|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **주차** | 29주차 | **기간** | 2023.01.09~ 2023.01.15 | **지도교수** | (서명) |
| 이번주 한 일 요약 | - 씬을 캡처하고, Mesh Distance Field를 기반으로 한 페인팅 구현  - RVT(런타임 버추얼 텍스처)을 사용해 페인팅 구현  - 반투명 메타볼 개선 | | | | |

<상세 수행내용>

연구과제 슬라임 상호작용 중 표면 페인팅 효과를 계속해서 구현 중에 있다.

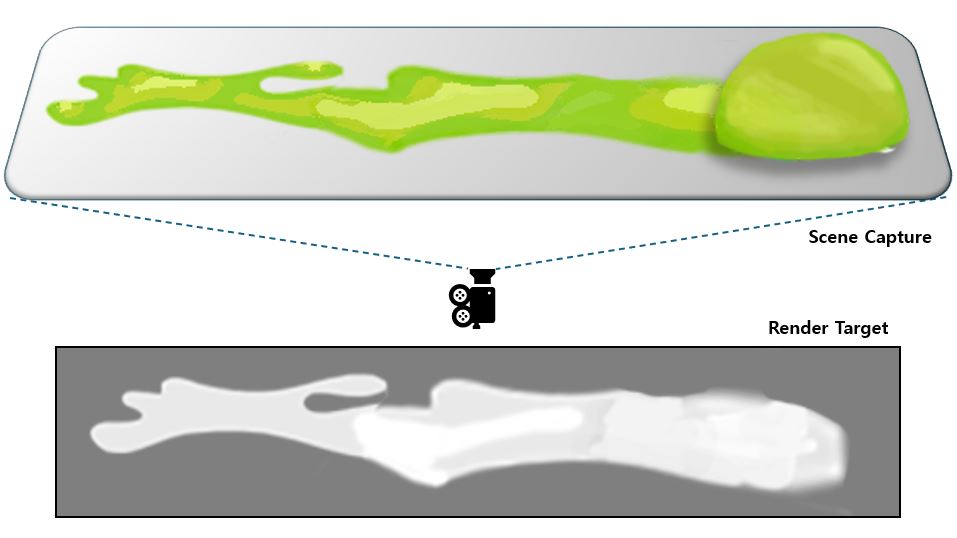
일반적인 표면 페인팅 3가지 방식

* 메시 정점에 페인팅 (Vertex Painting)
* 라이트맵 좌표를 사용하여 페인팅
* UV 공간의 텍스처에 페인팅

이 중에서 표면을 슬라임 점액으로 더럽히기 위해 ‘UV 텍스처를 활용하는 방식’을 채택했다.

페인팅이 된 영역과 그렇지 않은 영역의 위치를 구분해서 슬라임이 점액으로 페인팅 된 영역에 들어섰을 때는 일시적으로 감소되던 크기가 멈추는 메커니즘을 만들고자 한다.

UE에서 ‘렌더타겟’ 텍스처를 활용해 타겟팅되는 영역을 텍스처로 그릴 수 있다.  
즉, 위의 메커니즘처럼 이미 지나온 점액 영역을 구분하기 위해서 렌더타겟 텍스처에 마스킹된 영역을 추출할 수 있으면 된다.



==============================================================

(페인팅 자국을 남긴 위치를 알 수만 있다면 다른 방식으로도 가능하다.   
런타임 버텍스 페인팅 플러그인에서는 렌더타겟 텍스처를 사용하지 않고 페인팅된 메시 정점을 실시간으로 읽을 수 있다.)

[Fire Showcase - Runtime Vertex Color Paint & Detection Plugin] : <https://www.youtube.com/watch?v=A80A4GcPD2g>

(반대로 UE에서 지원하는 Decal을 사용하는 방식은 페인팅되는 좌표를 따로 관리하는 것이 필요하다. 무엇보다 Decal 마다 DrawCall이 하나씩 발생하기 때문에 지속적인 페인팅이 필요한 경우에는 적합하지 않다)

[UE5 Paint in Game 2차 구현] <https://www.youtube.com/watch?v=5p-yw-6oTY0>

* 슬라임 액터을 타겟팅하는 메시 간의 표면 거리에 따라 페인팅 작동
* M\_SlimeCharacter(슬라임 캐릭터 머테리얼) 반투명 메타볼 효과 구현(opacity mask)
  + ~~Translucent(반투명) 머테리얼이 디퍼드 렌더링되면서 메타볼 효과를 받지 못했음.~~
  + Opaque Masked(불투명 마스크) 머테리얼에서 디더링(dithering)을 하는 것으로 반투명 메쉬를 렌더링한 다음 디퍼드 렌더링되는 배경(카메라 화면 내 픽셀)과 결합
  + (영상 속 그림자 잔상이 출력되는 문제는 프로젝트 옵션에서   
    Virtual Shadow Maps -> Shadow Maps로 변경하는 것으로 해결)

[UE5 Paint in Game 3차 구현] <https://www.youtube.com/watch?v=TUB8t2oXT1k>

* RVT(런타임 버추얼 텍스처) 사용해 페인팅 범위 설정
* 슬라임 점프 제어

[UE5 Paint in Game 4차 구현] <https://www.youtube.com/watch?v=lXhfecYxwE0&feature=youtu.be>

* 렌더타겟 알파 마스킹 버텍스 노말 효과

[UE5 Paint in Game 5차 구현] <https://www.youtube.com/watch?v=pC3dSPRb64g&feature=youtu.be>

* 마테리얼 펑션으로 마테리얼 레이어 활용(점액 부위에 버텍스 노말 작용)
  + 마테리얼 펑션으로 마스킹 영역에 다채로운 페인팅 효과를 줄 수 있음

**슬라임 표면 페인팅 구현 목록**

* ~~렌더타겟으로 마테리얼 페인팅 구현~~
* ~~렌더타겟으로 액터 위치기준 depth 거리 탐색해서 페인팅 구현~~ (Update)
* ~~마스킹 영역에 복합적인 데칼 효과(메타볼 점액) 적용~~ (Update)  
  (normal map, Height map, AO map 등 PBR 머테리얼 작업까지 진행해서 마무리할 예정)
* 마스킹 영역에서 슬라임 액터가 특수 액션 구현(점액 위에서 빠르게 움직이기)
* 오브젝트 대상으로 마테리얼 페인팅 구현
* 마테리얼 블루프린트 결과물을 HLSL 코드로 변환

페인팅은 UV 좌표를 기준으로 판정하기 때문에 레벨에 배치된 메시의 UV를 정교하게 수정해야 하는 추가 작업이 필요하다.  
 메시에 대해 블루프린트 클래스를 만들어서 페인팅 로직을 처리하고 있다. 레벨에 배치될 메시마다 블루프린트 클래스를 제작하는 프로세스를 효율적으로 다듬을 필요가 있다. ~~Editor Utility BluePrint로 관리하는 액터를 제작할 계획.~~ (디버깅용도, 따로 관리할 수 있는 엔지니어링이 필요)

|  |  |  |  |
| --- | --- | --- | --- |
| **문제점 정리** | 맵 전체에 페인팅을 하기위해서는 페인팅될 메시마다 서로 다른 렌더타겟 텍스처, RVT(런타임 버추얼 텍스처)를 로드하고 있어야 한다. 벽과 바닥을 포함해 커다란 가구, 아키텍처에 대해서도 표면 페인팅 효과를 주려면 많은 텍스처를 로드하기 때문에 메모리 풀 제한이 걸린다. | | |
| **해결방안** | - 렌더타겟 텍스처 RBGA 비트 중 단일 채널 사용  - 다행인 건 3분 분량의 멀티플레이 맵은 넓지는 않다는 점 | | |
| **다음주차** | 30주차 | **다음기간** | 2023.01.16 ~ 2024.01.22 |
| **다음주 할일** | 슬라임 표면 페인팅 효과 이어서 작업   * 파티클 충돌 페인팅 구현 * 렌더타겟 마스킹 영역에서 액션 구현 * 렌더타겟, RVT 데이터 풀 프로세스 개선 * 오브젝트 타겟 페인팅 구현 (나이아가라 파티클 접목) | | |
| **지도 교수**  **Comment** |  | | |