

네트워크 게임 프로그래밍

추진계획서

[문서 부제]

2021180001 강민서

2020184019 박지훈

2020182017 신태양

목차

애플리케이션 기획

하이 레벨 디자인

로우 레벨 디자인

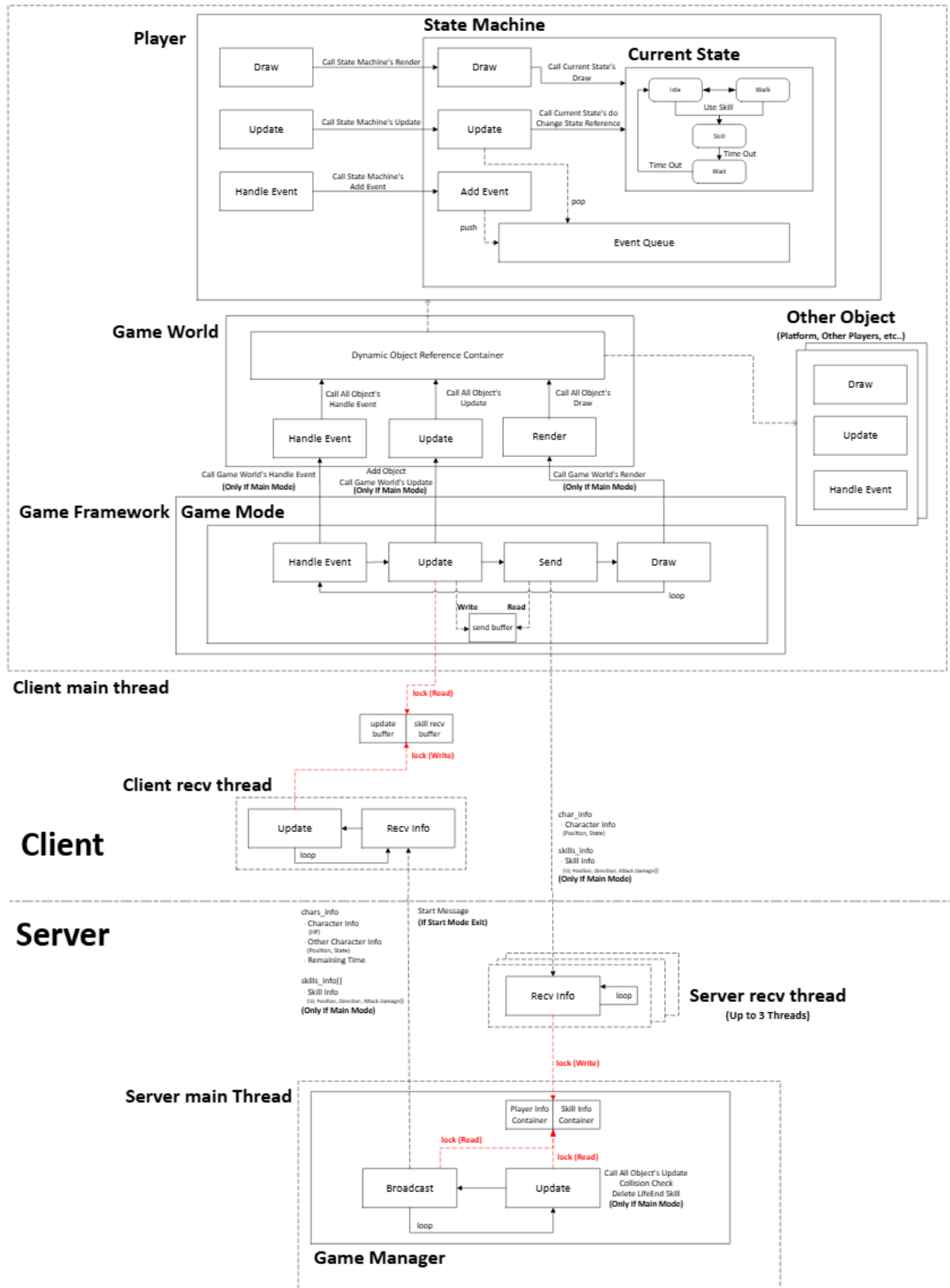
역할분담

개발 환경 및 일정

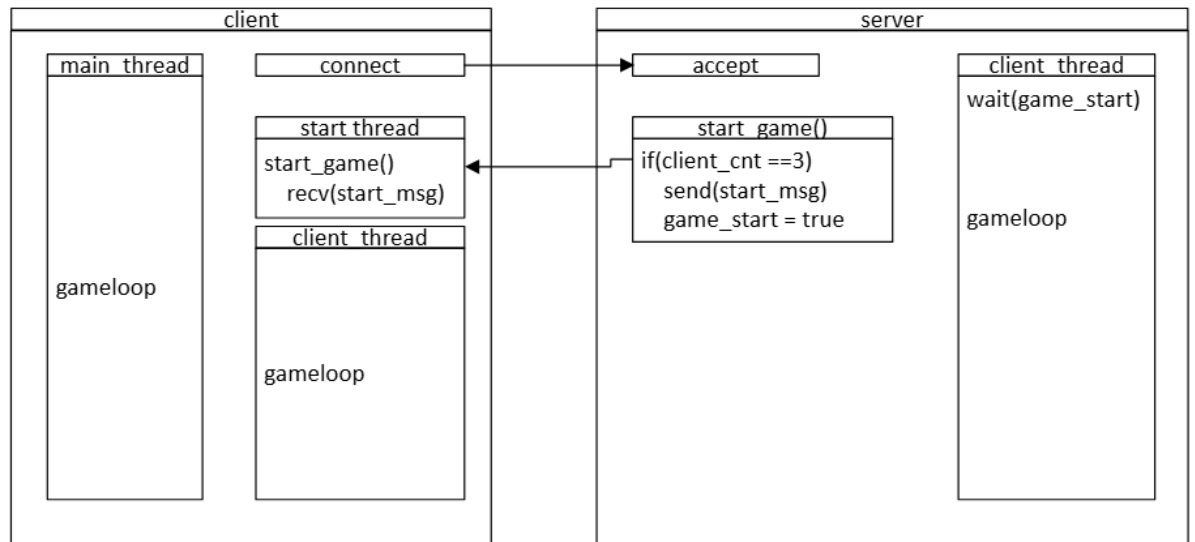
애플리케이션 기획

- 24년도 2D게임 프로그래밍 수업에서 python을 이용하여 강민서가 제작한 게임
- 기존 게임 방식
 - 사이드 뷰 로그라이트 액션게임
 - 매 스테이지 마다 보스를 처치 후 랜덤하게 뜨는 강화를 선택하여 진행하는 방식
- 변경된 게임 방식
 - 사이드뷰 pvp 액션게임
 - 플레이어 3명이 모였을 시 랜덤한 스테이지에서 1vs1vs1로 전투를 진행
 - 플레이어 사망 시 처치한 플레이어에게는 점수를 부여
 - 사망한 플레이어는 일정 시간 뒤에 부활
 - 주어진 시간동안 가장 많은 점수를 획득한 사람이 승리
- 통신 방식
 - TCP 프로토콜을 이용한 통신
 - 패킷 구조
 - ◆ 클라이언트 -> 서버
 - 캐릭터 위치 정보
 - 캐릭터 상태
 - 사용한 스킬이 있을 경우 스킬 정보
 - ◆ 서버 -> 클라이언트
 - 다른 캐릭터의 위치 정보
 - 다른 캐릭터의 상태
 - 내 캐릭터의 hp
 - 잔여 시간
 - 사용한 스킬이 있을 경우 스킬 정보 배열

하이레벨 디자인

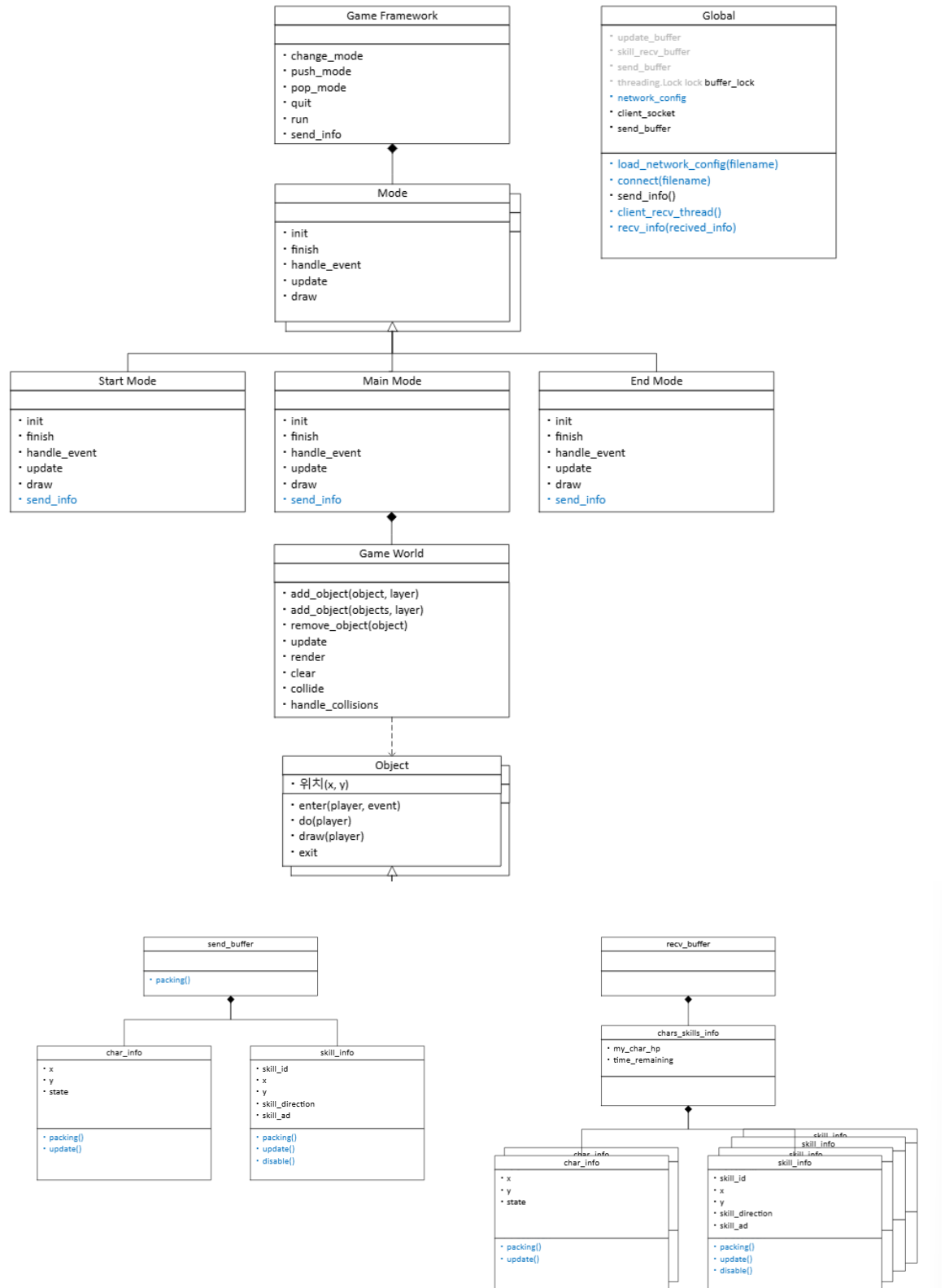


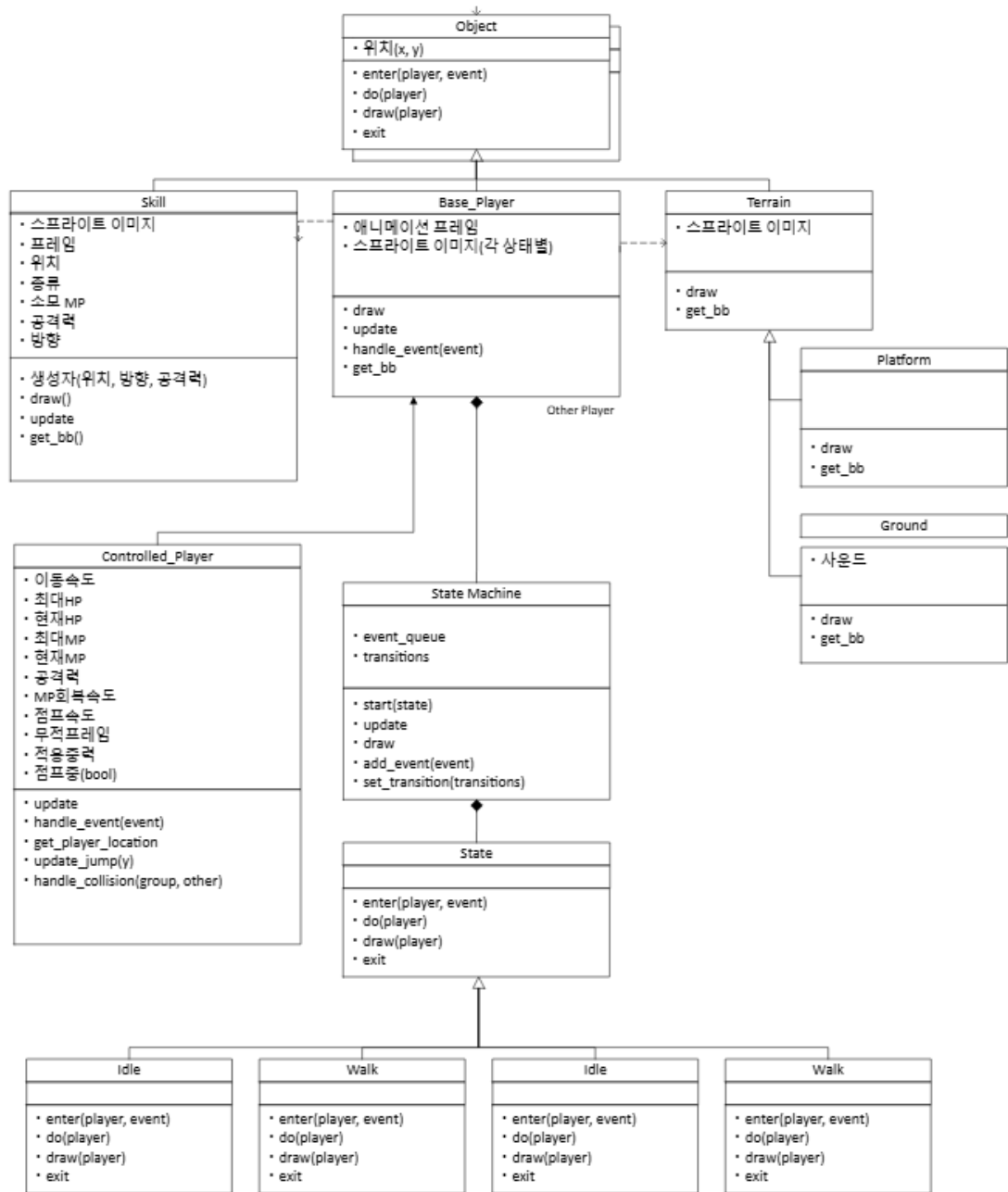
- 하이레벨: 게임 시작 대기



로우레벨 디자인

- 객체 설계





- 통신 설계

client_info
SOCKET sock
worldObjects* objects

player_info
SOCKET sock
int id

location
float x
float y

aabb
float max_x
float max_y
float min_x
float min_y

enemy
float x
float y
int frame
class state
void update()

client - main thread 추가내용
array<char_info> update_buffer
array<char_skill_info> skill_rcv_buffer
skill_info skill_send_buffer
threading.Lock lock
send_info()

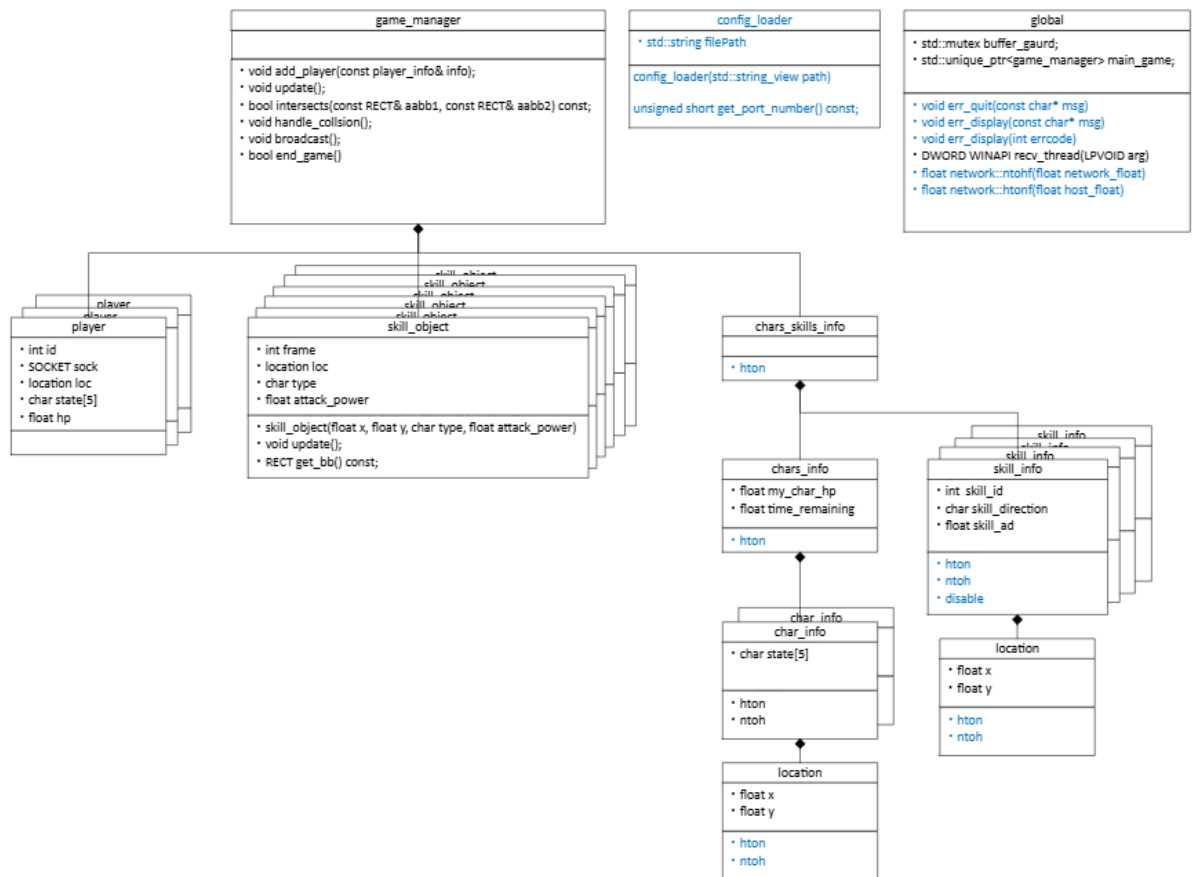
DWORD WINAPI client_sock_thread(client_info)
client_info info
int received_skills
int rcv_info(SOCKET)
void update_info(int)

DWORD WINAPI player_sock_thread(player_info)
player_info info
lock_gaurd<mutex> lock
void rcv_info(SOCKET)

char_skill_info								
<table> <tr> <th>skill_info</th></tr> <tr> <td>int skill_id</td></tr> <tr> <td>location loc</td></tr> <tr> <td>char skill_direction</td></tr> <tr> <td>float skill_ad</td></tr> </table> <table> <tr> <th>char_info</th></tr> <tr> <td>location loc</td></tr> <tr> <td>char[4] char_state</td></tr> </table>	skill_info	int skill_id	location loc	char skill_direction	float skill_ad	char_info	location loc	char[4] char_state
skill_info								
int skill_id								
location loc								
char skill_direction								
float skill_ad								
char_info								
location loc								
char[4] char_state								

chars_skills_info										
<table> <tr> <th>skills_info*</th></tr> <tr> <td>int skill_id</td></tr> <tr> <td>location loc</td></tr> <tr> <td>char skill_direction</td></tr> <tr> <td>float skill_ad</td></tr> </table> <table> <tr> <th>chars_info</th></tr> <tr> <td>float my_char_hp</td></tr> <tr> <td>location other_char_location*</td></tr> <tr> <td>char[4] other_char_state*</td></tr> <tr> <td>float time_remaining</td></tr> </table>	skills_info*	int skill_id	location loc	char skill_direction	float skill_ad	chars_info	float my_char_hp	location other_char_location*	char[4] other_char_state*	float time_remaining
skills_info*										
int skill_id										
location loc										
char skill_direction										
float skill_ad										
chars_info										
float my_char_hp										
location other_char_location*										
char[4] other_char_state*										
float time_remaining										

server																																
<table> <tr> <th>player</th></tr> <tr> <td>SOCKET sock</td></tr> <tr> <td>float x</td></tr> <tr> <td>float y</td></tr> <tr> <td>char[4] state</td></tr> <tr> <td>float hp</td></tr> </table> <table> <tr> <th>skill</th></tr> <tr> <td>int frame</td></tr> <tr> <td>float x</td></tr> <tr> <td>float y</td></tr> <tr> <td>char type</td></tr> <tr> <td>float attack_power</td></tr> <tr> <td>생성자(float, float, char, float)</td></tr> <tr> <td>void update()</td></tr> <tr> <td>aabb get_bb()</td></tr> </table> <table> <tr> <th>game_manager</th></tr> <tr> <td>2Darray<int, player> players</td></tr> <tr> <td>2Darray<int, skill> skills</td></tr> <tr> <td>void add_player(int, SOCKET)</td></tr> <tr> <td>void update()</td></tr> <tr> <td>bool intersects(aabb1, aabb2)</td></tr> <tr> <td>void handle_collision()</td></tr> <tr> <td>void broadcast()</td></tr> <tr> <td>bool end_game()</td></tr> <tr> <td>void start_game()</td></tr> </table> <table> <tr> <th>timer</th></tr> <tr> <td>float start_time</td></tr> <tr> <td>void start()</td></tr> <tr> <td>void stop()</td></tr> <tr> <td>void reset()</td></tr> <tr> <td>float get_time()</td></tr> </table> <table> <tr> <td>global mutex buffer_gaurd</td></tr> </table>	player	SOCKET sock	float x	float y	char[4] state	float hp	skill	int frame	float x	float y	char type	float attack_power	생성자(float, float, char, float)	void update()	aabb get_bb()	game_manager	2Darray<int, player> players	2Darray<int, skill> skills	void add_player(int, SOCKET)	void update()	bool intersects(aabb1, aabb2)	void handle_collision()	void broadcast()	bool end_game()	void start_game()	timer	float start_time	void start()	void stop()	void reset()	float get_time()	global mutex buffer_gaurd
player																																
SOCKET sock																																
float x																																
float y																																
char[4] state																																
float hp																																
skill																																
int frame																																
float x																																
float y																																
char type																																
float attack_power																																
생성자(float, float, char, float)																																
void update()																																
aabb get_bb()																																
game_manager																																
2Darray<int, player> players																																
2Darray<int, skill> skills																																
void add_player(int, SOCKET)																																
void update()																																
bool intersects(aabb1, aabb2)																																
void handle_collision()																																
void broadcast()																																
bool end_game()																																
void start_game()																																
timer																																
float start_time																																
void start()																																
void stop()																																
void reset()																																
float get_time()																																
global mutex buffer_gaurd																																



개발 환경 및 개발 일정

- 개발 환경



Visual Studio Code



PyCharm



Visual Studio



GitHub

- 개발일정(강민서)

일	월	화	수	목	금	토
26	27	28	29	30	31	1
2	3	4	5	6	7	8
	몬스터, 강화 효과 등 코드 정리	몬스터, 강화 효과 등 코드 정리		적 객체 선언		Enemy::update() 작성
9	10	11	12	13	14	15
			Client_rcv_thread() 작성	Recv_info() 작성		Recv_info() 작성
16	17	18	19	20	21	22
	Recv_buffer 작성		Update_info() 작성	Update_info() 작성		
23	24	25	26	27	28	29
	Update_info() 작성	Intersects() 작성	Intersects() 작성			
30	1	2	3	4	5	6
	로비 씬 제작	엔딩 씬 제작	Update_info() 작성	로비 씬 제작	엔딩 씬 제작	
7	8	9	10	11	12	13

매주 일요일에는 해당 주 구현 사항 합산 후 테스트

완료된 일정

추가된 일정

변경된 일정

- 개발일정(박지훈)

일	월	화	수	목	금	토
26	27	28	29	30	31	1
2	3	4	5	6	7	8
	서버 main() 작성	서버 main() 작성		통신에 필요한 구조체 선언		
9	10	11	12	13	14	15
	Server_rcv_thread read 작성		Recv_info()작성	Recv_info()작성		
16	17	18	19	20	21	22
	클라이언트 send_info() 수정 Skill 객체 선언	Skill 객체 내부 함수 구현	Send_buffer 작성 Game_manager::update() 작성			
23	24	25	26	27	28	29
	Handle_collision() 작성		Handle_collision() 작성	서버 start_game() 작성		
30	1	2	3	4	5	6
	서버 start_game() 작성	서버 end_game() 작성	서버 start_game() 작성			
7	8	9	10	11	12	13
	서버 start_game() 작성	서버 end_game() 작성				

작업의 일관성을 위해 일정 교환

- 개발일정(신태양)

일	월	화	수	목	금	토
26	27	28	29	30	31	1
2	3	4	5	6	7	8
		클라이언트 통신 관련 선언 소켓, 락 등	클라이언트 send_info() 작성			클라이언트 send_info() 작성
9	10	11	12	13	14	15
		Game_manager 선언 내부 변수 내부 함수	Broadcast() 작성	Broadcast() 작성		
16	17	18	19	20	21	22
	Skill 객체 선언 클라이언트 send_info() 수정	Skill 객체 내부 부함수 구현	Send_buffer 작성	Game_manager::update() 작성		
23	24	25	26	27	28	29
	Timer 작성		Send_buffer 수정	클라이언트 Start_game() 작성	Send_info 수정	
30	1	2	3	4	5	6
	클라이언트 Start_game() 작성	클라이언트 end_game() 작성	서버 동기화 구조 변경		서버 동기화 구조 변경	
7	8	9	10	11	12	13
	서버 동기화 구조 변경					