น class Material ซึ่งใน class เองก็จะมีการ set ว่าจะนำค่า s, t มาใช้อย่างไรและ ตั้งค่าเรื่องการ render texture ที่มีความลึกมาส่งผล และหลังจากนั้นก็ทำการโหลดรูปและ set ให้เป็น texture ในส่วนนี้สามารถไปอ่านได้ใน code ของ Github

ในวิดีโอที่ 5 จะเป็นเรื่องของการทำงาน Transformation ของรูปและจะเป็นวิดีโอที่เริ่มการเขียนรูป 3D

ซึ่งการทำงาน Transformation นั้นจะเป็นการนำ Transformation Matrix เข้ามาคูณเพื่อที่จะทำให้รูปบนหน้าจอสามารถเคลื่อนที่ได้ ซึ่งก็หลักๆ ก็จะมีเพิ่มใน โค้ด mainloop จะมีการเพิ่มการคูณ Transformation Matrix ดังภาพ

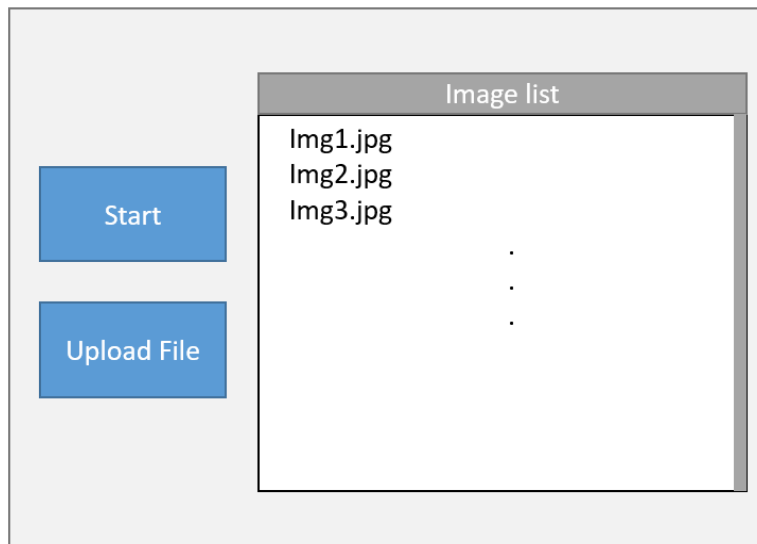
```
# Start with identity
model_transform = pyrr.matrix44.create_identity(dtype=np.float32)
"""
    pitch: rotation around x axis
    roll: rotation around z axis
    yaw: rotation around y axis
"""

# Multiply the transformation
# Rotation
model_transform = pyrr.matrix44.multiply(
    m1=model_transform,
    m2=pyrr.matrix44.create_from_eulers(
        eulers=np.radians(self.cube.eulers), dtype=np.float32
    )
)
# Translation
model_transform = pyrr.matrix44.multiply(
    m1=model_transform,
    m2=pyrr.matrix44.create_from_translation(
        vec=np.array(self.cube.position), dtype=np.float32
    )
)
glUniformMatrix4fv(self.modelMatrixLocation, 1, GL_FALSE, model_transform) # Upload the data
```

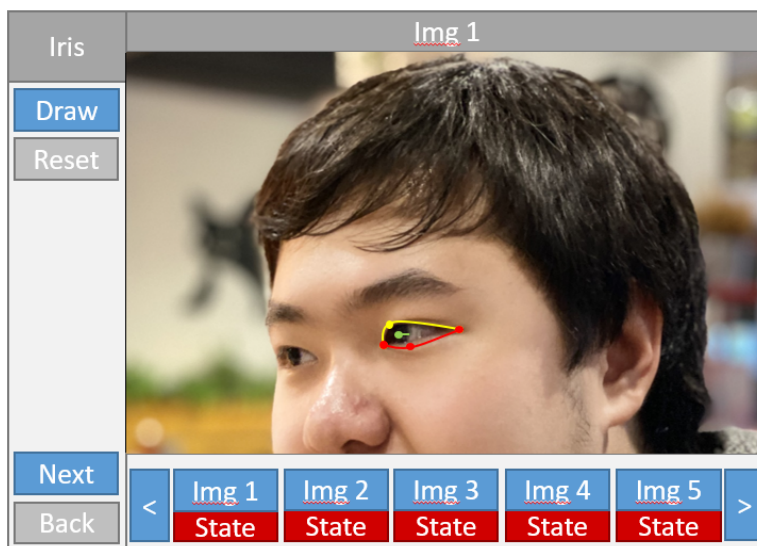
ในโค้ดก็จะเป็นการคูณ Rotation ก่อนที่จะทำ Translation ในการทำ Rotation Roll, Pitch, Yaw ที่เกิดขึ้นบนหน้าจอก็จะเป็นไปตามที่ comment ไว้ดังภาพ เนื่องจากแกนนั้นจะเป็นไปตาม Perspective ที่ได้ Set ไว้ใน code

UI Design

โดย UI นี้คือต้นแบบของตัวโปรแกรมที่จะนำไปใช้เป็น Mini-Project ที่ Site นี้ได้ให้เป็น Assignment



เป็นภาพร่างหน้าแรกของ UI ครบไว้สำหรับโหลดรูปภาพ



เป็นภาพร่างหน้าถัดๆ ไปสำหรับไว้ใช้ในการทำ Segmentation

จากการที่ได้นำตัวอย่าง UI แบบร่างนี้ไปคุยกับ Mentor ได้ประเด็นที่ทำให้ต้องนำกลับไปศึกษา และหา Solution เพื่อมาปรับใช้ UX ให้ดียิ่งขึ้น นอกจากนี้ยังมีเรื่องปรับความเข้าใจกับ mentor เรื่อง requirement อีกด้วย

1. สิ่งที่ mentor ที่ต้องการเบื้องต้นคือ Upload รูปแค่ File เดียวก่อนก็พอ
2. สิ่งที่ต้องการของการวาด segmentation ไม่ใช่แค่การใช้จุด 3 จุดแบบใน UI ที่พวกผมวาด แต่จะต้องวาดได้ละเอียดว่านั่นซึ่ง **ความละเอียดในการวาด**ขอบตาต้องการให้เป็นจุดที่ concern มากที่สุด
3. Output ของ Mini project นี้ คือให้สามารถ Export Segmentation Line ต่างๆ ให้ออกมา ในรูปแบบของ parameter ที่อยู่ในสกุล File (.txt) หรือ (.csv) เพื่อสามารถนำไปใช้ได้

4. ต้องไปทำการศึกษาเรื่อง Type of curves เพื่อให้เข้าใจและอาจจะนำไปใช้ใน Project เพื่อช่วยในการทำ curve ในการ segmentation