|  |  |  |  |
| --- | --- | --- | --- |
| **[ The Client : desert (의뢰인:사막) ]** | | | |
| 7 주 | 2022. 2. 6 ~ 2022. 2. 12 | 작성자 | 윤성주 |
| 이번주  한 일 | **[0] 공동**  02. 08 회의   * **맵** * 오픈 월드가 아닌 맵을 나눠 2개의 씬으로 구현하는 것으로 변경  (맵의 크기가 크면 터레인 가장자리 처리가 어렵고, 프레임이 떨어질 것을 생각하여 변경함.) * **플레이어 무기** * 무기(칼) 크기 키우기 * 무기의 충돌체는 무기의 크기에 맞도록 설정 * **안개** * 멀리 있는 물체를 흐리게하는 용도. 가까운 물체(몬스터)는 흐리게 하지 않음. * **플레이어 회전** * 기존 선형 보간을 쿼터니언을 이용한 구형 보간으로 변경. * **카메라 회전** * 기존 카메라 Up 벡터 기준 회전만 가능한 것이 아닌 Right 벡터 기준으로도 회전 가능하도록 변경 * **개강 전 추가로 할 일** * 윤성주: 애니메이션 블렌딩, 카메라 조정, 서버 프레임워크 * 최경훈: 안개, 씬전환(맵 변경), 오브젝트 배치   **[1] 윤성주**  **주간 목표**   1. **플레이어 애니메이션 블렌딩 모두 구현** 2. **플레이어 idle 구현 (공격 모드일 때의 idle)** 3. **~~마우스로 카메라 회전 구현 & 플레이어 좌표축 변경 구현~~** 4. **플레이어 회전 선형보간에서 구형보간으로 변경하기**   **진행률: 25%**  **작업 내용**   1. **카메라 회전 후 플레이어 이동 시 플레이어의 좌표축 카메라에 맞게 변경** 2. 그림과 같이 세타만큼 플레이어 회전하는 방법 텍스트이(가) 표시된 사진     자동 생성된 설명 => 카메라 위치의 y값과 플레이어 위치의 y값이 다르므로 위와 같은 방법으로 구현 시 플레이어보다 카메라가 더 많이 회전. 즉, 세타가 잘못 구해짐! 3. 카메라의 Look벡터만 사용 카메라의 Look벡터의 y값은 0.f로 하고, 정규화하여 플레이어가 이 벡터로 보간하여 회전하도록 한다.   => 구현 성공!   1. **플레이어 이동 카메라의 Look, Right 사용하여 하도록 변경**  * 기존 방법 (카메라가 회전하지 않을 때)   오른쪽으로 회전 시 (1, 0, 0)으로 회전시켜 현재 플레이어의 좌표축에 맞게 회전이 되지 않음.   * 변경 방법     => 카메라의 Right 방향으로 이동하도록   1. **플레이어 회전 (쿼터니언) 아직 구현하지 못함**  * 기존 방법 (벡터를 선형보간)   벡터 Lerp함수를 이용해서 보간하여 플레이어를 회전. 이 방법은 180 회전 시 보간이 안되는 문제가 있었음. (180도 회전의 경우 선형 보간 시 값이 한순간에 바뀐다. ex> (1, 0, 0) -> (-1, 0, 0)은 x값이 음수로 바뀌는 순간에 회전하게된다. )   * 변경 방법 (쿼터니언 구형보간) Matrix4x4 이름 공간에 있는 보간 함수 사용   **[2] 최경훈**  **개발 내용**     1. 컴포넌트 패턴을 이용하여 맵 오브젝트 컬링을 구현했다. 절두체 컬링도 직접 함수를 구현했는데, 알고보니 이미 DirectXCollision.h에 이미 절두체 컬링 함수가 있었다. (나중에 수정?) 어쨌든 덕분에 프레임은 최소 50정도 올랐다.      1. 사진과 같이 깊이 선형 안개를 구현해보았다. 하지만 아직 스카이 박스엔 처리가 안되고, 다른 부분도 어색해서 수정을 할 필요가 있다.       다음 사진과 같이 플레이어와 맵 오브젝트 충돌을 구현하였다. DirectXCollision에 구현된 변수와 함수를 이용하였고, 충돌체 정보는 유니티에서 Mesh를 불러올 때 BoundingBox 정보도 함께 불러와서 사용했다. | | |
| 다음주  할 일 | **[0] 공동**   1. 지금 작업중인 일이 끝나면 코드 합치기 2. 카메라 각도 결정하기   **[1] 윤성주**  주간 목표   1. 플레이어 애니메이션 블렌딩 모두 구현 2. 플레이어 idle 구현 및 idle 애니메이션 버그 수정 3. 플레이어 회전 쿼터니언 사용으로 변경 4. 카메라 회전 위아래 (Right 축) 회전 구현 (마우스 움직임에 따라 회전하도록)   **[2] 최경훈**  주간 목표   1. 플레이어가 맵 오브젝트를 뚫지 못하게 충돌처리 할 것. 2. 선형 안개 처리를 자연스럽게 해볼 것 3. 씬을 두 개로 나눠서 맵 오브젝트를 배치해 볼 것. | | |
| 문제점 | **[1] 윤성주**  **<마우스로 카메라 회전 시 카메라와 플레이어 거리 가까워지는 버그>**     * 카메라 회전에 딜레이를 줘서 부드럽게 이동하도록 구현된 부분인데, 이 부분에서 거리를 조절하면서 카메라와 플레이어 거리가 가까워지는 현상 발생. 이 부분을 삭제하니 정상적으로 잘 돌아갔다. 추가로, 카메라 무빙이 부드럽지 않은 지 테스트 했으나 플레이에 지장이 없어보여 그대로 사용할 예정.   **<마우스 회전->플레이어 회전 후 마우스 회전 시 이전 회전한 만큼 회전되는 버그>**   * 마우스 회전 시 플레이어가 회전한 만큼 마우스를 회전시키는 코드가 있었음. 카메라는 플레이어가 회전 시 회전하지 않으므로 이부분을 삭제하였더니 버그 수정.     => 이 코드 삭제  **<공격모드 Idle 추가 중 애니메이션 버그 발견>**   * idle 애니메이션이 되지 않았었다. 코드에서 idle로 계속 바꿔주어서 그런듯하다. idle일 경우 idle로 바꾸지 않도록하는 코드를 추가해야 할 듯   **<삽질을 하면서 느낀점>**   * 플레이어의 회전이 카메라의 Right, look 벡터만 사용하면 간단하게 구현이 가능한데 행렬에 대한 지식이 부족해서인지, 카메라에 대한 이해가 부족해서 인지 계속 어려운 방법만 생각했고 생각한 바를 제대로 구현하지 못해서 자괴감이 느껴졌었는데 경훈이의 도움으로 방법을 알아내어 바로 해결했다... 구현하기 전에 좀 더 여러가지 방법을 생각할 필요가 있겠다고 느꼈다.   **[2] 최경훈**   1. 절두체 컬링을 직접 구현할 때 계속 이상하게 컬링이 되는 버그가 있었다. 수학적 공식이 잘못된줄 알고 고생했지만 알고보니 포인터 변수를 잘못 사용하고 있었다. XMFLOAT3\* pPos 변수를 XMLoadFloat3(pPos) 이렇게 사용해야 하는데 XMLoadFloat3(&pPos[i]) 이렇게 써서 버그가 났다. 빨간줄도 안떠서 찾는데 시간이 좀 걸렸다. 앞으로 꼼꼼히 확인할 것. 2. 맵 오브젝트와 플레이어 충돌을 구현할 때 충돌 판정이 이상한 버그가 있었다. 매 프레임마다 BoundingBox정보를 월드행렬과 계산하여 업데이트 하는데, 한 오브젝트가 각 계층구조마다 업데이트해서 중복된 계산으로 인한 버그였다. 그래서 딱 한번만 충돌체 정보를 업데이트 하여 해결했다. | | |