

Professor.정내훈

REVENGER

2018182009 김승환
2018180046 허재성
2018180033 이세철

목차

1 연구 목적

2 게임 소개

- 1) 배경
- 2) 게임 정보
- 3) 조작 키
- 4) 게임 방법

3 유사 게임

4 개발 환경

5 중점 연구 분야

- 1) 서버 이중화
- 2) 레이 트레이싱

6 역할 분담 및 준비 현황

7 개발 일정

8 출처

Part 1

연구 목적

- Direct3D 12를 이용한 3D 게임 제작 경험
- 서버-클라이언트 프로그래머 간의 협업 프로젝트 진행 경험
- 버전 컨트롤 시스템 Git을 통해 프로젝트 관리 능력 증진
- 서버와 클라이언트의 협동 작업을 위한 프레임워크 제작
- IOCP를 사용하여 서버 개발에 있어 효율적인 제작 능력 증진

Part 2

게임소개

2

게임 소개

Introduction to the Game

1) 배경



<그림 1> 비행 화면 예시

공중 전투 게임과 지상 전투 게임을 즐길 수 있는 게임

팀원이라는 다른 플레이어와 함께
제한 시간 안에 적 NPC를 몰살하고
점령해야 하는 멀티 게임

2) 게임 정보

<맵>

- 1스테이지 당 1개의 맵 (스테이지 2개로 제한)
- 10km * 10km의 맵 크기

<UI/UX>

- 크로스 헤어(조준 점), 고도, 속도
- 플레이어 HP, 기체 파괴 현황
- 미니맵, 레이더
- 총 제한 시간, 점령 중인 시간
- 임무 진행에 따른 진행률
- 탄 개수, 특수 기능
- 적 HP, 점령 지역, 적 본진 지역

<오브젝트 종류>

- 헬기
 1. HP이 높고 방어력이 좋은 거대한 헬기
 2. 이동, 공격 연사 속도가 빠른 왜소한 헬기
 3. 공격력이 높은 평범한 크기의 헬기
- 기체 별 사용되는 총알 및 미사일
- 지상전에서 사용되는 총알, 수류탄
- 전투 시설(사격 벙커, 대공포)
- 지형에 따른 시설(전투 시설, 폐건물 등) 및 환경 오브젝트
- 약 500개의 오브젝트들이 존재

3) 조작 키



<그림 2> 사용하는 키보드, 마우스 표시

(공중) W/S: 기체 상승/하강
 (공중) A/D: 기체 회전
 (지상) W/A/S/D: 사람 이동
 R: 기본 공격 장전
 F2/F3: 헬기 카메라 전환

CRTL: 헬기, 사람의 특수 능력 사용
 마우스 좌측 클릭: 기본 공격
 마우스: 기체의 각도 회전 및 사람
 카메라 회전

4) 게임 방법



<그림 3> 게임 화면 예시

- 적 발견 시 적의 위치를 UI로 보여준다.
- 점령 지역은 위 사진의 '공격' 표시 처럼 위치를 알려준다.
- 스테이지는 총 2개이며 1 스테이지를 깨면 이어서 2 스테이지로 넘어간다.
- 1스테이지는 헬기를 사용하는 공중전, 2 스테이지는 실제 사람으로 전투 시설을 부숴야 하는 지상전으로 이루어져 있다.

4) 게임 방법



<그림 4> 무력 시스템 예시

- 플레이어는 헬기의 각 고유 특수 능력을 사용할 수 있다.
- 헬기가 공격을 받거나 장애물에 충돌할 경우 헬기의 어느 부위가 손상되었는 지 표시한다.
- 헬기의 HP가 0이 되거나 특정 부위가 파괴가 되면 기능 정지가 되고 플레이어는 10초 뒤 정해진 지역에서 부활한다.
- 적들은 부활하지 않으므로 모든 적을 처리하면 승리하게 된다.

Part 3

유사게임



<그림 5> Battle Field 4, 적 헬기에게 공격 시 화면 인 게임



<그림 6> World of Warplanes, 플레이어 특수 능력 - 적 지역에 폭탄 떨어트리는 화면 인 게임

게임 이름	유사점	차이점
Battle Field 4	<ul style="list-style-type: none"> 지상전, 공중전의 플레이 적과의 전투로 자신은 살아남아 상대를 처치하는 방식 	<ul style="list-style-type: none"> 적을 처치하면서 목표 지역을 점령 스테이지 형식으로 공중전 클리어 시 지상전으로 돌입 Player vs Player의 PVP가 아닌 Player들과 팀을 맺어 적 NPC들을 처치하는 방식 처치와 점령으로 인한 점수제가 없는 형태
War of Warplanes	<ul style="list-style-type: none"> 전투기 간의 유도 미사일 같은 특수 능력 전투기 모델을 선택하여 플레이 	

Part 4

개발환경

- Microsoft Windows 10
- DirectX 12
- Visual Studio 2022
- Unity 3D
- LUA Script
- 3DS MAX
- FMOD 2.02
- IOCP

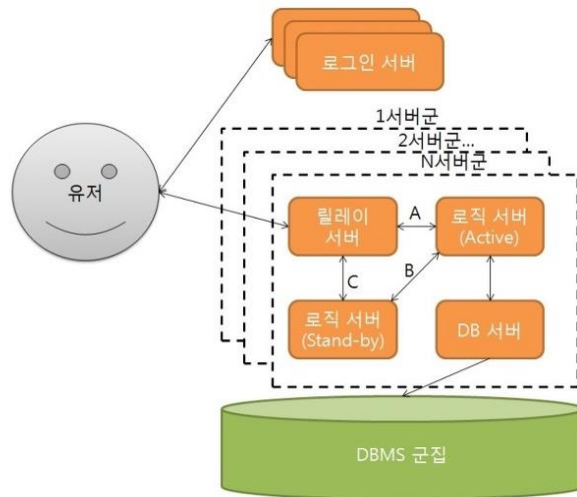
Part 5

중점연구분야

1) 서버 이중화

서버 이중화

서버 이중화를 통한
HA(High Availability,
고가용성)를 구현한다.



<그림 7> 서버 이중화 도식화 예시

운영중인 서비스의
안정성을 위해 각종
자원을 이중 또는 그
이상으로 구성한다.

하나의 서비스에 장애가 발생
하는 경우 다른
서버를 통해 서비스를
지속가능하게 한다.

결과적으로 서버 하나가
다운되어도 다른 서버로
연결되도록 하여 약간의
렉만 발생할 뿐,
사용자가 이를 인지하지
못하도록 한다.

기존 서버를 릴레이
서버와 로직 서버로
분리한다.

(릴레이 서버:
Connection Pool 역할
로직 서버:
실제 게임 로직 관리)

로직 서버는 Active-Standby
구조로 이중화한다.

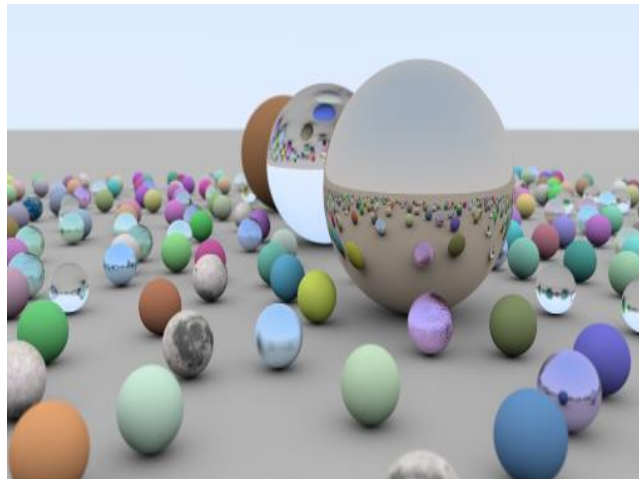
Active 서버와 Standby
서버는 Heartbeat를 주고 받
으며 서로 상태를
확인한다.

서버간의 데이터를
동기화한다.

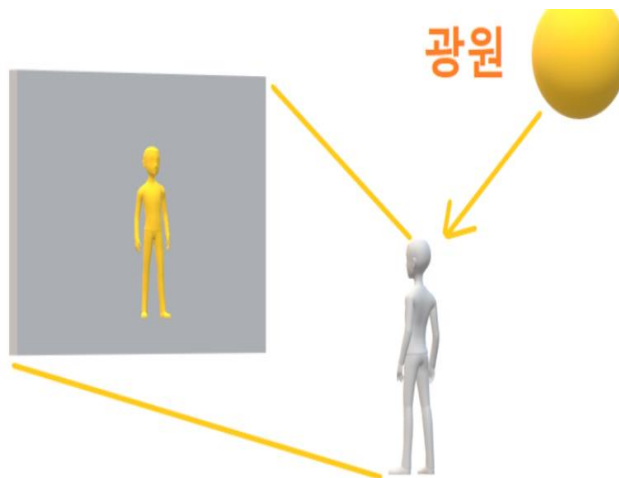
2) 레이 트레이싱

레이 트레이싱

실제와 근사한 모습의
그래픽을 구현한다.



<그림 8> 레이 트레이싱이 구에 적용된 모습



<그림 9> 레이 트레이싱의 원리 예시

물체의 색상은 반사된 빛이
우리 눈에 비친 결과물로,
반사된 빛은 눈으로
바로 들어오기도 하지만
다른 물체들을 통해
굴절이 되거나 반사가 된 후
눈에 들어오기도 한다.

왼쪽 상단의 사진은
레이 트레이싱 기법이
적용된 구들의 사진이며
레이 트레이싱을 적용하면
반사와 굴절에 대한 표현이
래스터라이제이션 보다
더 현실적인 그래픽을
보여준다.

빛을 추적하기 위한
카메라에서 발사된 광선을
1차 광선이라 할 때,
1차 광선이 어떤 물체에 닿지
않을 경우 간접광은 없음으로
판단한다.

그러나 물체에 닿을 경우
그림자에 가려지는 지,
반사와 굴절이 되는 지를
판단하고,

각 상황에 대해서 광선들을
계산하여 추가적으로 생성된
그림자 광선, 반사 광선,
굴절 광선들로 물체에
부딪힘 없이 광원에
도달할 때까지 재귀적으로
반복한다.

Part 6

역할 분담 및 준비 현황

<허재성>

- 자료구조
- 게임 수학
- STL
- 네트워크 게임 프로그래밍
- 3D 게임 프로그래밍 1,2

<김승환>

- C, C++ 프로그래밍
- 알고리즘
- STL
- 네트워크 게임 프로그래밍
- 데이터 베이스
- 인공지능
- 게임 서버 프로그래밍

<이세철>

- C, C++ 프로그래밍
- 알고리즘
- STL
- 스크립트 언어
- 네트워크 게임 프로그래밍
- 게임 기획 1, 2

허재성 (클라이언트)	김승환 (서버)	이세철 (기획/서버)
<ul style="list-style-type: none"> 조명, 그림자 처리 노멀 매핑 및 텍스처 블렌딩 모델 링킹 빌보드 처리 애니메이션 적용 레이 트레이싱 	<ul style="list-style-type: none"> 게임 서버 프레임워크 제작 로그인 서버 프레임워크 제작 각 서버와 클라이언트 간의 송수신 서버의 이중화를 위한 설계 	<ul style="list-style-type: none"> Lua Script (헬기 AI) Lua Script (전투 시설 AI) 물리적 움직임에 대한 로직 플레이어 자원 관리 (특수 능력, 파괴 수치 등)

Part 7

개발 일정

항목	1월	2월	3월	4월	5월	6월	7월	8월
리소스 수집								
클라이언트 프레임워크								
애니메이션								
서버 프레임 워크								
서버-클라이언트 통신								
UI, 충돌처리, 모델 링킹, 사운드								
NPC 인공지능								
레이 트레이싱								
서버 이중화								
자원 관리								
테스트 및 수정								

Part 8

출처

페이지 06 : 그림 1_ 비행 화면 예시
https://youtu.be/l2UkzfwD_S8

페이지 08: 그림 2_ 사용하는 키보드, 마우스 표시
<https://dpg.danawa.com/bbs/view?boardSeq=244&listSeq=4044271&past=Y>

페이지 09: 그림 3_ 게임 화면 예시
<https://www.youtube.com/watch?v=loI7OWbRs2c&t=347s>

페이지 10 : 그림 4_ 무력 시스템 예시
<https://from2015.tistory.com/1025>

페이지 12: 그림 5_ Battle Field 4, 적 헬기에게 공격 시 화면 인 게임
<https://www.youtube.com/watch?v=8R1XFU8ecEM>

페이지 12: 그림 6_ World of Warplanes, 플레이어 특수 능력 - 적 지역에 폭탄 떨어트리는 화면 인 게임
<https://www.youtube.com/watch?v=Cj8kp11kQUA>

페이지 16 : 그림 7_ 서버 이중화 도식 예시
<https://creazier.tistory.com/m/15309858>

페이지 17 : 그림 8_ 레이 트레이싱이 구에 적용된 모습
<https://blog.hybrid3d.dev/2019-11-15-raytracing-pathtracing-denoising>

페이지 17: 그림 9_ 레이 트레이싱의 원리 예시
<https://jntechreview.tistory.com/51>

THANK YOU