| 주차/일자 | 18주차 / 4.21~4.27 | 작성자 | 김정훈 |
| --- | --- | --- | --- |
| 개요 | 정기 보고 | | |
| 회의 내용 | 2024.04.27 정기 회의   * 작업 내용 병합   + 씬 전환   + 캐릭터 선택 및 캐릭터 선택 결과 다음 씬에 적용 * 중간 발표 PPT 제작 및 인지중인 오류 종합(해결 가능 or 불가능) * 발표 중 발생 가능한 오류(예외) 테스트   + 클라이언트     - 여자 캐릭터 바운딩 박스 처리     - 플레이 씬에서 캐릭터 중복 출력 불가능  -> 기획 내용에 따라 캐릭터 중복 선택 불가능 하도록 제작예정     - 모델 텍스쳐 밀림     - 캐릭터 선택 및 씬 전환 때 약간의 딜레이 (모델 로드 때문) (수정중)     - 캐릭터 선택을 많이 시도하면 메모리 누수로 인해 프레임 저하됨 (수정중)     - 프레임 저하 시 IDLE 애니메이션으로 전환     - 180도 반대 방향으로 전환 시 캐릭터 회전처리 안됨     - 바운딩 박스 충돌 처리가 너무 fit하게 처리됨     - felapsedTime의 잦은 사용으로 인해 기기에 따라 플레이 속도가 바뀜     - 캐릭터끼리 충돌처리 안됨   + 서버     - 서버에서의 맵에서 회전 값 미적용  (갈 수 있어 보이는 곳에 못 갈 수도있음)     - 서버 내의 클라이언트를 3명으로 제한 (GameRoom의 갯수를 1개만 만들었기 때문) ( -> 중간 발표 이후 GameRoom을 컨테이너에 담아서 관리하는 매니저를 구현할 예정)     - NPC 이동 동기화 (중간 발표 전까지 해결할 예정)     - 기획했던 sector를 미구현 -> 중간 발표 이후 구현 (NPC의 길찾기가 Sector내에서가 아닌 주변에서 가장 가까운 플레이어를 추적) | | |
| [다음 주 회의 안건]   * 다음 주 면담 후 계획 | | |
| 주간  수행 내용 | [김정훈 - 서버]   * 클라이언트에서 분할한 씬에 대해 기존의 Session::state 변수에 ST\_LOBBY를 추가(ST\_INGAME과 분리) * NPC의 길찾기   + start, target 좌표를 인자로 받아 vector<XMFLOAT3>를 반환하는 함수를 만들어서 NPC의 경로 queue에 넣을 수 있음. NPC는 queue.top()에는 목적지를 가기 위한 다음 경로가 저장 되어 있음. | | |
| [김진선 - 클라이언트]   * 모델 수정   + 몇몇 오브젝트의 바운딩 박스 범위가 메쉬와 차이가 커서 메쉬에 맞게 수정   + 기존의 경우 유니티에서 모델의 메쉬가 가지고 있던 바운딩 박스 값을 가져왔기에 수정이 불가능 -> 이를 유니티의 box collider 값을 가져오도록 수정   + 울타리 텍스처 밀림 수정 * 미션 제작 기반 제작   + 미션용 오브젝트 추가   + 캐릭터가 오브젝트를 바라보고, 일정 범위(바운딩 박스 값 +10)에 들어왔을 경우 상호작용이 가능하도록 제작중      * 플레이어 선택 동기화   + PREPARE SCENE에서 선택한 캐릭터가 PLAY SCENE에 적용 되도록 제작   + 서버 연동하여 각 플레이어가 선택한 캐릭터로 출력됨   + 예시) 우라냐 캐릭터를 선택하고 플레이 한 경우      * 발생 문제점   + 캐릭터 선택시 기존 캐릭터의 몇몇 정보가 지워지지 않고 남아있음 -> 프레임 저하 발생     - 동적할당한 이전 모델 정보가 지워지지 않은 걸로 보임. 계속 수정 중 | | |
| [이상민 - 클라이언트]   * 슬라이딩 벡터 연구      * 일반적인 방법은, 입사벡터 P 를 n 에 바로 투영시키는 것이다. 입사벡터 P 와 법선벡터 n 의 끼인각이 0≤ θ ≤ π/2 일때, P·n 의 값은 음수가 되므로, n 벡터의 역벡터 방향으로 투영벡터가 생성된다. 이렇게 얻어진 투영벡터 n(P·n) 을 입사벡터 P 에서 빼주면, 슬라이딩벡터 S 를 얻을수 있다. * 그러므로 슬라이딩벡터 S 는 * S = P - n(P·n) 출처: [https://toymaker.tistory.com/entry/미끄러짐-벡터-Sliding-Vector](https://toymaker.tistory.com/entry/%EB%AF%B8%EB%81%84%EB%9F%AC%EC%A7%90-%EB%B2%A1%ED%84%B0-Sliding-Vector) [ToyMaker:티스토리] * 이 정보를 토대로 필요한 것을 정리 해보았다.   + 입사벡터 P : Player의 LookVec or VelocityVec   + 법선벡터 n : 바운딩박스에서 충돌되는 면의 법선벡터 * 문제는 바운딩박스의 intersects 함수를 사용하게 되면 충돌되는 면의 정보는 알 수 없다는 것이다. 그래서 충돌되는 면을 찾기 위해 함수를 생성해 다시 탐색하였다. * 현재 가지고 있는 정보는 바운딩박스의 Corner점 8개의 정보와 Center점의 정보, Player의 위치 정보나 방향 벡터      * 이 정보를 토대로 Coner점 4개씩 이어서 면을 만들고 바운딩박스의 중심부터 Player의 위치로 향하는 벡터가 어느 면을 통과하는지 체크하였다. * 체크하는 코드 작성은 수식이 많이 필요하고 불필요한 시간을 줄이고자 OpenAI(ChatGPT 3.5)를 통해서 구현했다. * 바운딩 박스 처리   + 슬라이딩 벡터를 구현하기 위해서 바운딩 박스 처리 방식을 수정하여야 했다.   + 기본적으로 내가 움직일 위치와 오브젝트의 충돌처리를 해서 충돌이라면 움직일 수 없게 하거나 슬라이딩벡터 방식으로 처리 해야하기 때문에 내가 움직일 위치를 계산하는 과정이 필요했다.      * + 바운딩 박스 자체를 내가 이동할 위치로 옮길 예정이다. (현재 작업 중) * npc 이동   + npc 이동은 서버에서 받은 좌표로 플레이어가 움직이게 제작할 예정이다.   + npc의 이동 방식은 MoveForward(), Rotate()만으로 단순하게 작업할 것이다.   + 이동하고 싶은 위치의 좌표와 npc의 좌표의 방향 벡터와 npc의 LookVector를 비교해서 Rotate할 예정이며 이동은 남은 거리를 통해 작업할 예정이다. (계획 단계) | | |
| 특이사항 |  | | |
| 다음 주 수행 계획 | [공통]   * 교수님 면담 및 오류 수정   [김정훈 - 서버]   * NPC이동 동기화 (이동 경로에 따른 동기화) * 추가 구현된 항목들에 대한 멀티쓰레드 테스트   [김진선 - 클라이언트]   * 유니티에서 모델 회전 정보만 뽑아오기 * 미션 기반 구현(플레이어와 상호작용) * 스타팅 씬 정돈   [이상민 - 클라이언트]   * npc 이동 * BoundingBox 처리 * npc의 공격 구현, Player 피격 구현 | | |