게임서버 프로그래밍 텀 프로젝트 기획서

학과 : 게임공학부 엔터테인먼트 컴퓨팅전공

학번 : 2017184004

이름 : 김기윤

목차

[게임서버 프로그래밍 텀 프로젝트 기획서 1](#_Toc89978845)

[1. 개발 환경 3](#_Toc89978846)

[2. 게임 구현 - (추가 요소 빨간색으로 표현) 3](#_Toc89978847)

[3. 프로토콜 6](#_Toc89978848)

[4. 자료구조 10](#_Toc89978849)

[5. 게임흐름 14](#_Toc89978850)

# 1. 개발 환경

* OS : Windows 10
* IDE : Visual Studio 2019
* 네트워크 모델 : 멀티쓰레드 IOCP
* 언어 : C/C++
* 개발 하드웨어 성능 : 코어 8개, 램 16GB

# 2. 게임 구현 - (추가 요소 빨간색으로 표현)

- 기술

* MMORPG 제작
* 멀티쓰레드 IOCP이용
* DB (MS SQL, SSMS 사용)
  + 사용자의 정보 저장(player id, player name, x좌표, y좌표, Hp, Level, exp, Max Hp)
  + ID를 가지고 로그인
  + 이미 로그인 되어있는 아이디로 로그인 시도시 로그인 실패
  + 없는 아이디로 로그인 시도시 자동으로 아이디가 생성, 생성 후 로그인됨
* Script
  + 몬스터 배치
    - ( 랜덤값으로 준 초기 x, y좌표를 저장하고 어그로 당한 후 원래자리로 돌아가기 위해 사용)
    - 계속 쫒아갈 것인지 아닌지 판단
    - 공격 범위 설정과 NPC가 공격시 공격의 유효성 판단
* 컨텐츠
* 플레이어 이동( 키보드 화살표로 이동 )
* NPC-AI(고정형 Peace몬스터, 고정형 Agro 몬스터)
  + 레벨, HP, 이름은 스크립트에서 가져옴
  + 종류별로 클라이언트에서 띄어주는 이미지가 다름
  + 고정형 Peace 몬스터 : 공격을 하면 몬스터가 공격을 하고 도망가도 따라옵니다

이름 : HomeWork

레벨 : 10, hp : 250

이미지 : 텍스트이(가) 표시된 사진

자동 생성된 설명

* + 고정형 Agro 몬스터 : 근처 11x11영역안에 들어오면 몬스터가 공격을 하고 따라옵니다

이름 : Project

레벨 : 20, hp : 900

이미지 : 텍스트이(가) 표시된 사진

자동 생성된 설명

* + 플레이어가 특정 범위 이상 벗어나면 1초씩 이동하며 자기자리로 돌아갑니다
  + 죽으면 30초 후에 맨 처음 x, y좌표에서 부활합니다
  + 이동은 1초마다 한 칸씩 가능합니다
  + 공격에 성공하면 5초후 다시 공격을 하고 공격에 실패를 하면 1초후에 다시 공격을 시도하도록 만들었습니다
* 맵 제작
  + 2000X2000
  + 20X20 Window(시야 15x15)
  + 장애물 존재
* 캐릭터
  + ID로 선택
  + 경험치와 레벨, HP존재
  + N레벨 요구 경험치 : 100\*(2^(n-1))
  + 5초마다 10%의 HP회복
  + HP 0이 되면 경험치 반으로 깍은 후 HP회복 돼서 특정 위치(500, 500)로 이동
  + 모든 정보는 DB에 저장
  + 기본공격(A키), 스킬(1번키, 2번키)존재
    - 기본 공격 : 주위 4방향 몬스터 공격 (1초에 한번씩 가능)
    - 1번 스킬 : 주위 4방향 뿐만 아니라 대각선을 포함해서 공격 (3초에 한번씩 가능)
    - 2번 스킬 : 이동하는 방향으로 5칸을 이동하고 그 사이에 있는 몬스터 공격(5초에 한번씩 가능)
* UI
  + 좌측 상단에 MAX HP, HP, MAX EXP, EXP, LEVEL 현재좌표 표시
  + 좌측 하단에 채팅 로그와 채팅할 수 있는 창을 띄워 줍니다
* 채팅 가능
  + 영어로 채팅이 가능(채팅을 한 플레이어의 이름과 함께 메시지 표현)
  + 해당되는 플레이어의 전투 메시지를 띄어줍니다

# 3. 프로토콜

* 기본 제공 프로토콜
  + 서버 -> 클라
    - 로그인 성공을 보내주는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

(name,tribe(HUMAN, MONSTER, BOSS, OBSTACLE))가 없어서 추가로 작성하였습니다

* + - 움직였을 때 보내주는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

(움직이는 것은 플레이어들과 몬스터들이지만 하나의 컨테이너로 관리하기 때문에 id로 구분을 합니다)

* + - 시야에 새로운 객체가 들어오면 보내주는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

(로그인 성공 패킷에 있었던 tribe대신 object\_type을 사용합니다 – 시야에 들어온 객체가 무엇인지 판별해줍니다)

* + - 시야에 존재하던 객체가 사라지면 보내주는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

(움직임과는 다르게 장애물은 별도로 관리하기 때문에 object\_type이 필요하여 넣어주었습니다)

* + - 메시지를 보내주는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

* + - 로그인 실패시 보내주는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

* + - 플레이어의 정보가 바뀌면 보내주는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

* + 클라 -> 서버
    - 클라이언트가 로그인을 하기 위해 서버에게 보내주는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

* + - 플레이어의 방향키를 눌러 움직임을 보내주는 패킷

텍스트, 시계이(가) 표시된 사진

자동 생성된 설명

* + - 플레이어가 공격을 눌러 공격했다는 것을 보내주는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

* + - 플레이어가 채팅을 하면 채팅의 문자를 보내주는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

* + - 스트레스 테스트를 위해 여러 좌표로 이동시키는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

* 개인 제작 프로토콜
  + 서버 -> 클라
    - 플레이어가 죽었다는 것을 보내주는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

* + - 플레이어 부활 시에 보내주는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

* + 클라 -> 서버
    - 플레이어가 스킬 사용시에 보내주는 패킷

텍스트이(가) 표시된 사진

자동 생성된 설명

# 4. 자료구조

* + 열거형
    - 플레이어와 몬스터 상태를 나타냄



ST\_FREE : 아무것도 아닌 상태

ST\_ACCEPT : 연결은 되어있으나 로그인은 아직 되어있지 않은 상태

ST\_INGAME : 게임에서 살아있는 상태 ( 몬스터의 경우 Default값)

ST\_DEAD : 게임 내에서 죽은 상태

* + - 종족



HUMAN : 플레이어

MONSTER : 고정형 Peace 몬스터

BOSS : 고정형 Agro 몬스터

OBSTACLE : 장애물

* + 클래스(서버)
    - NPC 클래스

텍스트이(가) 표시된 사진

자동 생성된 설명

* Name, id, hp, maxhp, level, exp(사용하지 않음), x좌표, y좌표, tribe(객체의 종류), state(살아있음, 죽음을 나타냄)이 기본적으로 게임을 하기 위한 변수들
* \_active의 변수의 경우 타겟이 있을 경우 true, 타겟이 없거나 사라진 경우(시야에 벗어남, 타겟이 죽음, 타겟이 로그아웃하는 경우) false
* \_active가 false가 되면 자기자리가 아닌 경우 자기자리로 되돌아가고 공격을 멈춤
* 루아 스크립트를 사용하기 위한 가상머신 변수가 존재
* 공유자원의 데이터 레이스를 줄이기 위해 state\_lock과 lua\_lock이 존재, 나머지 변수에 대해서는 오류보다 성능이 중요하기 때문에 lock을 사용 안함
* 루아를 동시에 사용하는 경우 에러 발생(스택이 뒤섞임), 사용하기 직전 lock를 사용함
  + - Player 클래스

텍스트이(가) 표시된 사진

자동 생성된 설명

* Npc클래스를 상속받음
* npc클래스에 있는 \_id는 컨테이너의 접근을 위한 숫자, \_login\_id는 DB에 접근을 하는 진짜 id(Npc는 접근할 이유가 없으며 플레이어만 가지고 있다)
* \_attack\_active와 \_skill\_active는 각각 공격과 스킬의 사용여부를 알려주는 bool변수이다
* Viewlist와 ob\_viewlist는 각각 플레이어의 시야에 들어와있는 플레이어/Npc, 장애물의 객체 id가 저장되어있는 컨테이너
* \_socket과 \_recv\_over, \_precv\_size는 네트워크에 사용되는 변수
* last\_move\_time 변수는 스트레스 테스트를 위해 사용하는 변수
* direction은 이동방향을 알려주고 스킬 사용시 사용됨
  + - Obstacle 클래스(장애물 클래스)

텍스트이(가) 표시된 사진

자동 생성된 설명

* x, y좌표를 가지고 있으며 tribe는 OBSTACLE로 설정
  + - EXP\_OVER클래스

텍스트이(가) 표시된 사진

자동 생성된 설명

* OVERLAPPED구조체의 확장
* \_comp\_op : 어떤 작동을 하는지 알려준다
* \_wsa\_buf, \_net\_buf : recv랑 send시에 사용
* \_target : 타겟이 필요한 경우가 존재하며 그때 값을 넣어줌
  + - Timer\_event 구조체

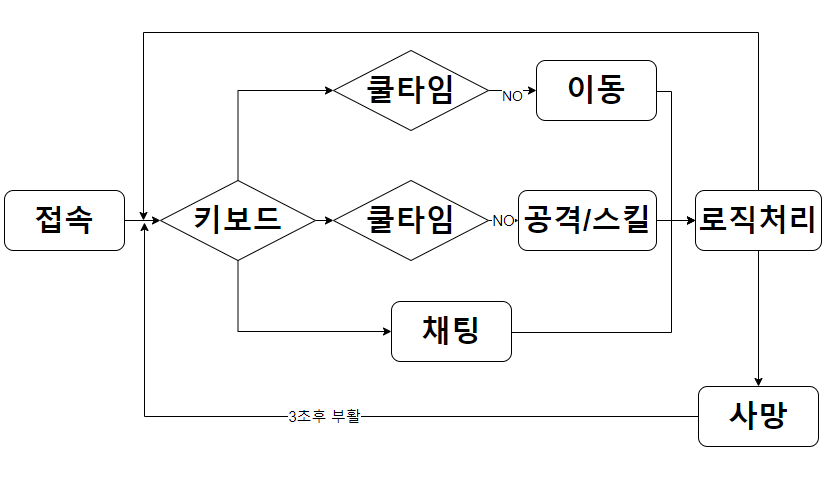
텍스트이(가) 표시된 사진

자동 생성된 설명

* 타이머가 있는 행동을 처리하기 위한 구조체

# 5. 게임흐름

* + 플레이어



* + Npc

