

Game Graphic Programming

Homework2

IIIXR Lab. DongHeun Han

1__dong@khu.ac.kr

Goal

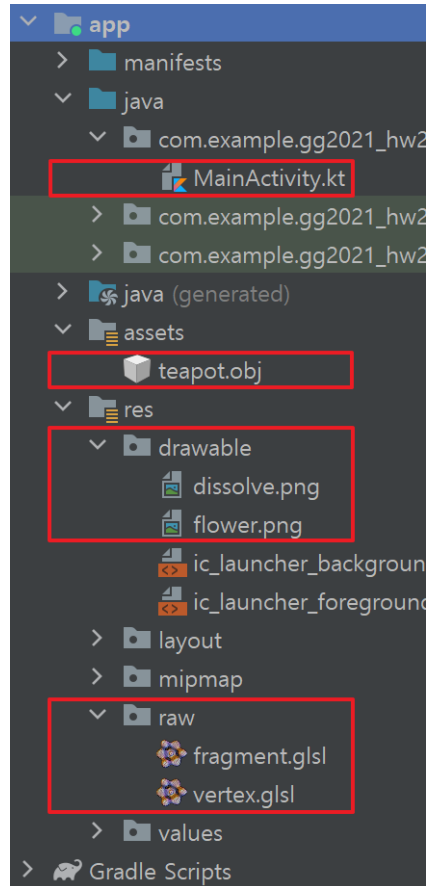
Goal

1. To implement .obj loader, parsing vertex (v), texture (vt), and normal (n).
2. To put a texture on the teapot.
3. To implement Phong lighting with 2 color point lights.
4. To implement alpha blending using an extra dissolve texture.

Problem

Project setting

- Project name: GG2021_HW2_학번 (ex. GG2021_HW2_2017103754)
- Project directory:



Problem

1. Obj loader

- Implement .obj loader, parsing vertex (v), texture (vt), and normal (n).
- Fill the commented part of the code below.
- Our .obj loader stores triangles for glDrawArrays.
- Use the given teapot.obj file and show glDrawArrays works fine.

```
line.startsWith( prefix: "vn ") -> {  
    //-----  
    // Problem 1  
    // Implement .obj loader, parsing vertex (v), texture (vt), and  
    // vn parsing  
  
    // Code  
    //-----  
}  
line.startsWith( prefix: "vt ") -> {  
    //-----  
    // Problem 1  
    // Implement .obj loader, parsing vertex (v), texture (vt), and  
    // vt parsing  
  
    // Code  
    //-----  
}  
line.startsWith( prefix: "f ") -> {  
    //-----  
    // Problem 1  
    // Implement .obj loader, parsing vertex (v), texture (vt), and  
    // v parsing  
  
    // Code  
    //-----  
}
```

```
//-----  
// Problem 1  
// Implement .obj loader, parsing vertex (v), texture (vt), and normal (n)  
  
// % Note  
// vertex: v(xyz) - vn(xyz) - vt(uv) / 3 - 3 - 2 float structure  
// 1 triangle: 3 vertices  
  
// Here, store triangles in the buffer in order based on the obj file.  
// We already implemented the buffer part using "triangleVertices".  
// Implement the rest of the code using "triangleVertices".  
  
// Code  
//-----
```

MainActivity.kt

MainActivity.kt



Problem

2. Texture

- Put a texture on the teapot.
- Fill the commented part of the code below.
- Use the given flower.png file for the teapot texture.
- The Texture's "number" parameter should be started from 0 and incremented by 1 per construction.

```
void main() {
    vec3 normal = normalize(v_normal);
    vec3 view = normalize(v_view);
    vec3 lightL = normalize(v_lightL);
    vec3 lightR = normalize(v_lightR);

    //-----
    // Problem 2
    // Put a texture on the teapot.
    // Change the code below to get the texture value.

    vec3 color = texture2D(textureDiff, v_texCoord).rgb;

    //-----
}
```

fragment.glsl

```
//-----
// Problem 2
// Put a texture on the teapot.

// % Note
// The number parameter of the Texture object must be incremented by 1
// for each texture construction.\
// Also, change the teapot Obj construction code below to get the texture.

// val teapotTexture =
// val dissolveTexture =
// teapotMaterial =
teapot = Obj(mContext, filename: "teapot.obj", mProgram)

//-----
```

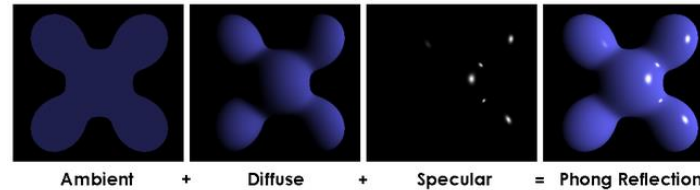
MainActivity.kt



Problem

3. Phong Lighting

- Implement Phong lighting with 2 color point lights.
- Fill the commented part of the code below.
- All preliminary preparations are complete. You only need to change fragment.glsl and vertex.glsl files.
- 2 Lights are distinguished by “L” and “R”.



```
//-----  
// Problem 3  
// Implement the phong shader using 2 color point lights.  
  
// world-space vertex normal  
// v_normal = ;  
  
// view vector  
// v_view = ;  
  
// light vectors  
// v_lightL = ;  
// v_lightR = ;  
  
// attenuations  
// float distL = ;  
// float distR = ;  
// v_attL = ;  
// v_attR = ;  
//-----
```

vertex.glsl

```
//-----  
// Problem 3  
// Implement the phong shader using 2 color point lights.  
  
// diffuse term  
// vec3 matDiff = ;  
// vec3 diffL = ;  
// vec3 diffR = ;  
// vec3 diff = ;  
  
// specular term  
// vec3 reflL = ;  
// vec3 reflR = ;  
// vec3 specL = ;  
// vec3 specR = ;  
// vec3 spec = ;  
  
// ambient term  
// vec3 ambiL = ;  
// vec3 ambiR = ;  
// vec3 ambi = ;  
  
// color = ;  
//-----
```

fragment.glsl



Problem

4. Alpha blending

- Implement alpha blending using an extra dissolve texture.
- Fill the commented part of the code below.
- Use the given dissolve.png file for the dissolve texture.
- Use Material.threshold to control alpha blending.

```
//-----  
// Problem 4  
// Implement the alpha blending using an extra dissolve texture.  
  
// % Note  
// You should first construct a dissolve Texture object and set it into the teapot material first.  
  
// Code  
  
//-----
```

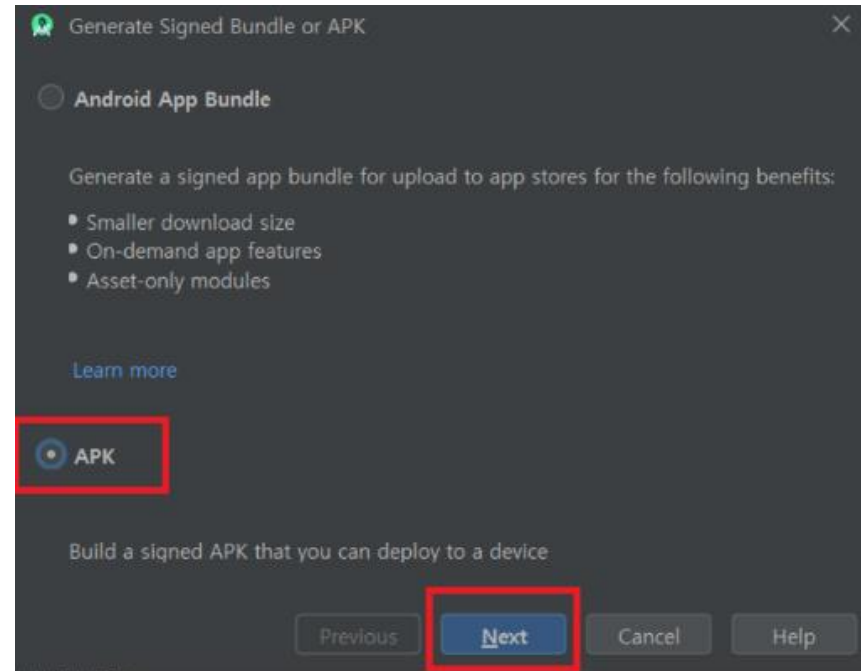
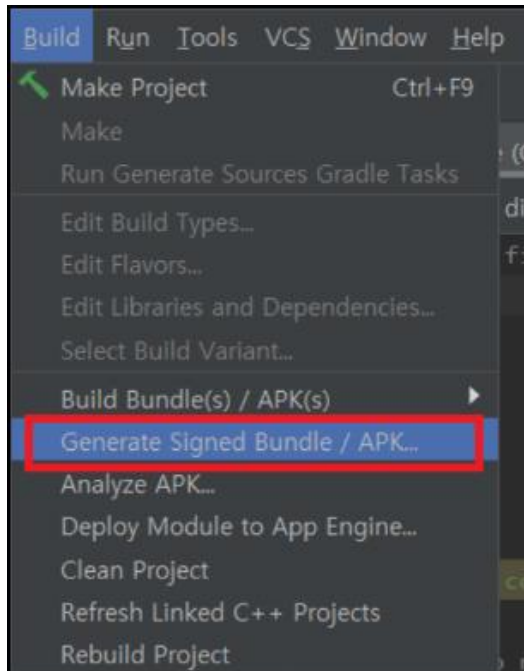
MainActivity.kt

```
float alpha = 1.0;  
//-----  
// Problem 4  
// Implement the alpha blending using an extra dissolve texture.  
  
// Code  
  
//-----
```

fragment.glsl



Generate APK



Generate APK

Generate Signed Bundle or APK

Module: app

Key store path:

Key store password:

Key alias:

Key password:

☐ Remember passwords

New Key Store

Key store path: C:\Users\sharm\HW1Key.jks

Password: Confirm:

Key:

Alias: key0

Password: Confirm:

Validity (years): 25

Certificate

First and Last Name: Siamiz

Organizational Unit:

Organization:

City or Locality:

State or Province:

Country Code (XX):

Submission

Deadline

- 06.03. 23:59

Submit followings to e-campus

- Make an apk file and upload it to your git repository. Then, share your git URL and password via e-campus.
- Submit to e-campus (?week.Homework2): git URL text file, GG2021_HW2_학번.zip (MainActivity.kt, vertex.glsl, fragment.glsl)

TA

- 한동현 (1__dong@khu.ac.kr)

