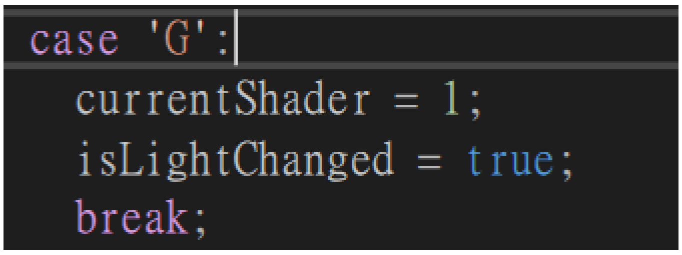
310605009 吳公燿 HW2

1. phong shader-

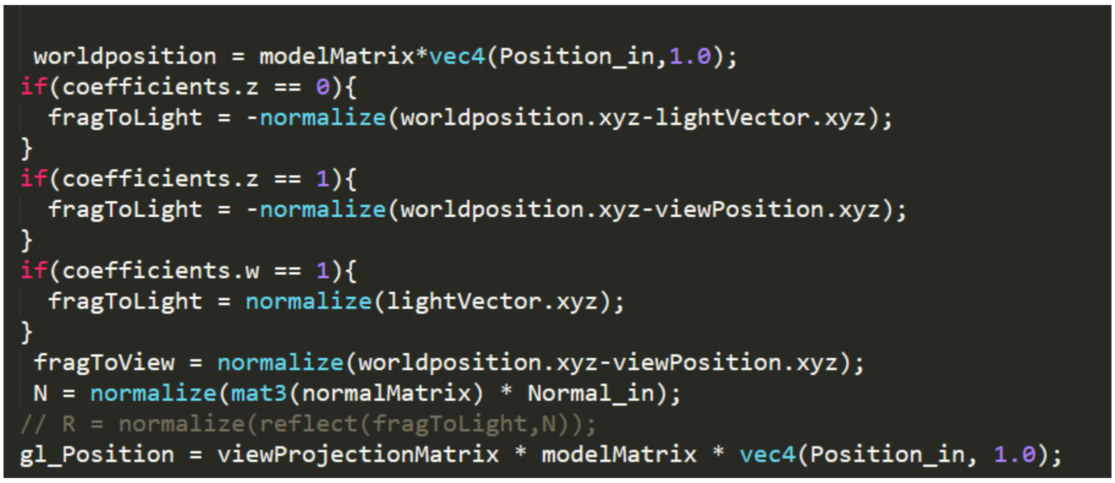
   按 G 來轉換成 phong shader



```
case 'G':
    currentShader = 1;
    isLightChanged = true;
    break;
```

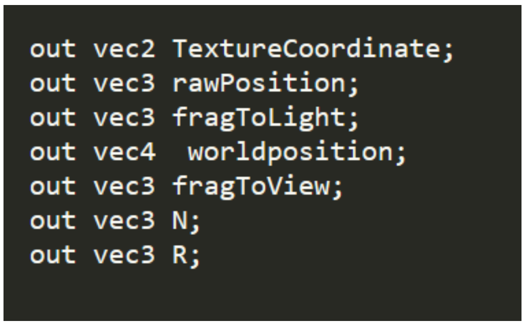
在 Main rendering loop 裡面會透過 shaderPrograms[currentShader].use()轉換成前已經寫好的 shaderPrograms 進入 phong.vert 這邊有遇到的問題就是剛開示燈的方向沒有寫好，以及誤用 rawPosition 來計算世界座標。

phong.vert

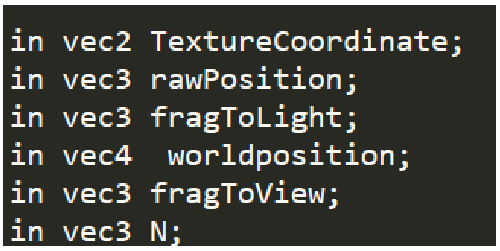


```
 worldposition = modelMatrix*vec4(Position_in,1.0);
if(coefficients.z == 0){
  fragToLight = -normalize(worldposition.xyz-lightVector.xyz);
}
if(coefficients.z == 1){
  fragToLight = -normalize(worldposition.xyz-viewPosition.xyz);
}
if(coefficients.w == 1){
  fragToLight = normalize(lightVector.xyz);
}
 fragToView = normalize(worldposition.xyz-viewPosition.xyz);
 N = normalize(mat3(normalMatrix) * Normal_in);
// R = normalize(reflect(fragToLight,N));
gl_Position = viewProjectionMatrix * modelMatrix * vec4(Position_in, 1.0);
```

之後透過 out 將傳到 phong.frag 的變數傳出



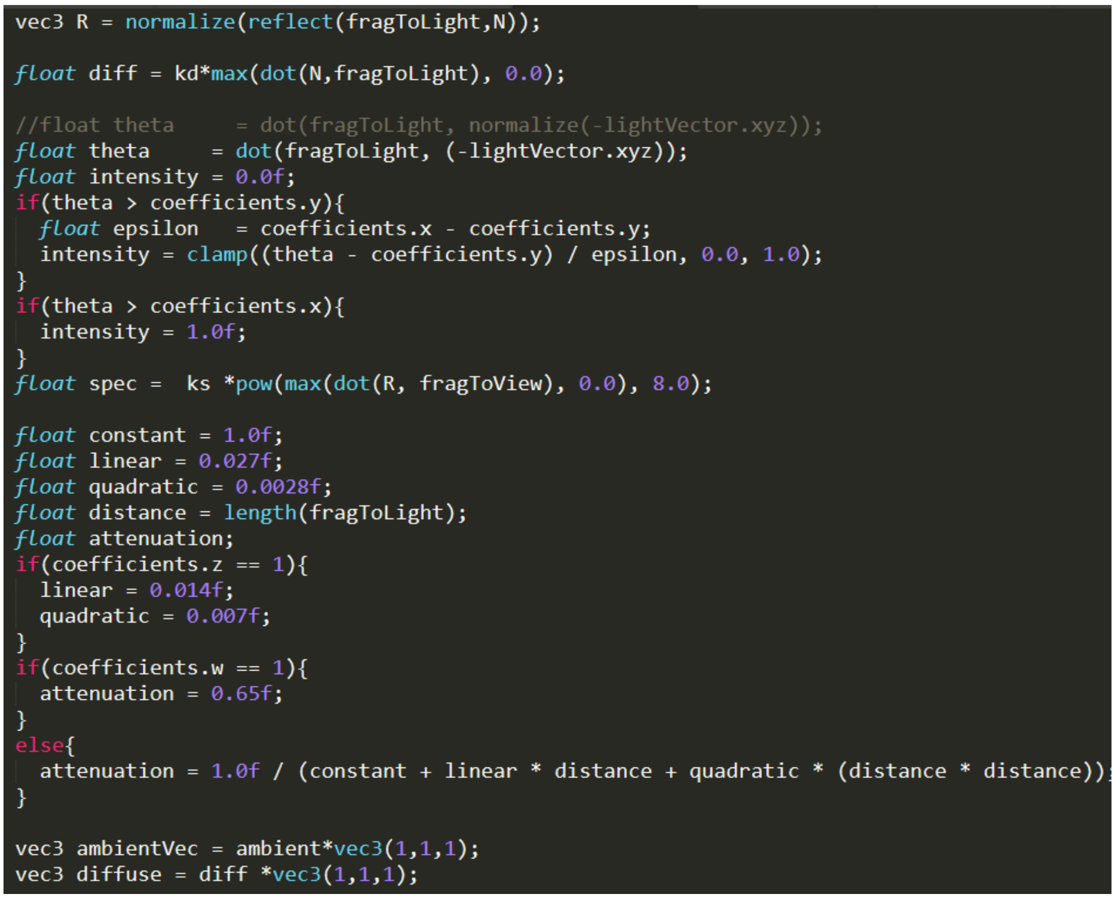
```
out vec2 TextureCoordinate;
out vec3 rawPosition;
out vec3 fragToLight;
out vec4  worldposition;
out vec3 fragToView;
out vec3 N;
out vec3 R;
```



```
in vec2 TextureCoordinate;
in vec3 rawPosition;
in vec3 fragToLight;
in vec4  worldposition;
in vec3 fragToView;
in vec3 N;
```

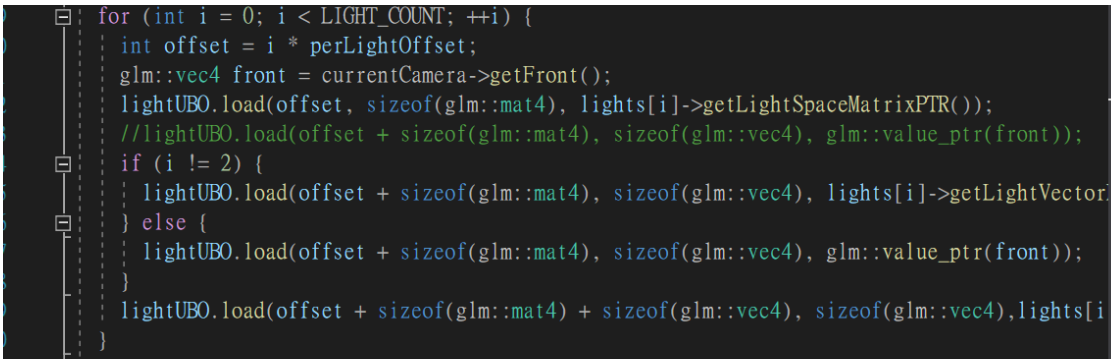
再根據不同的同的燈給予他的 ambient、 attenuation、diffuse 、specular，這邊遇到的問題像是剛開有些 direction 不知道如何找到，因此有參考網站閱讀製作方法以及了解原理。包括 spotlight 剛剛使沒有區分 inner and outer 會有不同 intensity 而使得界線不是很明顯。

phong.frag



```
vec3 R = normalize(reflect(fragToLight,N));

float diff = kd*max(dot(N,fragToLight), 0.0);

//float theta     = dot(fragToLight, normalize(-lightVector.xyz));
float theta     = dot(fragToLight, (-lightVector.xyz));
float intensity = 0.0f;
if(theta > coefficients.y){
  float epsilon   = coefficients.x - coefficients.y;
  intensity = clamp((theta - coefficients.y) / epsilon, 0.0, 1.0);
}
if(theta > coefficients.x){
  intensity = 1.0f;
}
float spec =  ks *pow(max(dot(R, fragToView), 0.0), 8.0);

float constant = 1.0f;
float linear = 0.027f;
float quadratic = 0.0028f;
float distance = length(fragToLight);
float attenuation;
if(coefficients.z == 1){
  linear = 0.014f;
  quadratic = 0.007f;
}
if(coefficients.w == 1){
  attenuation = 0.65f;
}
else{
  attenuation = 1.0f / (constant + linear * distance + quadratic * (distance * distance))
}

vec3 ambientVec = ambient*vec3(1,1,1);
vec3 diffuse = diff *vec3(1,1,1);
```

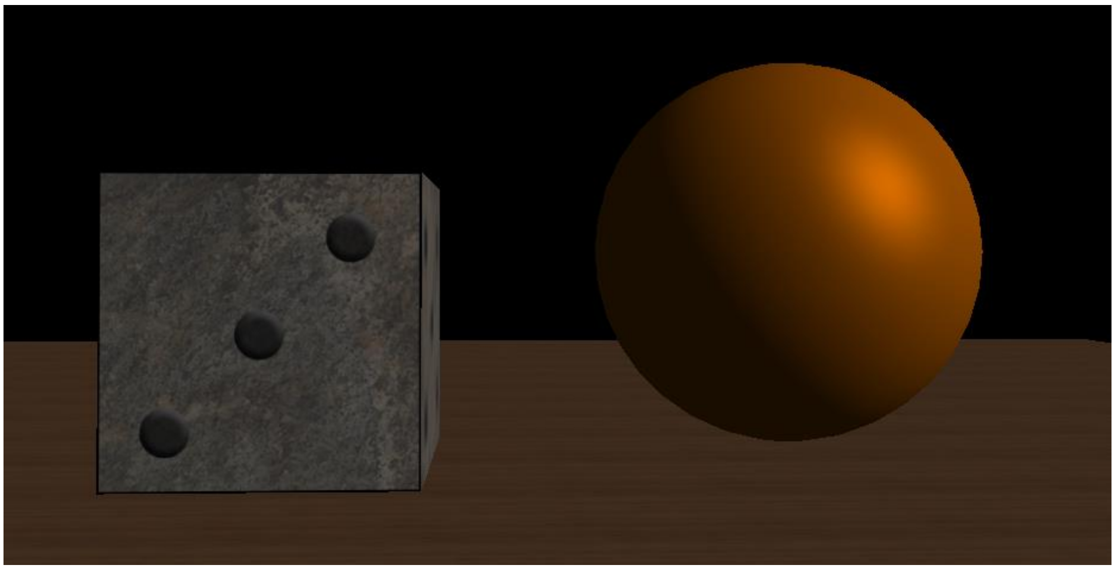
且這邊的燈是由下圖 load 再由 main render 判斷當時燈的種類



```
for (int i = 0; i < LIGHT_COUNT; ++i) {
  int offset = i * perLightOffset;
  glm::vec4 front = currentCamera->getFront();
  lightUBO.load(offset, sizeof(glm::mat4), lights[i]->getLightSpaceMatrixPTR());
  //lightUBO.load(offset + sizeof(glm::mat4), sizeof(glm::vec4), glm::value_ptr(front));
  if (i != 2) {
    lightUBO.load(offset + sizeof(glm::mat4), sizeof(glm::vec4), lights[i]->getLightVector
  } else {
    lightUBO.load(offset + sizeof(glm::mat4), sizeof(glm::vec4), glm::value_ptr(front));
  }
  lightUBO.load(offset + sizeof(glm::mat4) + sizeof(glm::vec4), sizeof(glm::vec4),lights[i
}
```

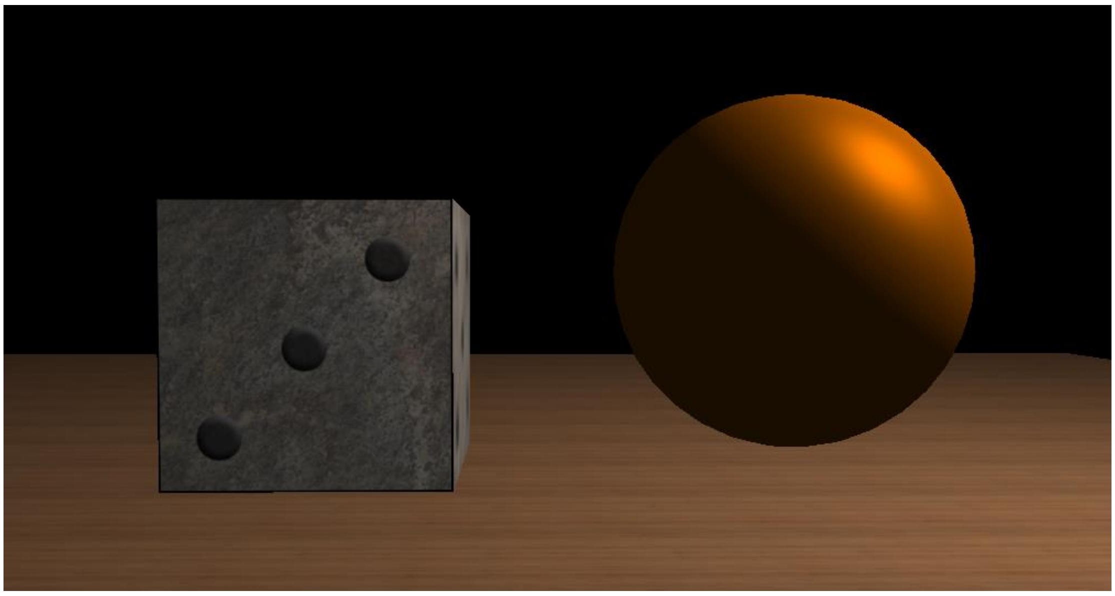
1.1phong shader- directional light

按 R 來轉換成 directional light



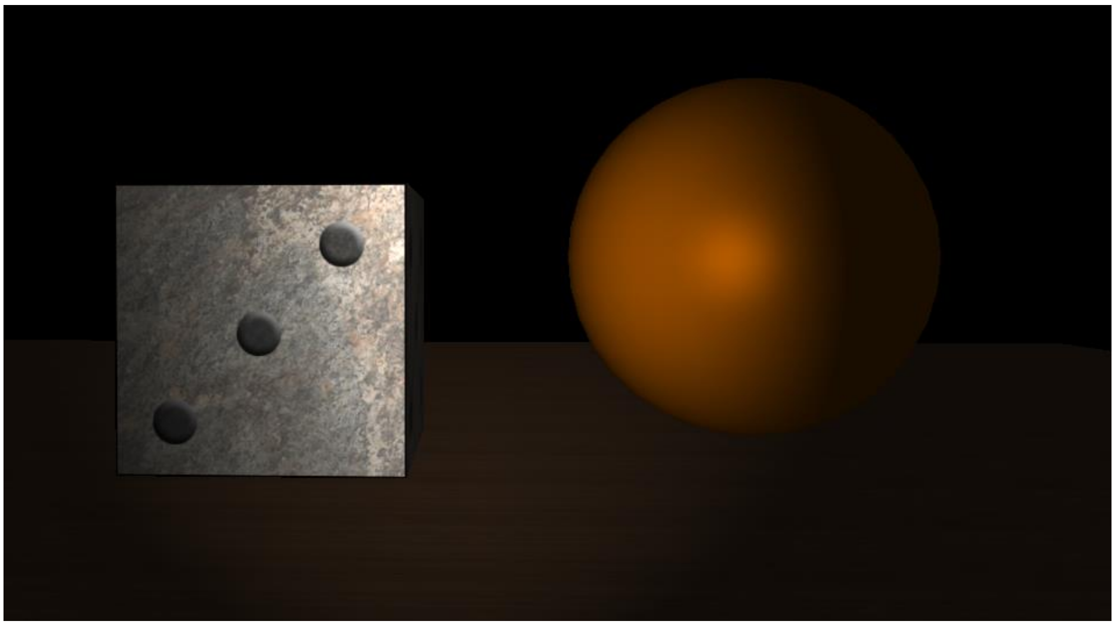
1.2 phong shader- point light

按 T 來轉換成 point light



1.2phong shader- spotlight

按 Y 來轉換成 spotlight

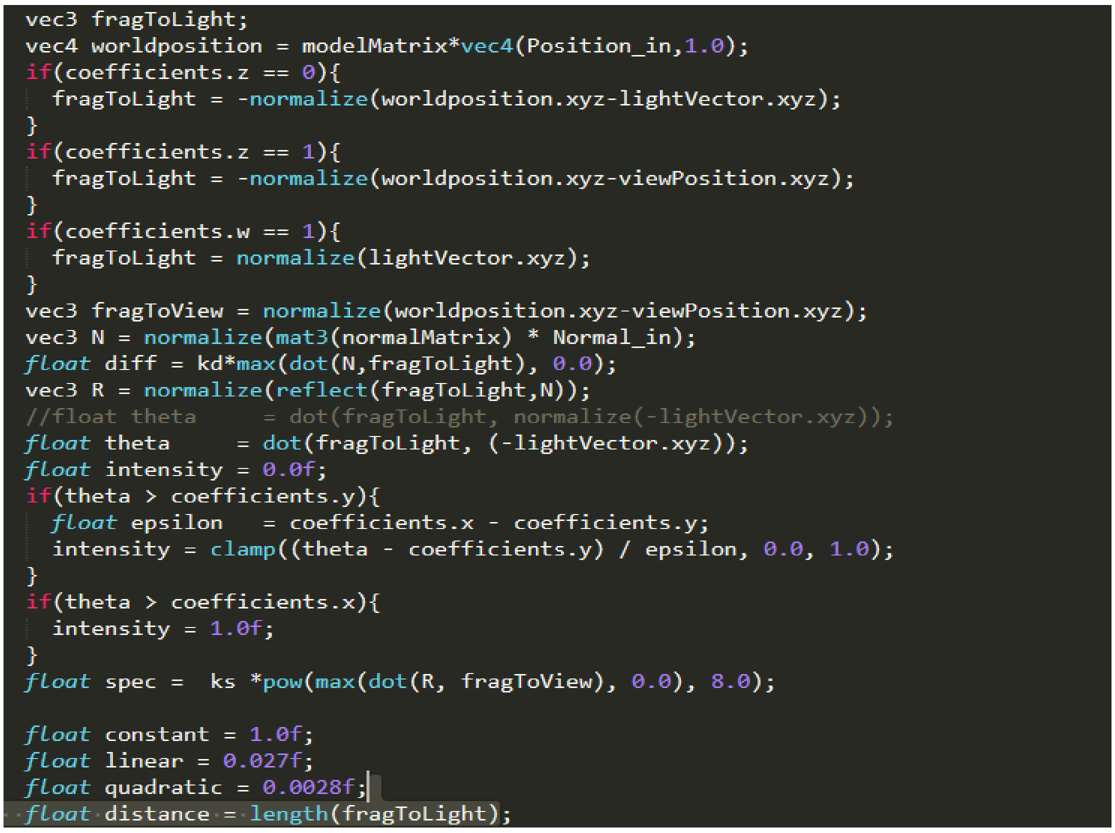


2.1 gouraud shader- directional light

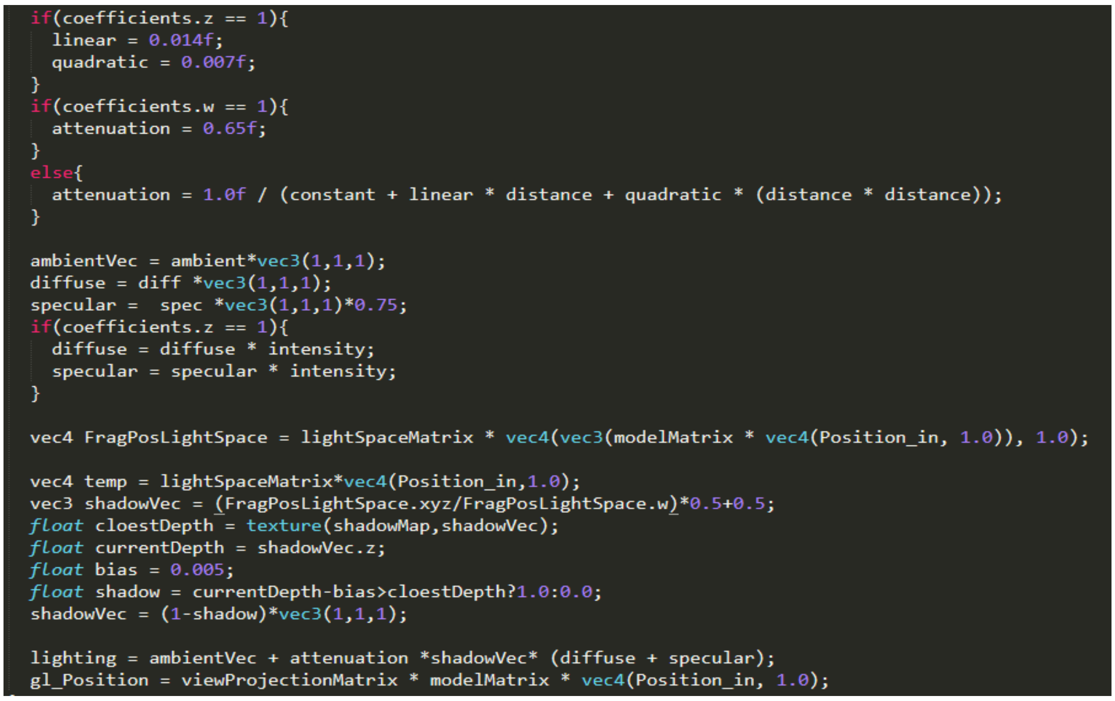
按 F 來轉換成 gouraud shader

基本上和 gouraud shader 方法差不多但一個事先計算完再 out 給 fragment 上色，而 phong shader 則是在 fragment 地方進行計算

gouraud.vert

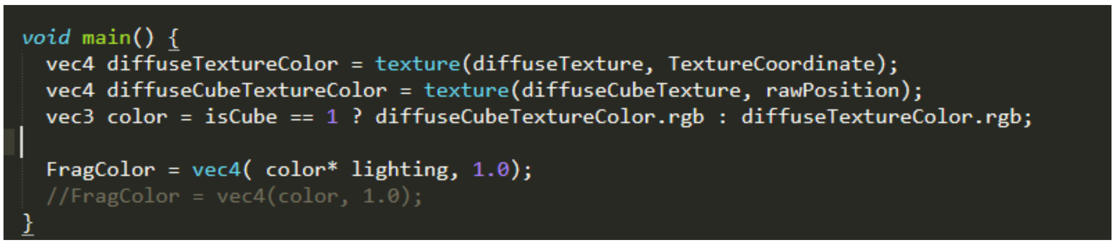


```
vec3 fragToLight;
vec4 worldposition = modelMatrix*vec4(Position_in,1.0);
if(coefficients.z == 0){
  fragToLight = -normalize(worldposition.xyz-lightVector.xyz);
}
if(coefficients.z == 1){
  fragToLight = -normalize(worldposition.xyz-viewPosition.xyz);
}
if(coefficients.w == 1){
  fragToLight = normalize(lightVector.xyz);
}
vec3 fragToView = normalize(worldposition.xyz-viewPosition.xyz);
vec3 N = normalize(mat3(normalMatrix) * Normal_in);
float diff = kd*max(dot(N,fragToLight), 0.0);
vec3 R = normalize(reflect(fragToLight,N));
//float theta     = dot(fragToLight, normalize(-lightVector.xyz));
float theta     = dot(fragToLight, (-lightVector.xyz));
float intensity = 0.0f;
if(theta > coefficients.y){
  float epsilon   = coefficients.x - coefficients.y;
  intensity = clamp((theta - coefficients.y) / epsilon, 0.0, 1.0);
}
if(theta > coefficients.x){
  intensity = 1.0f;
}
float spec =  ks *pow(max(dot(R, fragToView), 0.0), 8.0);

float constant = 1.0f;
float linear = 0.027f;
float quadratic = 0.0028f;
float distance = length(fragToLight);
```



```
if(coefficients.z == 1){
  linear = 0.014f;
  quadratic = 0.007f;
}
if(coefficients.w == 1){
  attenuation = 0.65f;
}
else{
  attenuation = 1.0f / (constant + linear * distance + quadratic * (distance * distance));
}

ambientVec = ambient*vec3(1,1,1);
diffuse = diff *vec3(1,1,1);
specular =  spec *vec3(1,1,1)*0.75;
if(coefficients.z == 1){
  diffuse = diffuse * intensity;
  specular = specular * intensity;
}

vec4 FragPosLightSpace = lightSpaceMatrix * vec4(vec3(modelMatrix * vec4(Position_in, 1.0)), 1.0);

vec4 temp = lightSpaceMatrix*vec4(Position_in,1.0);
vec3 shadowVec = (FragPosLightSpace.xyz/FragPosLightSpace.w)*0.5+0.5;
float cloestDepth = texture(shadowMap,shadowVec);
float currentDepth = shadowVec.z;
float bias = 0.005;
float shadow = currentDepth-bias>cloestDepth?1.0:0.0;
shadowVec = (1-shadow)*vec3(1,1,1);

lighting = ambientVec + attenuation *shadowVec* (diffuse + specular);
gl_Position = viewProjectionMatrix * modelMatrix * vec4(Position_in, 1.0);
```

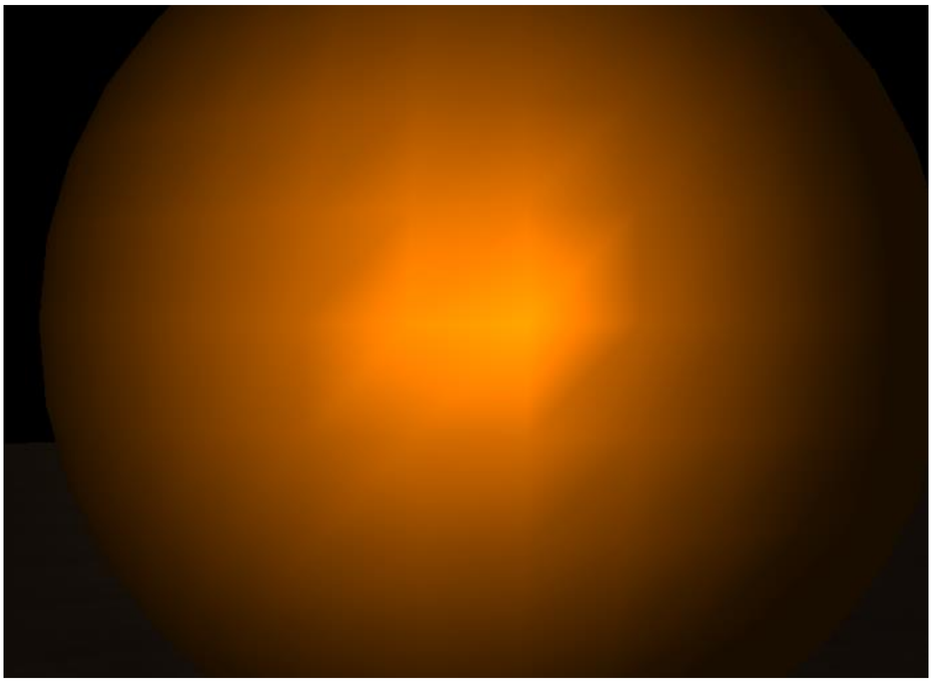
gouraud.frag



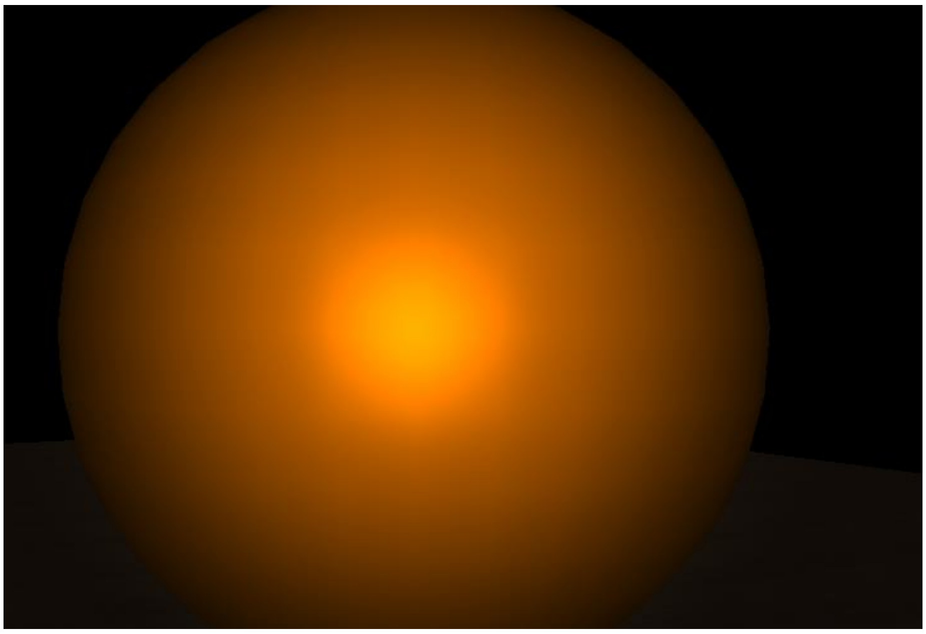
```
void main() {
  vec4 diffuseTextureColor = texture(diffuseTexture, TextureCoordinate);
  vec4 diffuseCubeTextureColor = texture(diffuseCubeTexture, rawPosition);
  vec3 color = isCube == 1 ? diffuseCubeTextureColor.rgb : diffuseTextureColor.rgb;

  FragColor = vec4( color* lighting, 1.0);
  //FragColor = vec4(color, 1.0);
}
```

可以由下圖看到 gouraud shader 有明顯的網格 phong shader 則較平順

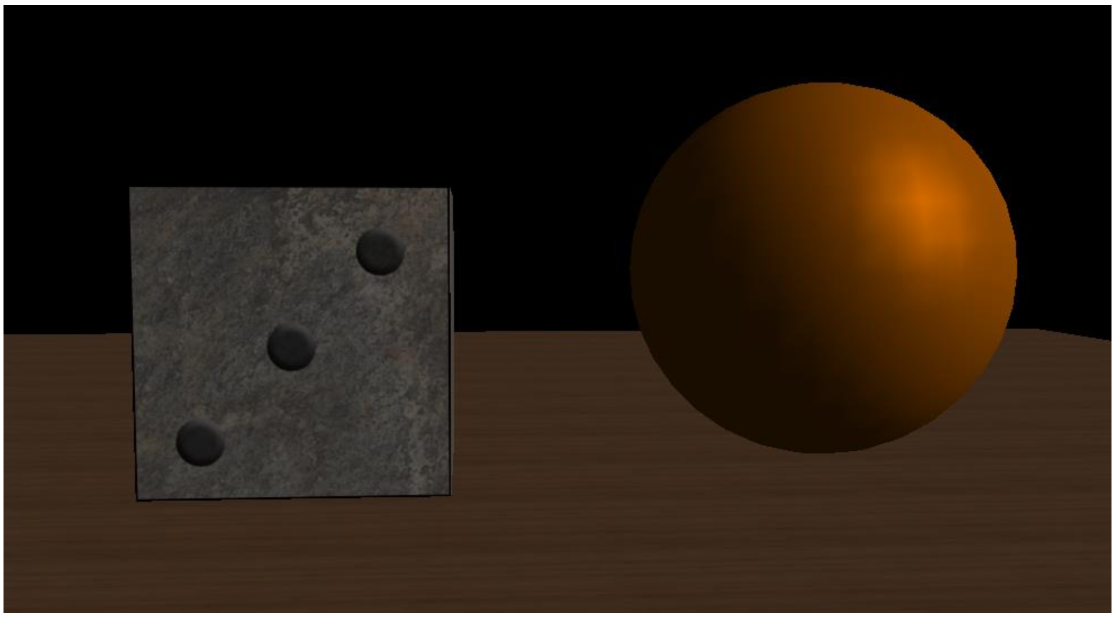
gouraud shader



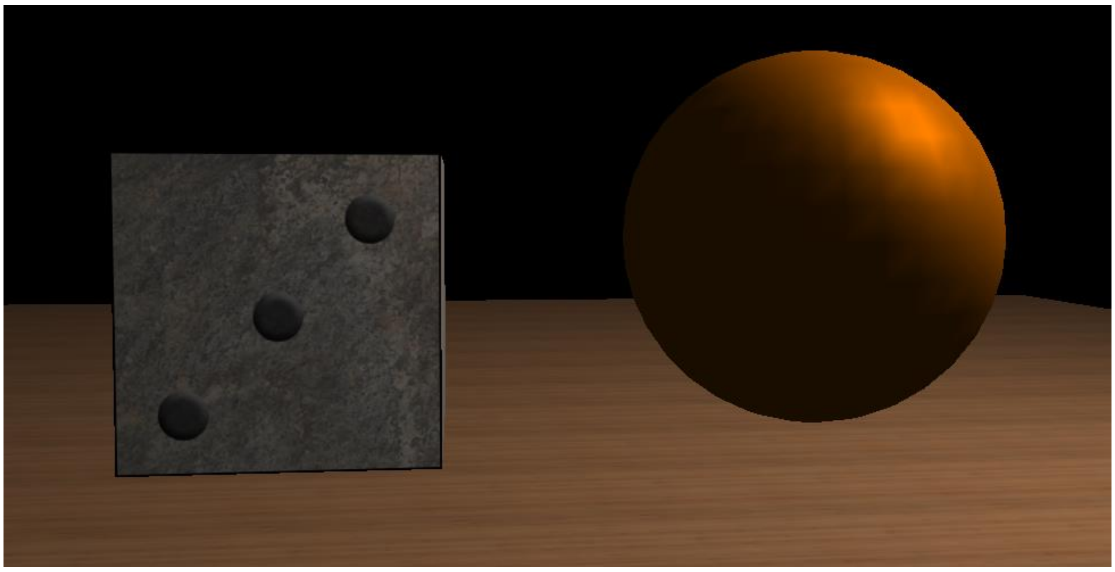
phong shader



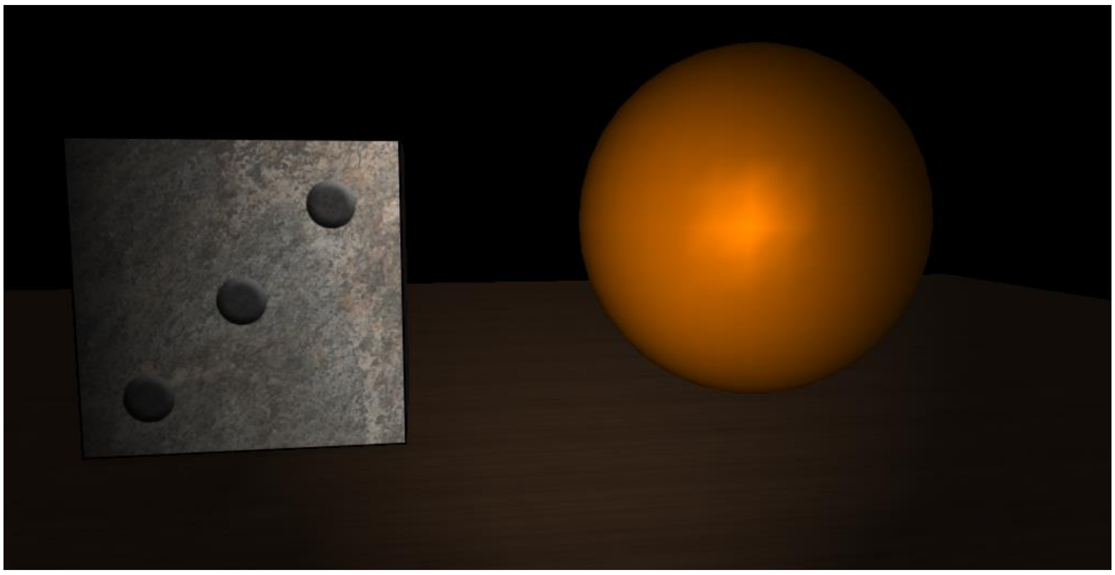
2.1 gouraud shader- directional light



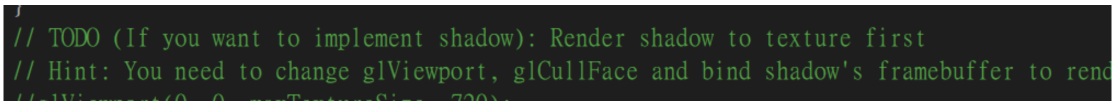
2.2 gouraud shader- point light



2.3 gouraud shader- spot light



最後有事試著做出 shadow 改了 light 的 projectionMatrixu 以及寫.vert 跟.frag 試圖做 shadow 加到 FrafColor 裡面但做出黑色部分但因為不知道如何改下面的東西沒法將 shadow 放在正確的位置



```
// TODO (If you want to implement shadow): Render shadow to texture first
// Hint: You need to change glViewport, glCullFace and bind shadow's framebuffer to rend
```