| 주차/일자 | 5주차 / 1.21~1.27 | 작성자 | 김진선 |
| --- | --- | --- | --- |
| 개요 | 회의 미진행 주차 | | |
| 회의 내용 | \*회의 미진행 | | |
| [다음 주 회의 안건]   * 대각선 이동 값 수정 방안 | | |
| 주간  수행 내용 | [김정훈 - 서버]   * 서버-클라이언트 연결 오류 해결   + 문제점: 확실히 서버 코드에는 문제가 없었고, 클라이언트에서의 send\_packet(void\* ) 함수의 버퍼 주소를 인자로 보내는 문제. : 디버깅 중 send가 버퍼의 값이 아닌 버퍼의 주소를 보내고 있어서 오류 * 서버-클라이언트 간의 통신은 테스트 패킷으로 잘 되는 것을 확인. * 클라이언트의 recv를 매 프레임으로 할 지 callback으로 할 지 연구: 우선 매 프레임으로 통신하고, 동기화 작업을 마치고 난 후 성능 비교할 예정. | | |
| [김진선 - 클라이언트]  \*여행으로 인해 작업량이 비교적 적음   * 충돌처리 작업중   + 이용희 교수님 프로젝트 중 7-3번을 참고   + 해당 예제에서 bounding 박스값을 받아오는 방식과 현재 사용중인 모델 export 스크립트가 서로 다르기 때문에 비교 후 수정 필요   + b 키 입력시 바운딩 박스를 빨간 line으로 표시하도록 제작 중   + 예제 내의 함수     위: 모델 파일 읽어오는 함수 중 / 아래: model exporter 중     * 기존 졸업작품의 파일 로드 중 bound      * 스카이 박스 변경   + 노을 배경으로 변경 | | |
| [이상민 - 클라이언트]  <일>  휴식  <월>  오류가 있었던 모델 추출을 다시해서 Player\_Test.bin 파일을 생성했지만 원하는 결과물이 나오지 않았다.  그 이유는 또 텍스쳐의 문제인 것 같다. 이 모델 추출 오류해결 방법은 김진선(클라이언트)가 잘 알고 있는 분야이기 때문에 다음 주에 다시 해결 방법을 듣고 수정할 예정이다.  그 후에는 플레이어의 카메라 세팅을 바꿔봤다. 우리의 게임은 숄더뷰로 구현할 것이기 때문에 카메라를 조금 당겨 보았다. 하지만 이 카메라의 Look은 플레이어의 Position 값과 같기 때문에 플레이어의 발을 보는 문제점이 있었다. 그래서 이 플레이어의 위치보다 플레이어가 바라보는 방향으로 5.0f 정도 떨어진 시점으로 카메라의 Look을 변환해 보았다. 그러나 카메라의 회전에 오류가 생겨서 일단 이 부분은 급한 것은 아니니 적당히 카메라의 위치만 옮기고 카메라의 Look은 추후에 수정할 예정이다.  <화>  플레이어의 이동 방법에 대해서 파악했었다. 기본적으로 중력, 마찰, 속도, 최대 속도의 값을 플레이어가 가지고 있으며 이를 토대로 가속도와 감속도의 값을 계산해서 플레이어에게 표현된다. 그 부분에 대해서 여러 과정을 실험해 보았다.    그 결과 이동에 관해서는 대각선으로의 이동을 할 때에 두 가지의 방향성 모두로 이동함수를 동작하는 것이기 때문에 이동속도가 두 배로 적용된다.  이 오류에 대해서는 해결방법이 무궁무진해서 다음 회의 때 안건으로 제출할 예정이다.  <수>  플레이어의 애니메이션에 대해서 파악하고 실험해 보았다.  기본적으로 현재 추출한 모델에는 애니메이션이 2가지가 들어있다.  애니메이션을 추출하는 스크립트는 애니메이션을 갯수만 추출할 수 있는 것이 아니라 모델에서 가지고 있는 애니메이션의 크기만큼 전부 뽑을 수 있다. 현재 사용하는 것은 idle, Run 이렇게 두 가지를 사용하고 있으며 이 두 가지에 대해서 먼저 블렌딩을 해보고 나머지 애니메이션을 넣어서 사용할 예정이다.    지금 우리가 사용하는 애니메이션 컨트롤러는 다음과 같이 동작한다.  우선 AnimationTrack을 통해서 기본적인 애니메이션의 정보를 관리한다.  기본적인 동작은 무한히 position(애니메이션 키)의 값을 시간X속도 만큼 증가시키고 position 값이 이 애니메이션의 길이보다 커지면 position의 값을 -0.00165f로 변환한다. 그리고 항상 position이 0보다 작아지면 position은 0이 된다.  이런 방식으로 루프를 돌며 각 키에 입력되어 있는 정보를 모델의 프레임에 적용시키고 그 정보를 토대로 메쉬의 정보를 변환한다.    현재는 이 애니메이션의 Move 함수를 동작시키면 1번 트랙의 애니메이션 Enable을 true로 설정하고 0번은 false로 설정해서 달리는 애니메이션을 키고 속도가 0이 되면 반대로 세팅해서 idle 애니메이션이 켜지게 된다.  일단 블렌딩을 다이렉트X로 처음하다 보니 처음에는 어리숙한 작업을 계속했다.  그 중 처음 한 작업은 idle 애니메이션은 오른발이 뒤에 있고 Run애니메이션은 왼발이 먼저 앞으로 나간다. 그래서 이 부분을 수정하고자 플레이어가 idle 동작을 하고 있으면 Run 애니메이션의 키프레임 부분을 오른발이 앞으로 가는 부분에 맞춰 보았다. 하지만 두 애니메이션은 다른 애니메이션이라 그렇게 해도 애니메이션이 튕기는 느낌을 받았다.  두 번째 작업은 애니메이션의 속도 조절이다. 이 부분은 나중에 필요해 보였다. 우선 달리는 애니메이션의 초반 부분의 속도를 우리가 사용할 가속값과 같게 해야하므로 블렌딩 작업이 끝난 후에 만지는 것이 좋아보인다.  <목>  본격적으로 애니메이션 블렌딩 작업에 들어갔다. 현재까지 애니메이션 블렌딩 작업은 블렌딩이 아닌 애니메이션의 기본 설정값들을 바꿔 주는 방법이었다.  하지만 다시 공부해 보니 블렌딩이란 두 개의 애니메이션에서 원하는 키프레임 부분의 뼈가 가지고 있는 값들을 보간을 통해 중간값을 찾아 다른 애니메이션을 작업하는 것이다. 그리고 보간하는 과정에서 어떤 애니메이션의 비중을 크게 할 지 설정하며 자연스러운 동작을 구현해 내는 것이 관건이었다. 아직 실제 수정작업은 들어가지 않았으며 이론과 영상을 통해 공부를 했었다.  금, 토  학생회 LT | | |
| 특이사항 | [김정훈 - 서버]   * 컴퓨터 오류로 인한 작업 시간 저하   -> 컴퓨터가 너무 자주 꺼져서 사람 불렀다가 컴퓨터 열어서 그래픽 카드 PCIe 뽑았다가 끼니 안정됨.(1/24)   * 다음 주에 1.5배 더 할 것… | | |
| 다음 주 수행 계획 | [김정훈 - 서버]   * 이동 회전 동기화 작업 계속.   [김진선 - 클라이언트]   * 충돌처리 및 바운딩 박스 확인 기반 작업   [이상민 - 클라이언트]   * idle, Run 프레임값 보간을 통한 블렌딩 작업 | | |