

Intensity normalize using UBO

이번 주 진행사항

- UBO를 이용한 min-max mipmap normalize

UBO min-max mipmap normalize

- last_mip을 저장하는 uniform buffer UMX 생성
- SSBO 계산을 위해 `UMX->bind_base_as(GL_SHADER_STORAGE_BUFFER, 7);`
 - `glBindBufferBase`의 wrapper, `bind_as`와 `bind_base` 합친 버전
 - `effect->bind_shader_storage_buffer`에서도 동일한 기능을 수행
 - `bind_base_as`에 비해서 variable name으로 `get_shader_storage_block_binding` 과정이 추가
 - 지정된 `target(GL_SHADER_STORAGE_BUFFER)`의 `index(7)` 위치에 `buffer(UMX)` binding하여 shader가 해당 index 타고 buffer access
 - SSBO처럼 read&write 가능
- minmax mipmap MMX의 `last_mip()`을 UMX에 저장
- UMX 사용해서 normalize
- rendering process
 - `compute_glare`
 - DST에 pixel별 각 ghost의 2d gaussian intensity compute
 - minmax process
 - `build_minmax`
 - DST를 통해 MMX texture에 minmax mipmap setting
 - `copy_last_mip`
 - UMX에 `bind_as`를 통해 SSBO binding
 - `MMX->last_mip`을 가져와서 UMX에 `vec2` type 저장
 - `normalize`
 - UMX를 통해 min, max로 normalize하는 TEMP texture 생성
 - `b_normalize = true` 일 때, TEMP를 DST에 copy

benchmark 결과

- texture (sampler2D) based normalize
 - mean GL time = 0.362 ms
 - VGlare_FSD2 = **0.257** ms
- UBO based normalize
 - mean GL time = 0.376 ms
 - VGlare_FSD2 = **0.261** ms
- unnormalize (minmax process 포함)
 - mean GL time = 0.359 ms
 - VGlare_FSD2 = **0.257** ms
- unnormalize (minmax process X, normalize 이전 버전)
 - mean GL time = 0.273 ms
 - VGlare_FSD2 = **0.177** ms

추후 진행 계획

- 중간 과정에서 TEMP texture 생성 없이 normalize를 수행할 수 있는지 시도