

# Toy Ground

2 0 1 6 1 8 0 0 0 6

김 동 석

2 0 1 6 1 8 0 0 1 2

김 영 준

# 목차

---

01

---

게임 소개

02

---

개발과정  
및 일정

04

---

개발 내용

03

Git hub

05

향후 계획  
및 문제점

# 01

## 게임 소개

장르: 대전 액션 TPS 게임

게임 목적: 상대를 견제하며 보석을 획득하는 것

최종 승리: 보석을 10개 이상 먹고 살아남기

플레이 방식: 1:1, 2인 대전 게임 플레이



# 02

## 개발 과정



## 클라이언트

# 02

## 개발 과정

```
class LobbyServer {
public:
    LobbyServer(short lobby_id);
    ~LobbyServer();

    void ClientAccept();
    void Worker();
    void ProcessPacket();

public:
    void SendLoginOk();
    void SendLoginFail();
    void SendSignIn();
    void SendSignOut();
    void SendMatchStart();
    void SendUserReady();
    void SendCancel();

    void SendPacket();

    void error_display();

    std::uniform_int_distribution<int> dist;
    std::default_random_engine eng;

private:
    HANDLE iocp;
    WSADATA wsaData;
    SOCKET listenSocket;
    SOCKADDR_IN serverAddr;
    DWORD flags;
    int addrlen = sizeof(SOCKADDR_IN);

    SOCKADDR_IN clientAddr;
    SOCKET clientSocket;

    short lobbyID;
    std::array<CLIENT*, MAX_PLAYERS> clients;
};
```

### PC에서의 공유 메모리

- 다른 코어에서 보았을 때 업데이트 순서가 틀릴 수 있음
- 메모리의 내용이 한 순간에 업데이트 되지 않을 때도 있음
- 언젠가는 메모리에 대한 일관성을 보장할 수 있음
- 자기 자신의 프로그램 내부에서만 사용
- 캐시의 일관성은 지켜야 함
- 캐시라인 내부의 쓰기

### 해결책

- 공유 메모리, 메모리 일관성
- lock을 통해 순서의 어
- 문제가 되는 어긋남을
- 기존의 멀티쓰레드용

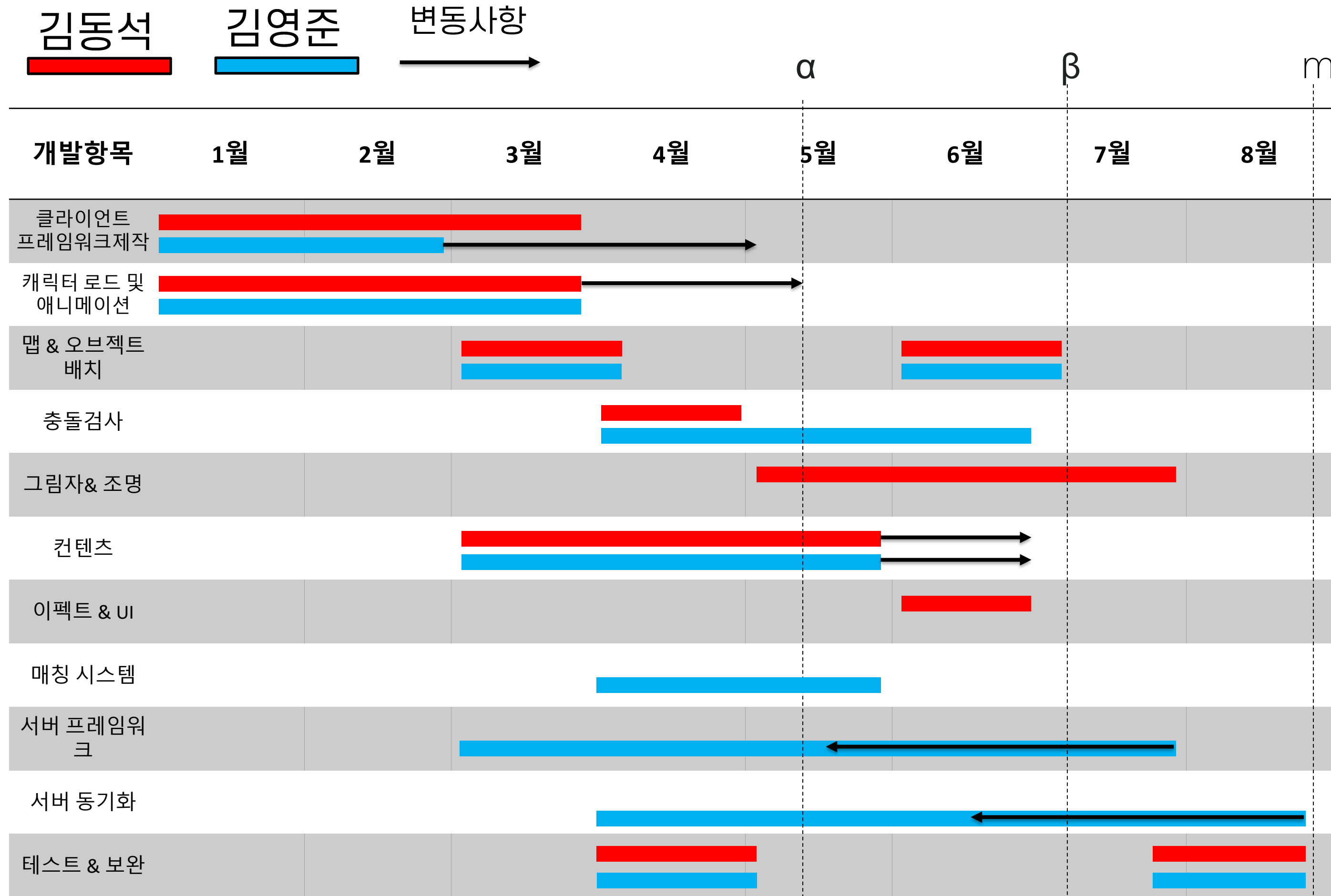
```
!!
num connection: 0
LOGIN OK ID: 1
```

```
99 void Dummy::ProcessPacket(int id, unsigned char packet[])
100 {
101     switch (packet[1]) {
102     + case LC_LOGIN_OK: { ... }
110     case LC_LOGIN_FAIL: cout << "login_fail\n"; break;
111     + case LC_USERINFO: { ... }
121     + case BC_AUTO_ACCEPT_OK: { ... }
129     + case LC_MATCHSTART: { ... }
140     case BC_PLAYER_ROT: break;
141     case BC_PLAYER_POS: break;
142     + case BC_JOIN_OK: { ... }
147     case BC_JOIN_FAIL: cout << "join fail\n"; break;
148     case BC_AUTO_ACCEPT_FAIL: cout << "auto accept fail\n"; break;
149     case BC_ROOM_ENTERED: cout << "room entered\n"; break;
150     case BC_NEW_ROOM_HOST: cout << "new room host\n"; break;
151     case BC_LEFT_TIME: cout << "left time\n"; break;
152     case BC_READY: cout << "ready\n"; break;
153     case BC_GAME_START: cout << "start\n"; break;
154     + case BC_GAME_START_AVAILABLE: { ... }
158     + case BC_UPDATED_USER_INFO: { ... }
162     + default: { ... }
166     }
```

서버

# 02

## 개발 과정



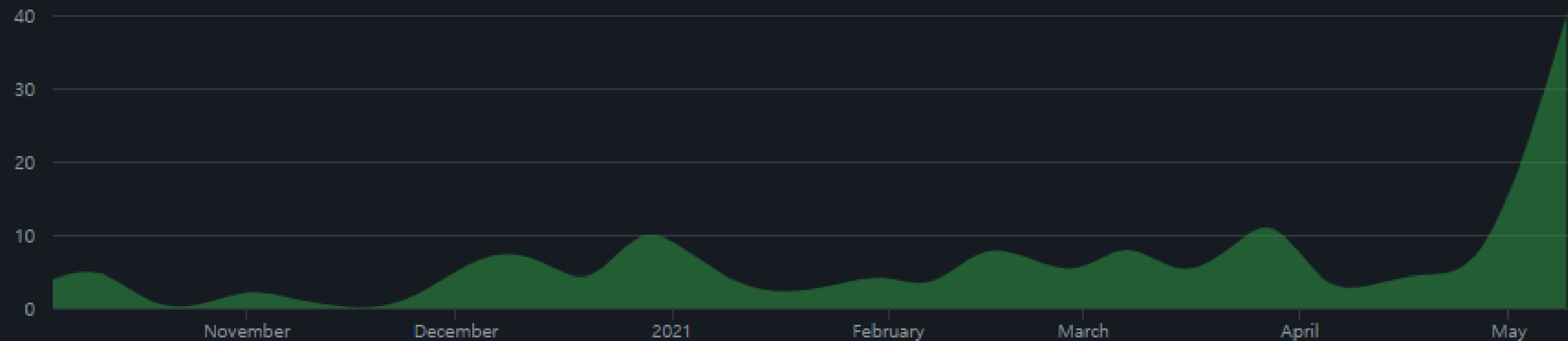
# 03

## Git hub

Oct 4, 2020 – May 11, 2021

Contributions: Commits ▾

Contributions to main, excluding merge commits and bot accounts

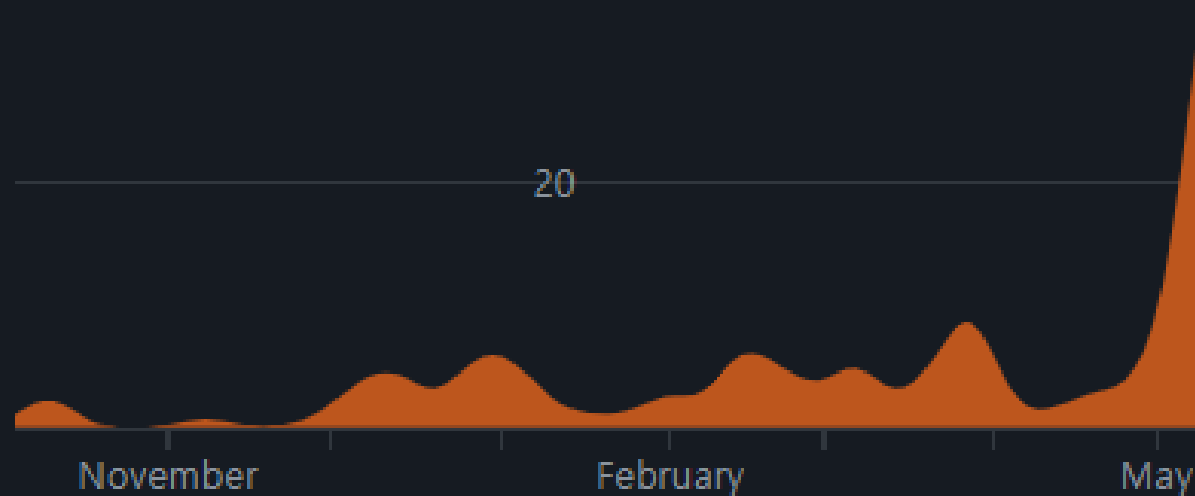


김동석

**DorongSeok**

128 commits 4,311,572 ++ 3,119,891 --

#1

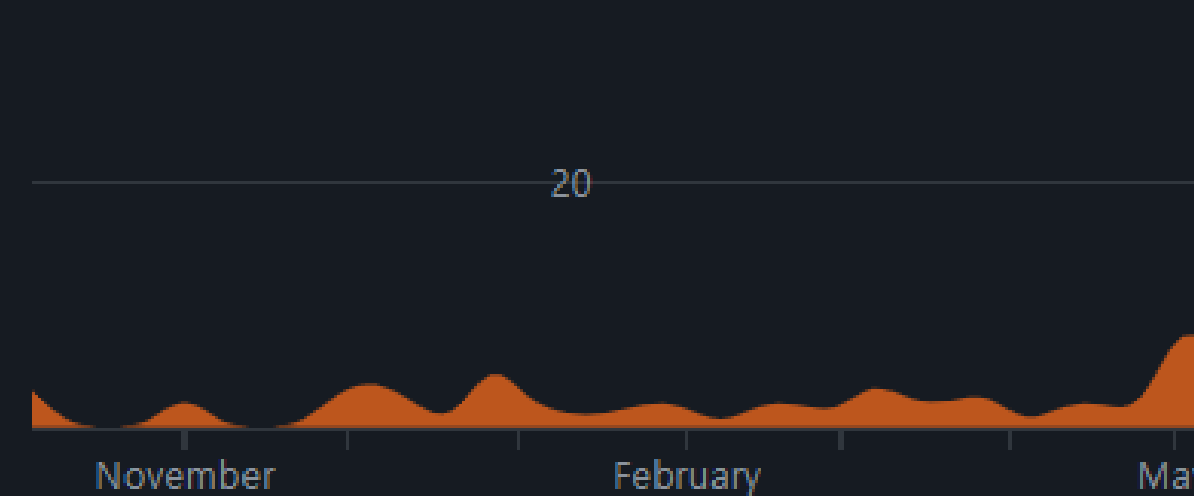


김영준

**rladudwns4643**

65 commits 596,467 ++ 19,816 --

#2





# 04

## 개발 내용

Client	김동석	프레임워크 개발 (100%) 애니메이션 (80%) FBX 모델 로딩 (100%) 맵 Loader 구현 (100%) 맵 및 오브젝트 배치 (100%) 충돌체크 (50%) Bounding box (80%)
Server	김영준	Server: iocp 모델 소켓 입출력 (90%) 멀티 쓰레드(timer, worker) (90%) Client: Nonblocking io모델 소켓 입출력 (90%) Singleton, OOP 개념을 적용한 클라이언트~서버 연동 (90%)



# 05

## 향후 계획 및 문제점

### 문제점

Client	김동석	무료로 구한 애니메이션이 부자연스러워 직접 편집해 보려고 한다.  그려야 하는 객체가 많아 FPS 관리가 어렵다.
Server	김영준	매칭 시스템 구현을 위해 새로운 프로젝트를 만들지, 기존 프로젝트에 새로운 쓰레드 추가로 해결할지 고민 DB, AWS, SQL 등 데이터에 대한 구현

### 향후 계획

Client	김동석	게임 콘텐츠 추가 그림자 및 조명 배치 UI 추가 배치 쉐이더 이펙트 추가 애니메이션 수정
Server	김영준	매칭 시스템 최적화 및 더미 클라이언트를 통한 최대 동시접속자 수 확인 및 트러블 슈팅 추가되는 콘텐츠에 대한 패킷 교환 시스템 추가