|  |  |  |  |
| --- | --- | --- | --- |
| [Cranding(크랜딩)] | | | |
| **9주차** | **2024.02.18~2024.02.24** | **작성자** | **정경서** |
| **주간 회의** | | | |
| **회의 날짜** | | 2024.02.21 | |
| 1. 창업신청  * 사업자 등록 완료  1. 서버 클라 통신 오류 같이 고쳐보기 | | | |
| **이번주 한 일** | | | |
| 1. 신동엽(서버)   - 클라이언트 이동 동기화  - 디자인 패턴, 명령 패턴 공부  (노트북 고장으로 인하여 내용을 간략하게 적었습니다. 다음주 보고서에 모두 추가하겠습니다)   1. 정경서(클라이언트)   1. 디퍼드 렌더링 공부  텍스트, 사람, 인간의 얼굴, 의류이(가) 표시된 사진  자동 생성된 설명   * 포워드 렌더링은 오브젝트를 그릴 때마다 라이팅을 즉시 처리하는 반면, 디퍼드 렌더링은 라이팅을 지연시켜 정보를 저장하고 스크린 스페이스에서 처리합니다. 이로 인해 디퍼드 렌더링은 오브젝트 개수에 제약을 받지 않고, 픽셀 단위로 처리되므로 연산량이 줄어들어 성능이 향상됩니다. * 라이트 처리에 필요한 정보로는, 픽셀의 깊이, 노말, 디퓨저 등이 있으며, 이를 통해 3D 공간에 위치를 얻는다.   (해당 내용을 필기한 워드 파일이 날아갔습니다 ,, ㅠㅡㅠ)  유티니 코리아에 올라온 렌더링 기법에 대한 개념 영상을 보며 공부를 했다.  디퍼드 렌더링의 개념에 대해 확실하게 이해를 한 뒤 코드 공부를 시작했다  2. Labproject 08-9-1 공부  해당 코드는 3학년 2학기 디퍼드 렌더링 예제로 교수님이 올려주신 코드다.  해당 랩프로젝트를 기반으로 디퍼드 렌더링을 공부하려 했으나, 이 코드 디퍼드 렌더링이 적용된 코드가 아니라는 사실을 코드 공부 4일차에 알게 되었다.  빛 계산이 텍스처를 통해 계산 되는 것이 아닌 버텍스 쉐이더에서 넘어오는 값을 가지고 픽셀 쉐이더에서 계산되는 점과 코드 상 각각의 키를 눌렀을 때의 과정을 볼 때 G-Buffe에 기록된 텍스쳐가 아닌 키를 눌렀을 때 후처리를 통해 보여주는 것 같다.  이때 생각난 것이 해당 코드를 통해 포스트 프로세싱을 하는 방법을 알아내고 멀티 렌더 타겟을 통해 각 장면을 텍스쳐로 뽑아내는 과정을 공부를 하면 그것이 곧 디퍼드 렌더링을 하는 방법이 아닐까? 하는 생각을 하고 공부를 이어 나갔다.  3. 프로젝트 오류 해결하기  프로젝트를 처음 열 때 설정을 잘못했는지  텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진  자동 생성된 설명  해당 위치에서 계속해서 오류가 발생하며 프로젝트 실행이 되지 않는 문제가 발생했다  어선셜 오류가 난다고 하여 다양한 방법을 통해 오류를 해결하려고 했지만 시간도 많이 쓸 것 같고 의미가 없는 작업이라고 판단하여 새 프로젝트를 만들어서 코드를 옮기는 것으로 해결하였다.   1. 홍예나(클라이언트)   1. 애니메이션 상, 하체 분리  키가 눌렸을 때 상, 하체가 분리되는 것은 나중에 조건으로 추가하려고 생각 중이기 때문에 우선 애니메이션의 상, 하체가 분리된 채 잘 실행되는 것에 집중하여 구현하였다.  <기존의 애니메이션 코드(블렌딩x)>    <상, 하체 분리를 추가한 애니메이션 코드>    저번에 상체인지 구분할 수 있도록 GameObject 클래스 내부 변수로 m\_bUpperBody를 생성하였고, 상체일 경우 해당 변수의 값을 true로 변경해주었다.  이 m\_bUpperBody 변수를 이용하여 상체인 경우 상체 부분의 transform 행렬을 가져와서 저장하도록, 하체인 경우 하체 부분의 transform 행렬을 가져와서 저장하도록 하였다.  otherNum이라는 변수를 생성한 이유는 상, 하체 분리가 필요한 경우가 총을 쏠 때 뿐이기 때문에 애니메이션 2개만 이용한다는 점에서 착안하였다. 달리면서 총을 쏘게 될 경우, 총 쏘는 애니메이션을 pAnimationSet2에 넣도록 하기 위해 사용할 예정이다.  그런데 위와 같이 구현할 경우, 애니메이션이 아주 잠깐 실행되다가 터지는 오류가 발생하였다.    오류를 살펴보니 interpolate 함수에 이상한 값이 넘어가는 것을 확인할 수 있었다.  해당 문제의 원인은 m\_ppxmf4x4KeyFrameTransform[j]가 j번째 뼈의 transform행렬이 아닌, j번째 키프레임의 전체 뼈 행렬이었기 때문에 터지는 것이었다.  GetSRT 함수의 내부를 살펴보면    위와 같이 m\_ppxmf4x4KeyFrameTransform[m\_nKeyFrames][nBone]으로 사용하는 것을 알 수 있다.  따라서 GetSRT를 직접 사용하여 j번째의 transform 행렬을 가져오기로 하였다.    따라서 위와 같이 코드를 구현하게 되었다.  이제 애니메이션의 상, 하체가 분리되어 실행은 되지만 또 다른 문제가 발생한다.  하체의 fPosition을 사용하다보니, 상체와 하체 중 길이가 더 짧은 애니메이션이 먼저 끝난 후  마지막 포지션 그대로 더 긴 쪽의 애니메이션이 끝날 때까지 그대로 멈춰져있다.  해당 사항은 아직 해결하지 못하였지만 상, 하체의 fPosition을 모두 사용하면 해결할 수 있을 것 같다.  2. 애니메이션 블렌딩  플레이어의 애니메이션을 바꾸고 여러 코드들을 추가하다보니 애니메이션 블렌딩 코드가 꼬여서 키를 눌렀을 때, 키를 뗐을 때 총 두 번 애니메이션 블렌딩이 진행되어야 하는데 키를 눌렀을 때 애니메이션 블렌딩은 블렌딩 코드는 타는데 실행은 안되고, 키를 뗐을 때의 애니메이션 블렌딩은 실행은 잘 되지만 2번 연속으로 실행되는 문제가 발생하였다.  <기존의 애니메이션 블렌딩 코드>    <변경된 애니메이션 블렌딩 코드>    뭐가 문제인지 찾아보니, 키를 뗐을 때, 달리는 애니메이션에서 idle 애니메이션으로 변경될 때 애니메이션 블렌딩 부분이 아예 실행되지않는다는 문제를 발견하였다.  그 원인을 찾아보니 키를 눌렀을 때의 m\_fBlendingTime은 1.0f가 되기 전 for문을 빠져나오기 때문에 m\_bIsBlending이 false가 되어야하는데 false가 되지 않고 넘어가게 된다.  그 후 m\_bIsLastBlending과 m\_bIsBlending 둘 중 하나만 true여야 블렌딩 코드가 타지는데 m\_bIsBlending이 false로 바뀌지 못했기 때문에 둘 다 true가 되어 키를 뗐을 때의 애니메이션 블렌딩 코드가 실행되지 않는 것이었다.  따라서 위 코드와 같이 둘로 나누어서 구현하였고, 아래와 같이 update 함수의 538번 if문도 변경해주었다.    위와 같이 구현하니 키를 뗐을 때 애니메이션 블렌딩이 2번 연속 실행되던 문제가 해결되었다.  하지만 여전히 키를 눌렀을 때 애니메이션 블렌딩 코드는 실행되지만 블렌딩은 되지 않는 문제는 해결하지 못하였다. | | | |
| **다음주 할 일** | | | |
| 1. 신동엽(서버)  * 회전 동기화 * 컨텐츠 개발 * 회전 동기화 * 서버프레임워크작업  1. 정경서(클라이언트)  * 디퍼드 렌더링을 통해 외곽선 추출하기 * 포스트 프로세싱 코드 공부  1. 홍예나(클라이언트)  * 애니메이션 블렌딩 * 상하체 분리 마무리 * 가우시안블러 개념 공부 | | | |
| **비고** | | | |
|  | | | |