

~호수 시뮬레이션 게임~ 이것은 호수입니다.

팀명 풀악셀드리프트

팀원 이 성학 (소프트웨어학과)

지도교수 고 옥

개발 동기 및 목적

최근 모바일기기의 성능이 발전하면서, 다양한 렌더링 기술을 활용한 모바일게임들이 출시되고 있다. 그럼에도 사실적인 물 표현을 구현한 게임은 적은 편이다.



(모바일게임 '원신'의 플레이화면. 동적인 물체에 대한 반사효과도 구현했지만, 한 물체가 겹쳐 보이거나, 각도에 따른 Fresnel 반사가 제대로 나타나지 않는다.)

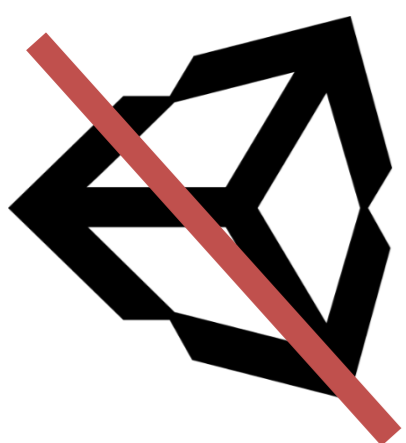
이에 모바일 환경을 위한 물 렌더링 모델을 개발하여, 적은 성능으로도 사실적인 물 표면을 나타내고자 하였다.

개발 내용

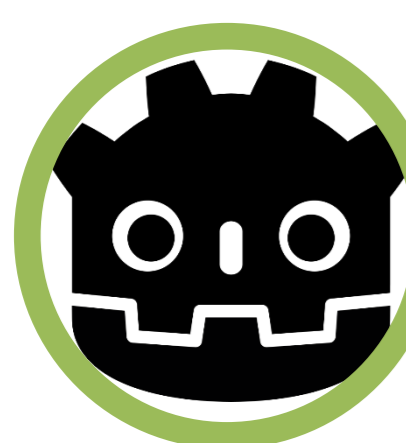
Plan



강력한 성능을 가지고 있지만, 포토 리얼리스틱한 렌더링을 중점으로 하는 특징이 모바일 환경과 맞지 않다고 판단하였다.



모바일 빌드가 간단하고 에셋 스토어를 활용할 수 있지만, GC등의 최적화 문제가 모바일에서 크게 느껴지고 셰이더 접근성이 불편하다.



간편한 셰이더 작성이 가능하고, 오픈소스 엔진이라 렌더링 파이프라인에 대한 내용도 쉽게 알아볼 수 있어 개발 엔진으로 선택하였다.

Features



Reflection

단순히 Sky만 반사하지 않고, 동적인 물체에 대한 반사도 구현한다.



Refraction

물 표면의 Height, Normal map을 적용하고 그로 인한 물 속의 굴절 효과를 구현한다.



Splash

강체가 물 표면에 닿을때 출렁이는 효과를 구현한다.



Environment

시간의 흐름, Fog, Rain등의 환경과 물 표면의 상호 작용을 구현한다.

주요기술

◆ Screen Space Reflection(SSR)

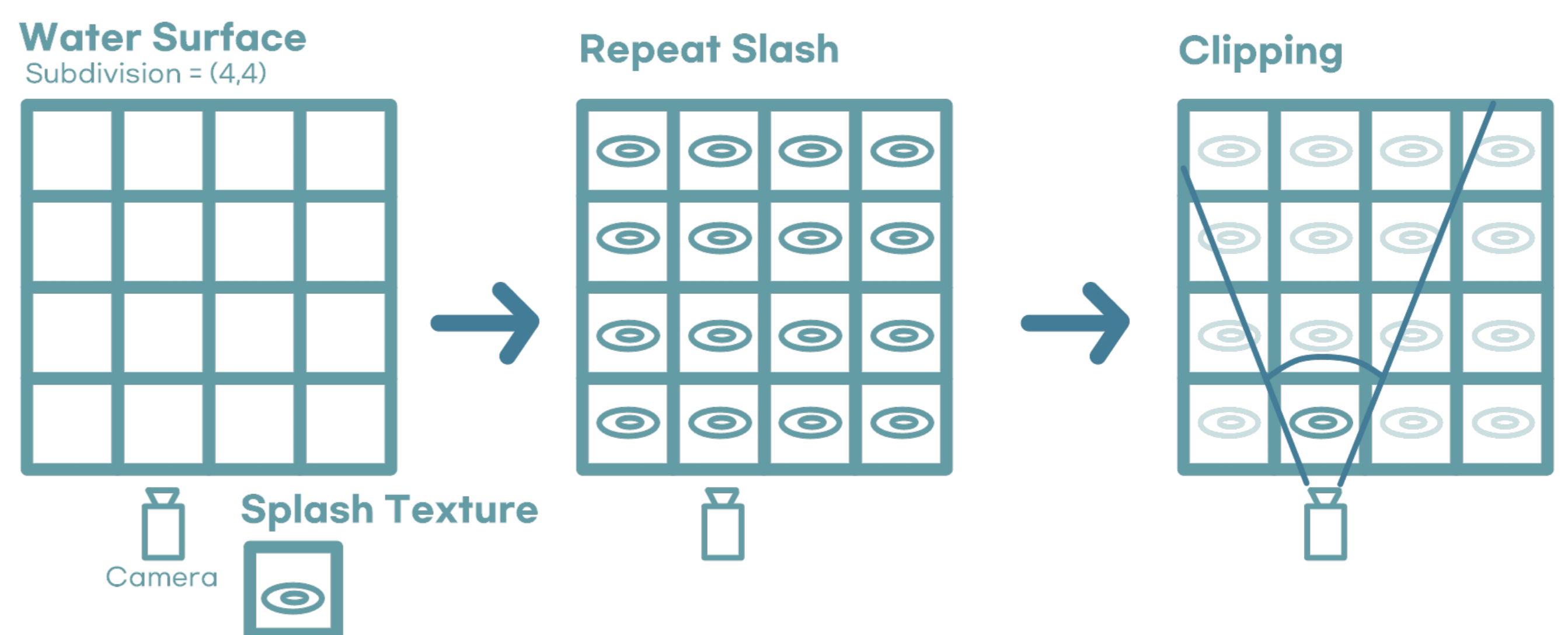
SSR은 Screen버퍼와 Z-버퍼를 이용하여 글로벌한 반사효과를 적은 성능으로 나타내는 기술이다.

1. Fragment 셰이더의 Normal정보를 이용해 반사 벡터를 계산
2. Ray-Marching과 비슷한 방식으로, 표면에서 반사벡터 방향으로 진행하며 Z-버퍼의 값을 비교한다.
 - 2-1. Depth의 경우 x,y와 다르게 Linear Interpolation을 적용할 수 없기 때문에 따로 Depth Interpolation해주어야 한다.
3. Ray가 Z-버퍼에 닿으면 그 부분의 Screen UV에 해당하는 Color를 반환한다.

◆ Subdivision Splash Mapping

물에 강체가 닿을 경우 그 부분의 Texture에 색을 칠하며, Splash된 Texture를 Buffer texture에 저장하여 다음 프레임에 사용하여 Splash 효과를 구현한다.

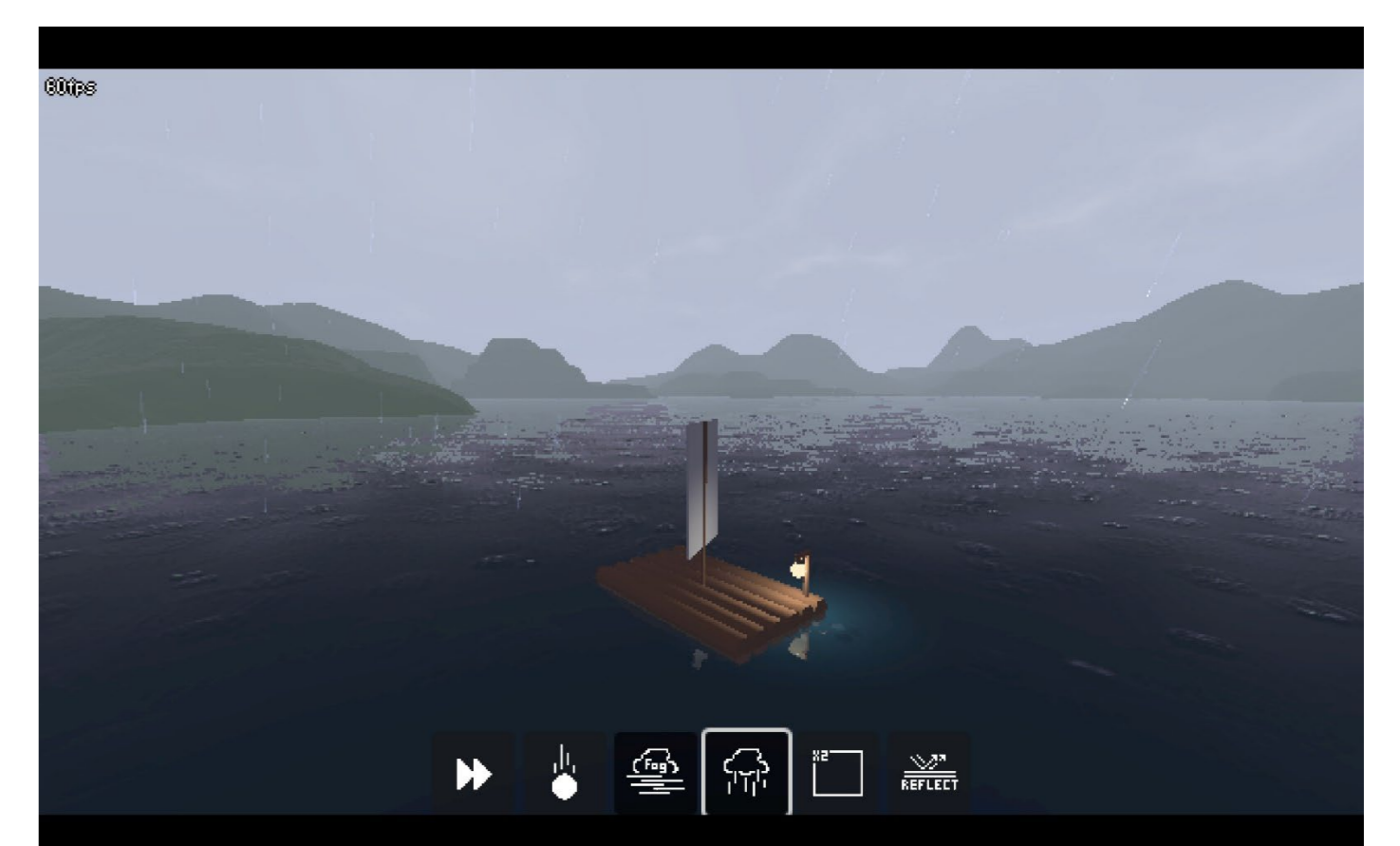
물 표면의 면적이 커지면 커질수록 Splash Texture도 커져야 하는 점을 보완하기 위해, 작게 나뉘어진 Splash Texture를 Repeat하여 물 표면에 적용하고, 멀리 떨어진 부분을 Clip하는 방식으로 최적화하였다.



결과 및 분석

◆ 테스트 결과

- 움직이는 물체에 대한 반사와 반사된 표면이 Splash에 의해 변형되는 등의 사실적인 효과를 구현하였다.
- 절반의 해상도에서 60fps를 유지, 원본 해상도에서 평균 40fps를 보여주었다.



(테스트 환경 : Samsung Galaxy Tab S7)

◆ 보완 요소

- 물 표면에 동적인 Level of Detail(LoD)를 적용하여 최적화
- Lighting처리를 보완하여 자연스러운 Fog 효과 구현

오픈소스 URL

<https://github.com/JyarlJung/LakeSimulation>

