| 주차/일자 | 2주차 / 12.31~1.6 | 작성자 | 김정훈 |
| --- | --- | --- | --- |
| 개요 | 초기 계획 및 정기 보고 | | |
| 회의 내용 | 2024. 01. 05일 주간회의   * git repository 생성 (초기 프로젝트 코드 완성해 오기)   + <https://github.com/Olsangmin/CyberZ> * 플레이어 에셋 구매   + 완료(상민) [146,000원]   + 드라이브 리소스 폴더에 저장 * 서버 연결은 캐릭터의 애니메이션 블렌딩 완료 후 연결 * 계획 수정   + 오브젝트 배치  3~10주차 -> 8~10주차   + 방학 기간 중에는 캐릭터 에셋으로 동작하는 것들  위주로 진행 * 클라이언트에 캐릭터 2개이상 띄워주기 * 클라이언트 변경내용 merge (conflict X) | | |
| [다음 주 회의 안건] (2024.01.12)   * 애니메이션 fbx sdk 사용 관련 * 개인별 진척도 확인 * 문제점 파악 및 피드백 | | |
| 주간  수행 내용 | [김정훈 - 서버]  i) 서버 코드 빈껍데기로 초기화(클라이언트 연결X)   * 기존에 있던 플레이어(Session)의 좌표, 시야등을 모두 제거하고 ID만 남겨놓고 점차 추가할 예정.   ii) 하나의 파일로 작성됐었던 텀프로젝트 코드를 역할에 따라 분리   * 우선 Object단위만 수행. * 앞으로 분리해야 될 class   + network   + thread   + 플레이어 관리(오브젝트)   + NPC   + 맵 | | |
| [김진선 - 클라이언트]   * 다렉 개발 기반 프로젝트 제작   + 터레인 : 평지+한쪽만 솟아오름 (솟아오른곳은 지형 충돌 체크용)   + 적 class 기반 제작   + 기반 프로젝트에서 졸업작품에 사용하지 않을 class, 리소스 삭제 * 플레이어   + 생성자 수정     - 플레이어 마다 다른 모델을 사용하도록 수정     - 플레이어 객체를 선언할 때 사용할 모델의 정보를 받아와 사용하도록 구현   + 에셋 모델 수정     - 게임의 특성상 플레이어가 적을 공격하지 못하므로 플레이어 모델의 무기류 삭제       수정전(좌), 수정후(우)   * + 구매한 에셋 적용 (60% 진행) | | |
| [이상민 - 클라이언트]   * 자료구조와 프레임워크 구조 수정   + 이 부분에 대해서는 조금 더 공부가 필요해 보인다. 디스크립터 힙과 디바이스 등 모든 상태에 대해 건드려야 하기 때문에 이 부분은 잠시 나중으로 미루기로 판단하고 지금 할 수 있는 것을 하려고 한다. * 오브젝트 풀에 대한 공부   + New로 메모리를 여러 번 할당받는 과정에서는 컴파일러가 임의로 FreeStore에 공간을 제공해 주기 때문에 메모리 단편화 문제가 발생할 수도 있고 계속해서 System Call을 불러와 CPU의 동작이 많아진다.   + 이러한 문제로 인해서 오브젝트풀을 사용할 예정이다. 공부해본 바로는 메모리풀이 오브젝트보다 더 넓은 범위이다. 하지만 그렇게 구현을 할 경우에는 쓸데없는 동작도 많아지고 한 번에 많은 데이터를 할당받을 때에는 오히려 풀링 방식이 비효율적이므로 게임에 사용되는 객체를 호출할 때에만 오브젝트풀을 구현해서 사용할 예정이다.   + 참고자료 :<https://jartlife.tistory.com/59> & 졸업작품 실험용 Project(소스.cpp)   + 모든 참고자료는 직접 실험해 가면서 사용했다. 졸업작품 실험용 Project에서 소스파일을 계속 저장해 나가며 과정을 모두 기록해둘 예정이다. * 깃 허브 레파지토리 생성   + 우선 작업환경을 만들기 위해서 클라이언트의 기본적인 이름이나 파일 형식을 바꿔서 깃 레파지토리에 기록하고 그에 대한 개인적인 작업을 할 수 있도록 개인 브랜치를 팠다. 거기서 기본적인 틀 및 김진선(클라이언트)와 소통하며 필요한 내용을 역할을 나눠서 계속 작업했다. * 플레이어 추가 생성      * 기존에는 \*형태로 관리했지만 \*\*형태로 바꿔 주어서 다수의 플레이어를 관리할 수 있도록 변경했다. GameFramework 안에서의 내용을 수정하고 추가적으로 서버에 연동 시 플레이어 마다 고유아이디가 주어질 수 있도록      * stdafx.h에 define으로 선언해두었다. 또한 카메라는 FIRST\_PLAYER만 따라다니게 해두었다. * 추가적으로 두 플레이어 모두 렌더 가능하게 했으며 두 플레이어는 다른 조작키로 이동할 수 있도록 수정했다. * 이 후의 모델 로드 과정은 김진선(클라이언트)이 하는 것으로 조율을 했다. * 3D게임프로그래밍 복습      * 프레임워크 수정 과정에서 이 구조가 굉장히 중요했다는 기억이 있어서 이 내용들을 계속 외우며 공부했다.      * 명령 큐와리스트의 Flags 상태 설정 중 하나인데 아직 어떤 상황에서 써야하는지 구분이 안가서 따로 찾아보았다. 하지만 자료가 많지 않았으며 중요하지 않다고 판단해서 상태에 대해서는 굳이 건들지 않기로 결정했다. | | |
| 특이사항 | [김진선 - 클라이언트]   * 몇몇 플레이어 모델에 문제가 발생하여 확인 필요 * 타 모델로 진행시 동일 문제가 발생하지 않은 것으로 보아 유니티로 추출하는 과정에서 생긴 문제로 보임 | | |
| 다음 주 수행 계획 | [김정훈 - 서버]   * Shared\_ptr 공부 전 C++ 상속에 대해 자세하고 깊은 복습 및 정리 (protected, virtual, override 등 다형성을 갖춘 오브젝트 구현 목적) * 서버 프레임 워크 구상 및 정리 * ~게임서버 텀프로젝트 분석   [김진선 - 클라이언트]   * fbx sdk 공부 * 플레이어 모델 적용 완료   [이상민 - 클라이언트]   * fbx sdk 공부 * 기본적인 프레임워크 공부 | | |