Tech document

1. Firework

   利用 particle 的方式每 30msec update

```
void TrainView::fireworkUpdate() {
    srand(clock());
    int radius = 30;
    float offset = 2.5f;
    if (fire_pos[0][3][1] > 150) {
        fireworkRadius++;
        if ((int)fireworkRadius % 2 == 0) {
            fire_pos.clear();
            fire_angle.clear();
        }

        for (int j = 0; j < fireworkCircle; j++) {
            glm::mat4 model = glm::mat4(1.0f);
            float angle = (float)j / (float)fireworkCircle * 360.0f;
            float displacement = (rand() % (int)(2 * offset * 100)) / 100.0f - offset;
            float x = cos(angle*3.1415926/180) * fireworkRadius + displacement;
            displacement = (rand() % (int)(2 * offset * 100)) / 100.0f - offset;

            float y = fire_pos[0][3][1] + sin(angle * 3.1415926 / 180) * fireworkRadius;

            displacement = (rand() % (int)(2 * offset * 100)) / 100.0f - offset;
            float z = 0;
            model = glm::translate(model, glm::vec3(x, y, z));

            // 2. scale: scale between 0.05 and 0.25f
            float scale = (rand() % 50) / 300.0f + 0.05;
            model = glm::scale(model, glm::vec3(scale));

            // 3. rotation: add random rotation around a (semi)randomly picked rotation axis vector
            float rotAngle = (rand() % 360);
            model = glm::rotate(model, rotAngle, glm::vec3(0.4f, 0.6f, 0.8f));

            fire_pos.push_back(model);
            fire_angle.push_back(angle);
        }
        if (fireworkRadius > radius) {
            fireworkOver = true;
        }
    }
    else {
        fireworkSpeed *= 1.03;
        fire_pos[0][3][1] += fireworkSpeed;
        fire_angle[0] = rand() % 180;
    }
}
```

2. mipmap

   生成多張不同解析度的貼圖來做 LOD

```
int w= this->size.x, h= this->size.y;
uchar* img_data[10];
for (int i = 0; i < 10; i++) {
    img_data[i] =new uchar[w*h*3];
    gluScaleImage(GL_RGB, this->size.x, this->size.y, GL_UNSIGNED_BYTE, img.data, w, h, GL_UNSIGNED_BYTE, img_data[i]);
    glTexImage2D(GL_TEXTURE_2D, i, GL_RGB8, w, h, 0, GL_BGR, GL_UNSIGNED_BYTE, img_data[i]);
    w /= 2;
    h /= 2;
}
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST_MIPMAP_LINEAR);
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST_MIPMAP_LINEAR);
```

3. post process

   1.馬賽克濾鏡

```
if(pixel_enable){
    vec2 uv = TexCoords.xy;
    float dx = pixel_w*(1.0/rt_w);
    float dy = pixel_h*(1.0/rt_h);
    vec2 coord = vec2(dx*floor(uv.x/dx),dy*floor(uv.y/dy));
    vec3 tc = texture2D(screenTexture, coord).rgb;
    FragColor = vec4(tc, 1.0);
}
```

   2.Cartoon shading

   參考資料: http://coding-experiments.blogspot.com/2011/01/toon-pixel-shader.html

   3.Sketch shading

   參考資料: https://gist.github.com/jcayzac/1192583

4. Billboard Object

   根據物體作標與相機座標來計算物體需要轉的角度，讓物體永遠朝向相機方向。

```
float building_angle = atan2(building_pos[i].x - viewPos.x, building_pos[i].y - viewPos.z) * (180 / 3.1415926535);
glm::mat4 building_transfer = building_modelMatrices[i];
building_transfer = glm::rotate(building_transfer, glm::radians(building_angle + 90), glm::vec3(0.0, 1.0, 0.0));
glUniformMatrix4fv(glGetUniformLocation(loadmodel_shader->Program, "model_matrix"), 1, GL_FALSE, &building_transfer[0][0]);
building_model->Draw(*loadmodel_shader);
```

5. Paramatric Instancing

   參考資料: https://learnopengl.com/Advanced-OpenGL/Instancing