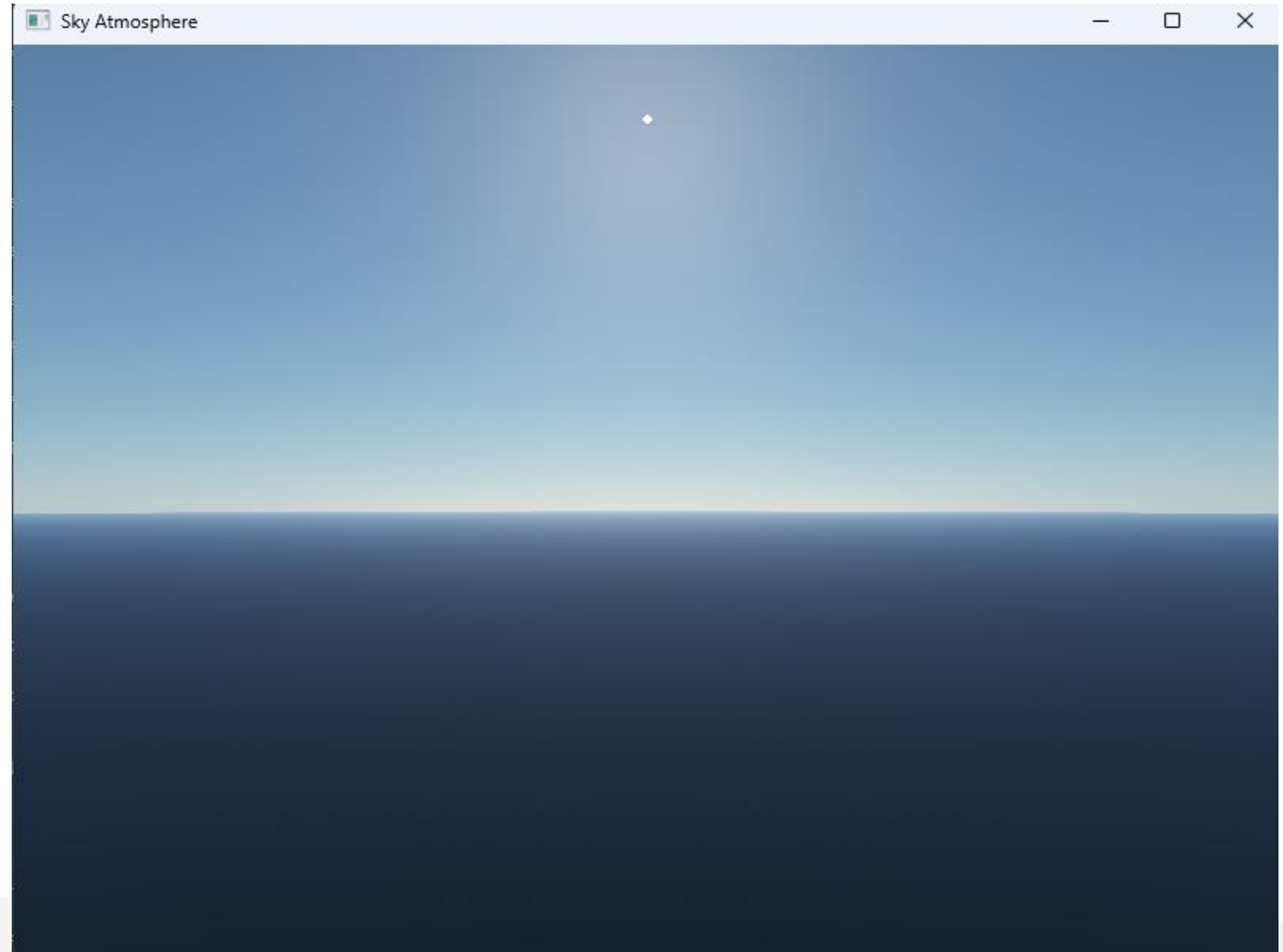


# Sky Atmosphere



- 대기환경 구현(개인 공부)
  - <https://sebh.github.io/publications/egsr2020.pdf>
  - <https://github.com/sebh/UnrealEngineSkyAtmosphere>
  - 위 링크의 DirectX 구현 버전 참고 OpenGL 구현
- 렌더 패스
  - *Transmittance LUT*
  - *SkyView LUT*
  - *PostProcess*

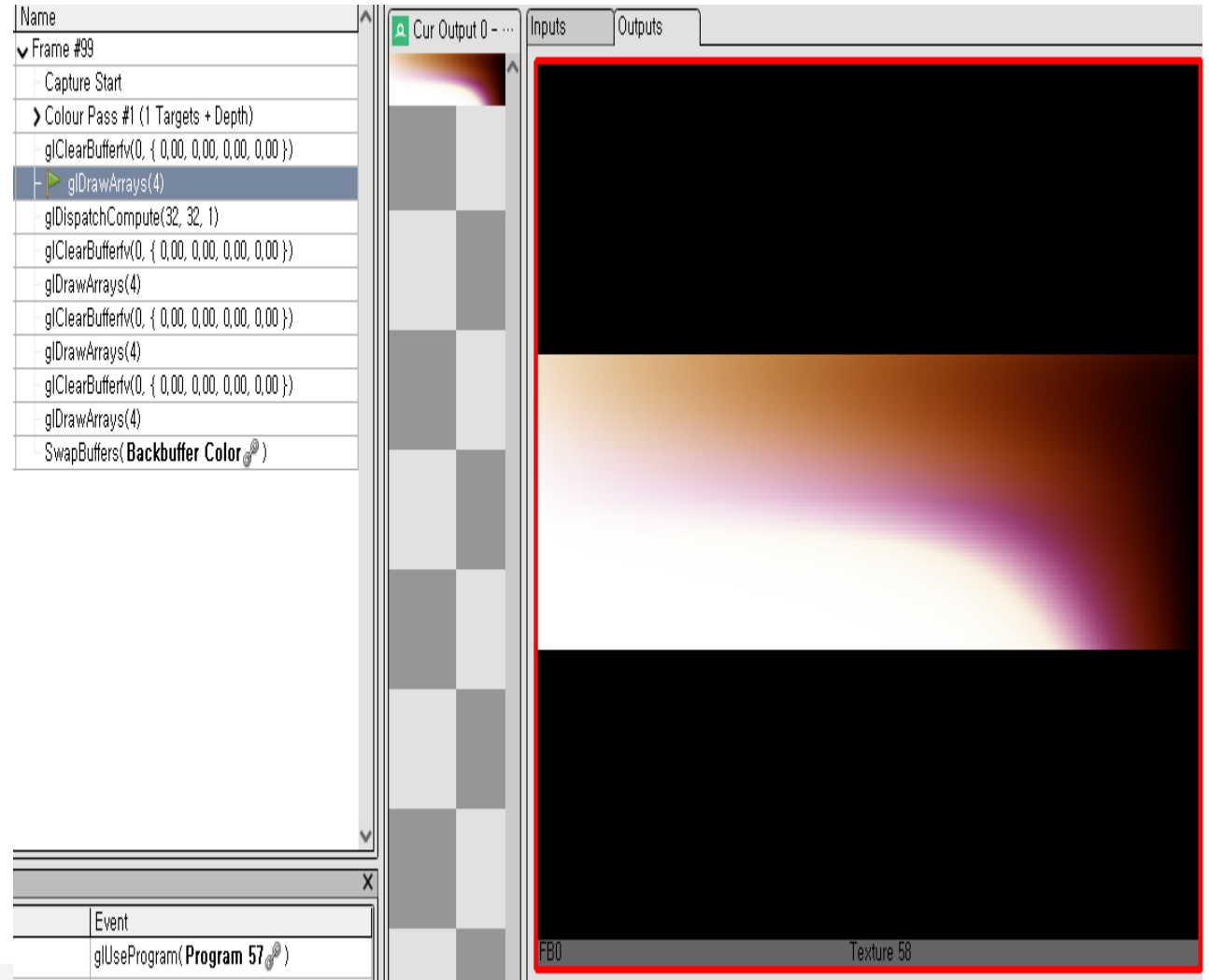


# Transmittance LUT



- 투과율 렌더링

- Fullscreen Quad
- $uv.x$  : 고도(0 Bottom, 1 Top)
- $uv.y$  : 천정 고도각
- 위 고도 및 천정 고도각으로  $WorldPos$ ,  $WorldDir$  계산
- 레이마칭 수행하여  $Extinction(Luminance \text{ 원본에서 남은 비율})$  누적 계산
- 레이마칭 수행 값을 텍스처에 저장



# SkyView LUT



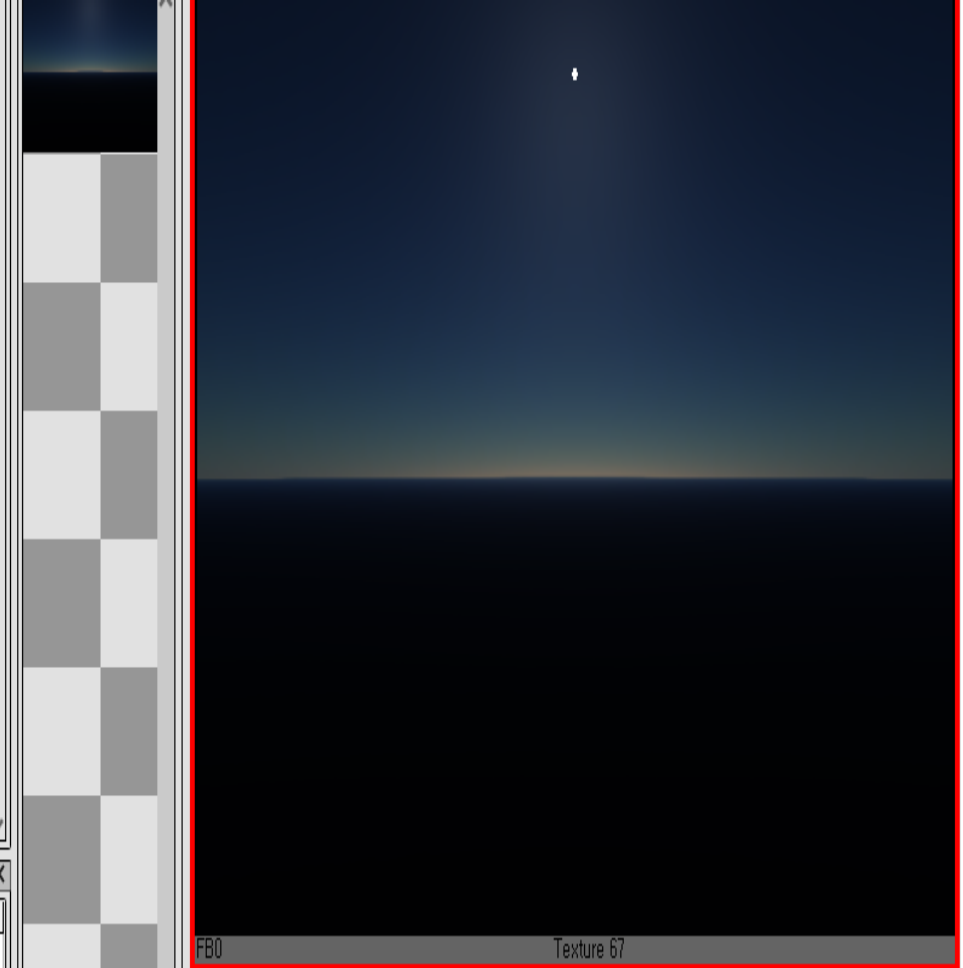
- 하늘 렌더링
  - Fullscreen Quad
  - 입력 텍스처 : Transmittance LUT
  - $U_v.y$  0.5 : 지평선
  - $U_v.y$  0.5 ~ 1.0 : 지평선 위, 경도
  - $U_v.y$  0.0 ~ 0.5 : 지평선 아래, 경도
  - $U_v.x$  : (0.0, 0.5, 1.0) -> (0도, 90도, 180도), 위도
  - WorldDir : 위도, 경도 이용 계산
  - WorldPos : 카메라 위치(camera + planet bottom radius)
  - SunDir : 태양 위치
  - 레이마칭 수행하여 남아있는 Luminance 누적 계산 :  
 $Luminance * Mie * Rayleigh * Transmittance\ LUT\ sample$

```
Capture Start
> Colour Pass #1 (1 Targets + Depth)
glClearBufferfv(0, { 0.00, 0.00, 0.00, 0.00 })
glDrawArrays(4)
glDispatchCompute(32, 32, 1)
glClearBufferfv(0, { 0.00, 0.00, 0.00, 0.00 })
glDrawArrays(4)
glClearBufferfv(0, { 0.00, 0.00, 0.00, 0.00 })
glDrawArrays(4)
glClearBufferfv(0, { 0.00, 0.00, 0.00, 0.00 })
glDrawArrays(4)
SwapBuffers(Backbuffer Color)
```

X

Event

glUseProgram(Program 66)

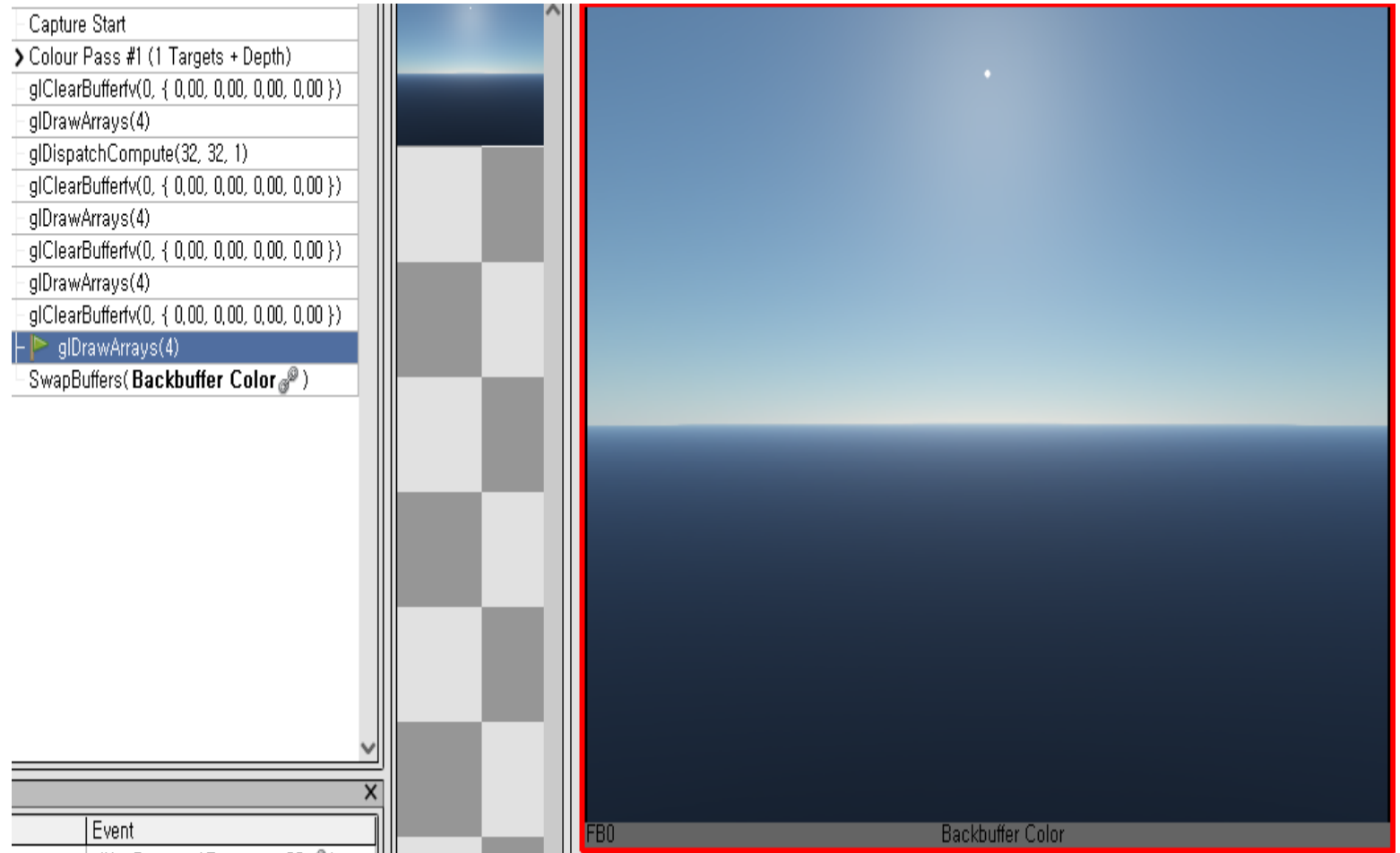


# Final PostProcess



- 노출 보정

- 입력 텍스처 : *SkyView LUT*
- 출력 : *Back buffer*
- 밝기 조정



# Misc

---



- 소스 코드

- <https://github.com/ugeun-ji/SkyAtmosphere>
- 실행 : SkyAtmosphereOpenGL.sln -> Debug, x64

- 미구현 사항(추후 구현)

- Multi Scattering LUT : Compute Shader, 코드에는 있으나 디버깅중(출력 적용은 안되어 있음)
- Arial Perspective LUT : 장애물 고려 Luminance 계산