**임시 기획서**

2018180009 김시인

2018180010 김연규

2018182034 전수민

게임 제목 : avoid\* him

게임 장르 : 멀티 서바이벌

스토리 – 평화롭지만은 않은 어느 날.

교수님들이 너무도 공부를 안하는 학생들을 공부시키기 위해 E동을 봉쇄하였다.

학생들은 교수의 공부지도를 피하여 E동을 탈출하기 위해 암호를 해독하여 탈출해야 한다.

개발 플랫폼: DirectX 12(클라이언트), IOCP(서버)

게임 플레이 :

1. 접속 화면 : 서버의 IP주소를 입력하는 화면이 제일 면저 나오고 IP주소를 입력받아 서버와 연결하고 로비화면으로 넘어간다.

2. 로비 화면 :

유저는 방을 만들거나 만들어진 방에 접속할 수 있다.

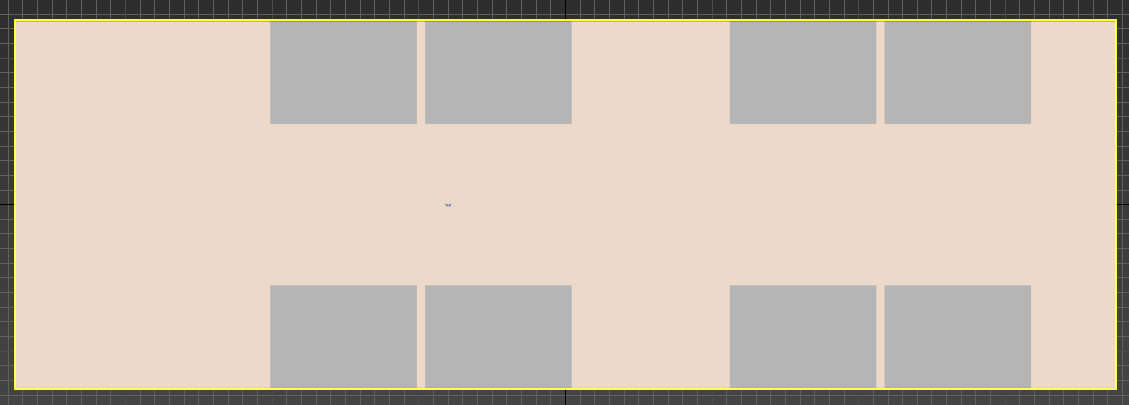
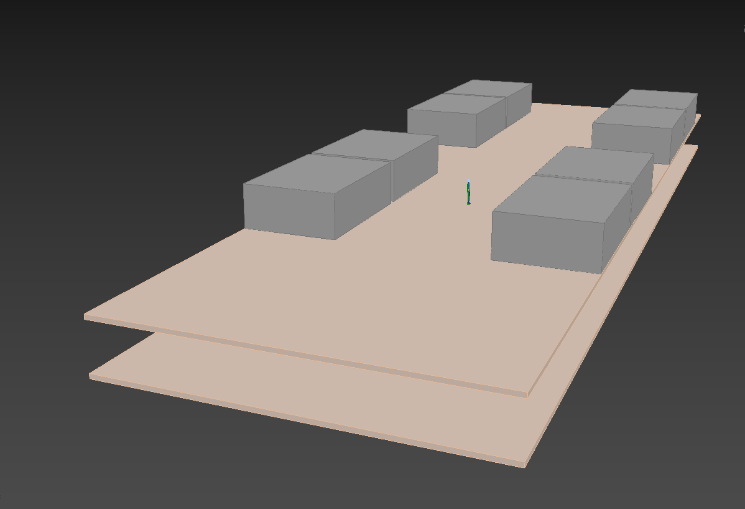
3. 대기 화면 :

유저는 최대 5명 접속할 수 있고 최소 3명의 유저가 접속해 있을 때 방장이 게임시작버튼을 누르면 게임이 시작된다.

4. 게임 플레이 화면 :

(사진 넣을곳)

처음 게임이 시작되면 한 명의 교수 플레이어가 랜덤으로 정해지고 나머지2~4명의 유저는 학생 플레이어가 되어 게임이 진행된다.



↑ 대략적인 맵 크기 (75mx25mx3m).

맵의 구조는, 스토리에 맞게 공학관 E동의 2층, 3층과, 2층의 내려가는 계단을 통해 내려가면 외부 지형으로 되어있는 마당으로 되어있다. 마당에는 포탈이 하나 설치되어있다.

2층에는 라운지가 있고, 3층에서는 2층의 라운지로 떨어질수 있는 구조이다. 각 층 마다 여러개의 방이 있는데, 방의 종류로는 강의실, 컴퓨터가 있는 실습실, 라운지, 화장실이 있고, 각 층에는 다른 층으로 이동할 수 있는 계단이 있다.

E동 1층을 만들지 않는 이유는, 모델링 구현 난이도에 비해서 사용빈도가 매우 낮을것 같다고 생각해서이다.

교수 플레이어는 모든 학생 플레이어들을 잡으면 (공부시키면) 승리하고, 학생 플레이어 들은 시간내에 그들 중 한명이라도 공학관에서 탈출할 시 승리하고, 탈출한 플레이어 수만큼 점수를 받는다.

교수는 탈출구 바로앞 마당에서 시작하고, 학생들은 3층 엘리베이터 앞에서 시작한다.

모든 실습실과 강의실에는 노트북과 컴퓨터가 겸비되어 있고, 랜덤으로 해독을 할 수 있는 컴퓨터가 정해진다. 해독 가능한 컴퓨터는 학생 플레이어 수만큼 있고, 해독하기 위해서 상호작용 키를 일정시간 누르고 있으면 해독이 된다. 이 컴퓨터들을 모두 해독시 마당에 있는 포탈이 활성화 되고, 이를 통해 밖으로 나갈수 있다.

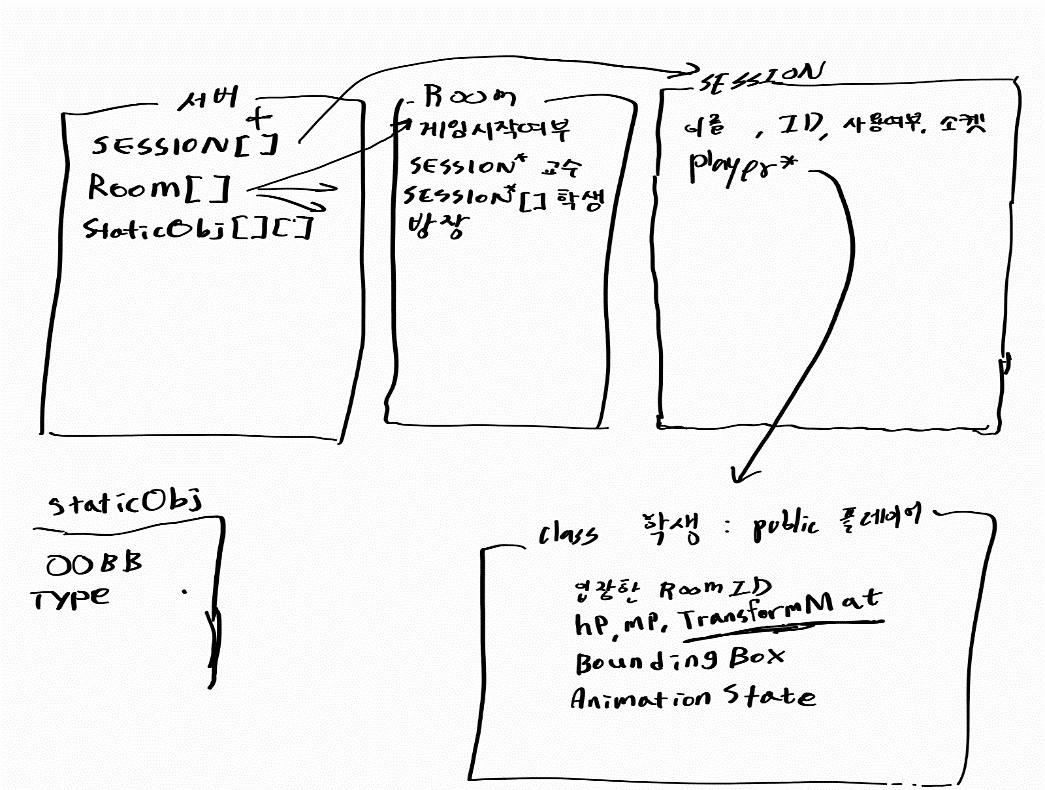
학생 플레이어와 교수 플레이어 각각 고유의 스킬을 보유하고 있고, 학생은 스태미너를 소모하여 대시를 사용할수 있다. 정수기에서 스태미너를 회복할수 있다. 교수는 기본적으로 학생보다 이동속도가 빠르고, 학생을 향해 공격 할 수 있고, 공격 시에 잠시 멈춘다.

학생이 체력이 전부 소모 되었을 경우, 3층 내에 있는 감옥에 갇히게 된다. 모든 학생이 감옥에 갇히면 교수의 승리이고, 제한시간이 모두 지나거나 한명의 학생이라도 탈출에 성공한 경우 학생의 승리가 된다.

5. 게임 종료 이후 화면

게임의 승패가 정해지면, 승패에 따라 플레이어들에게 승리, 패배 2D이미지가 출력되고, 몇 초 후에 다시 대기 화면으로 넘어간다.

구현(서버 관점) :



서버는 클라이언트의 정보를 담는 Session 컨테이너, 대기 화면상태의 방과 게임 플레이때 충돌체크 등에 필요한 정보들을 전부 가지고 있는 Room 컨테이너, 각 방에서 공통적으로 read되지만 write는 되지 않는, 즉 상태가 변하지 않는 ex) 건물 벽, 컴퓨터, 정수기 등... 의 오브젝트들의 정보인 StaticObject 컨테이너로 되어있다. 이 StaticObject 컨테이너는 벡터의 배열로 되어 있고, 각 Room은 전역에 있는 이 배열에 인덱스로 원하는 타입의 오브젝트를 접근 할 수 있게 할것이다. 세션은 이름, 세션id, 세션의 사용여부, 소켓 정보를 가지고, Player\* 로 플레이어의 정보를 가지고 있다. 플레이어 클래스 내에는 충돌이나 이동, 기타 게임 플레이에 필요한 hp,mp, 변환행렬, OOBB, 애니메이션 상태 등을 갖고, 이를 상속받는 학생, 교수 클래스에서 부가적으로 필요한 정보들이 들어있다.

로비 화면 - 클라이언트가 로비씬에 있을때 서버는 1초 마다 최신의 방에 대한 정보를 클라이언트들에게 전달하도록 구현한다. 만약 유저가 입장을 시도했는데 실패했을 경우(인원이 차서) 즉시 방에 대한 최신의 정보를 보내준다.

대기 화면 - 유저가 방에 접속하거나 나갈때 마다 방에 남아있는 모든 유저에게 그 정보가 담긴 패킷을 전달한다.

리소스 로드 - 게임이 처음 실행될때(응용프로그램이 실행될때) 서버와 클라이언트는 각자 필요한 모든 리소스를 미리 불러온다. -> 매판 마다 리소스를 로드하는 경우 가장 늦게 까지 로드가 끝나는 클라이언트 기준으로 게임이 시작되기 때문에 시간적인 측면에서 이점이 있다고 생각했다. + 맵이 여러개 일경우 게임이 끝나면 플레이 했던 맵의 대한 정보가 필요 없어지므로 해제를 하겠지만, 현재 만드려 하는 게임은 한개의 맵을 계속 사용하기 때문에 프로그램을 종료하는 것이 아니라면 리소스를 해제할 일이 없다.

구현(클라이언트 관점) : 클라이언트에서 충돌처리를 하는것이 아닌, 서버에서 클라이언트들이 누른 키를 받아 충돌 처리를 하므로, 오브젝트마다 바운딩 박스를 가지지 않는다. 클라이언트는 서버와 별도로 맵에 대한 클래스를 갖고 있고, 서버에서 받은 패킷을 기반으로 클라이언트 내의 오브젝트들의 변환만 진행해서 렌더링한다. fbx파일이나 맵의 데이터를 추출할때는 일단 유니티를 사용하기로 하였다. 유니티를 사용한다면 오브젝트 정보를 불러오는 것 뿐만 아니라 별도의 에디터를 구현하지 않아도 구현 했을 때 보다 맵 배치에 있어서 더 좋은 결과물을 얻을 수 있기 때문에 유니티도 같이 공부하여 별도의 FileMaker 유니티 프로젝트를 만들어 로드하는 기능의 프로토타입까지 만들어 본 상태이다.

구현(모델러 관점):

캐릭터를 하나만 써서 적 플레이어 캐릭터와 러너 플레이어 캐릭터의 텍스쳐와 노말맵을 다르게 써서 한 캐릭터 모델 하나로 다른게 보이며 다른 캐릭터를 만들어 낼 수 있다. 러너 캐릭터 3개는 외형의 차이가 없고, 텍스쳐 색깔의 변화만 있다.

한 맵 안에 여러 물체가 구역마다 여러 개씩 놓이며, 해당 모델링들은 각각이 텍스쳐와 노말맵을 가진다. 쓸 필수 모델링의 경우 다음과 같다.

----------------------------------------------------------------------

계단, 계단 손잡이(난간), 움직이는 여닫이문, 천장등, 정수기, 의자, 아이템 박스, 긴 책상, 일반 사각 책상, 전등, 컴퓨터, 노트북, 긴 책상, 의자, 철창, 소변기, 대변기 문, 세면대, 거울, 가방, 캔, 시계, 쓰레기통, 화분, 스크린, 책장, 택배박스, 라디오, 서류, 막대기

맵 모델링에 붙어있는 물체는 텍스쳐를 다르게 써서 마치 모델링이 잘 된 것처럼 보이게 한다. 해당 물체는 다음과 같다.

엘리베이터

상호작용이 가능하여 텍스쳐가 갈아 끼워지는 물체는 다음과 같다.

컴퓨터, 노트북

정지 모션: 상호작용 같은 경우 정지되어 작업중임을 표시하는 모션을 만든다. 만들 모션은 다음과 같다.

컴퓨터(노트북)조작중인 모션,

애니메이션: 캐릭터 애니메이션을 만들 것이며, 만들 애니메이션은 다음과 같다.캐릭터 공통: 기본 idle , 걷기, 뛰기,

적 캐릭터: 곤봉 휘두르기,

오디오:

BGM: 메인화면, 게임 시작후 게임 진행중, 게임 엔드 및 점수창,

효과음: 컴퓨터(노트북) 해독음, 적 캐릭터 공격음, 러너 캐릭터 타격당한소리,

메인화면: 버튼 클릭 소리.