

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 개요 및 조작

04 개발 내용

02 기술요소
중점 연구 분야

05 문제점 및
보완책

03 역할분담

06 향후 개발 일정
및 데모시연

01

개요 및 조작

장르: 대전 액션 TPS 게임

게임 목적: 상대를 견제하며 보석을 획득하는 것

최종 승리: 보석을 10개 이상 먹고 살아남기

플레이 방식: 1 vs 1, 2인 대전 게임



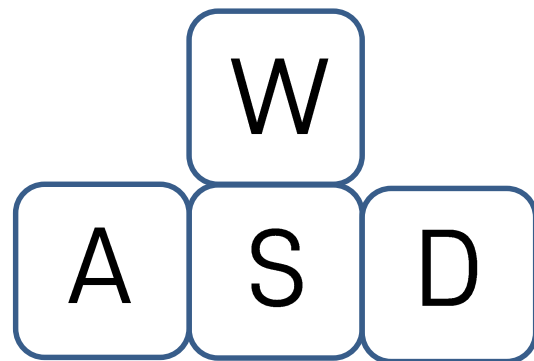
01

개요 및 조작

게임 조작

키보드

이동



스킬 On/Off

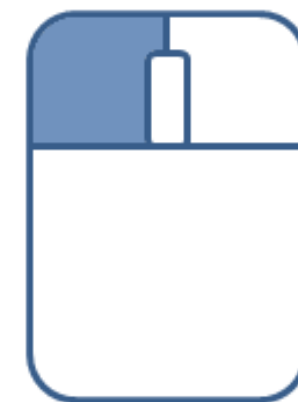


점프

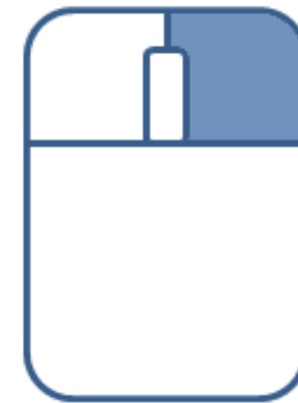


마우스

공격



조준
(공격범위표시)



02

기술요소
중점 연구 분야

Client Part | 김동석

디자인 패턴을 적용하여 프레임워크 제작
미리 제작되어 있는 모델의 구조를 파악 후 애니메이션 적용

Server Part | 김영준

패킷 크기와 게임 로직 처리를 위한 데이터 크기의 최적화
AWS 서비스를 통한 DB관리(RDS)

03

역할분담

Client Part
김동석

DirectX

프레임워크 개발

애니메이션

맵 및 오브젝트

충돌체크

UI 구현 및 배치

Server Part
김영준

IOCP

프레임워크 개발

로비-매칭 시스템

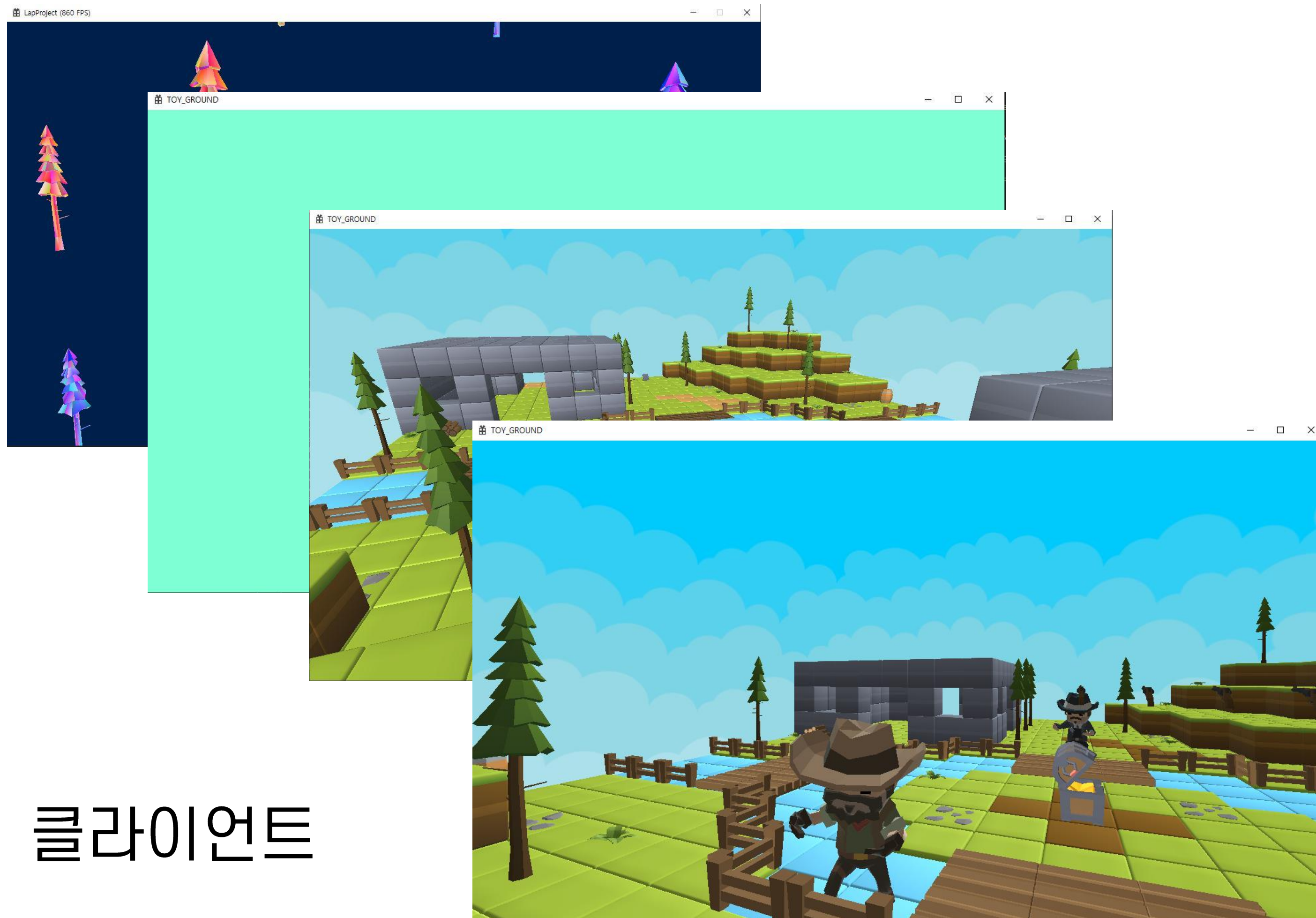
멀티 쓰레드

충돌체크

SQL

04

개발 내용



플라이언트

04

개발 내용

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

    void ClinetAccept(int id);
    void Worker();
    void ProcessPacket(int id, void* buf);

    class Object {
    public:
        Object();
        virtual ~Object() {};

        explicit Object(const Object& other) noexcept;
        Object& operator=(const Object& other);

    public:
        virtual bool Update(float elapsedTime, bool is_player);
        bool BulletUpdate(float elapsedTime);
        void AddForce(XMFLOAT3 force, float elapsedTime, bool isBullet);

    public:
        XMFLOAT4X4 GetMatrix() const;

        XMFLOAT3 GetPosition() const;
        void SetPosition(float x, float y, float z);
        void SetPosition(XMFLOAT3 xmfPos);

        XMFLOAT3 GetPrePosition() const;
        void SetPrePosition(XMFLOAT3 xmfPos);

        XMFLOAT3 GetLook() const;
        void SetLook(const float& x, const float& y, const float& z);
        XMFLOAT3 GetPreLook() const;
        void SetPreLook(XMFLOAT3 look);

        XMFLOAT3 GetUp() const;
        void SetUp(const float& x, const float& y, const float& z);
        XMFLOAT3 GetRight() const;
        void SetRight(const float& x, const float& y, const float& z);

        XMFLOAT3 GetRotation() const;
        XMFLOAT4 GetRotationQuaternion() const;
    };

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_CLIENTS> clients;
};
```

서버

```
C:\Users\wt1sna\OneDrive\Project\2021_graduation_0912\demo_server...
GET_DUMMY_LOGIN
Send ID:1 Packet: 30
ProcessPacket: 24
GET_CL_REQUEST_USER_INFO
Send ID:1 Packet: 34
ProcessPacket: 22
GET_CL_AUTOMATCH
user: 2connect
ProcessPacket: 99
GET_DUMMY_LOGIN
Send ID:2 Packet: 30
ProcessPacket: 24
GET_CL_REQUEST_USER_INFO
Send ID:2 Packet: 34
ProcessPacket: 22
GET_CL_AUTOMATCH
Send ID:0 Packet: 31
ProcessPacket: 41
SEND_READY_ROOM: 1
Send ID:1 Packet: 35
Send ID:2 Packet: 35
```

```
C:\Users\wt1sna\OneDrive\Project\2021_graduation_0912\demo_battle\Debug\demo_battle.exe
CreatePlayerSlot start
CreatePlayerSlot compleat
Maps\FirstMap.txt
Connect
1 : accept
SendBattleLoginOKPacket: 1
[main]: Send: 1 id: 1
2 : accept
SendBattleLoginOKPacket: 2
[main]: Send: 1 id: 2
SendAutoRoomReadyPacket: 0 room: 1
[main]: Send: 41 id: 0
SendRoomEnterPacket: to: 1 enterer: 1 ready: 0 player_no: 1 name: r m
mr: 1001 isMnr: 0
0 player_no: 1 name: r mnr: 1001 isMnr: 0[main]: Send: 5 id: 2
[main]: Send: 5 id: 1
SendRoomEnterPacket: to: 2SendRoomEnterPacket: to: 1 enterer: 2 ready: 0 player_no: 1 name: r mnr: 1002 isMnr: 0
[main]: Send: 5 id: 1
enterer: 2 ready: 0 player_no: 1 name: r mnr: 1002 isMnr: 0
[main]: Send: 5 id: 2
SendRoomJoinSuccess: 1 mnr: 0
[main]: Send: 3 id: 1
SendRoomJoinSuccess: 2 mnr: 0
[main]: Send: 3 id: 2
[main]: Send: 8 id: 1
[main]: Send: 8 id: 2
SendGameStartPacket: 1
[main]: Send: 9 id: 1
SendGameStartPacket: 2
```

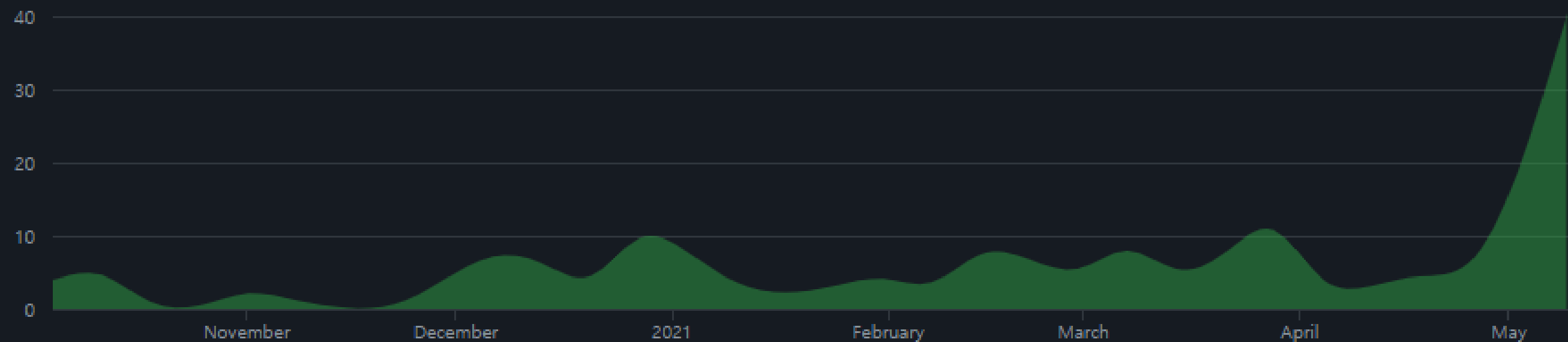

04

개발 내용

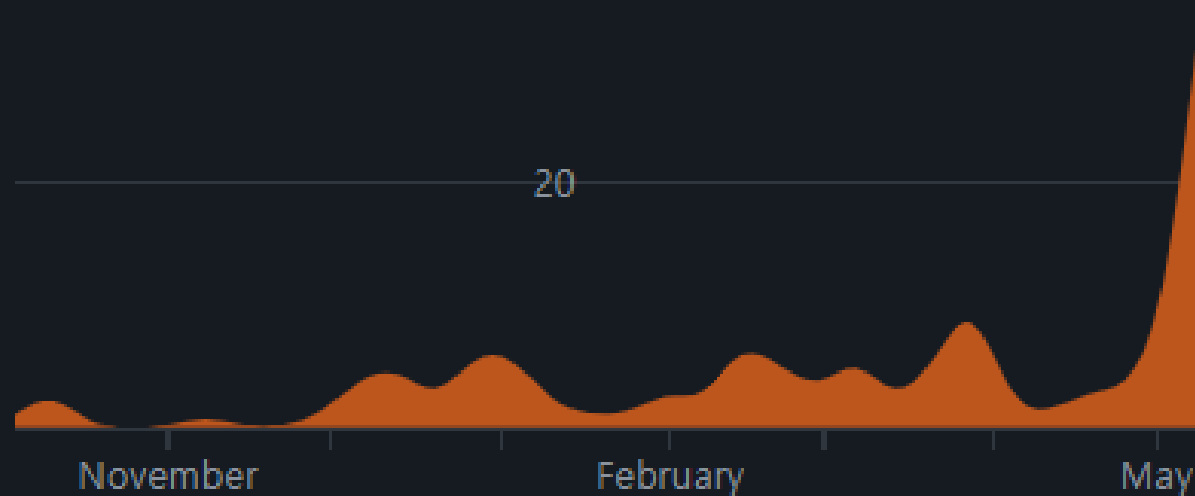
Oct 4, 2020 – May 11, 2021

Contributions: Commits ▾

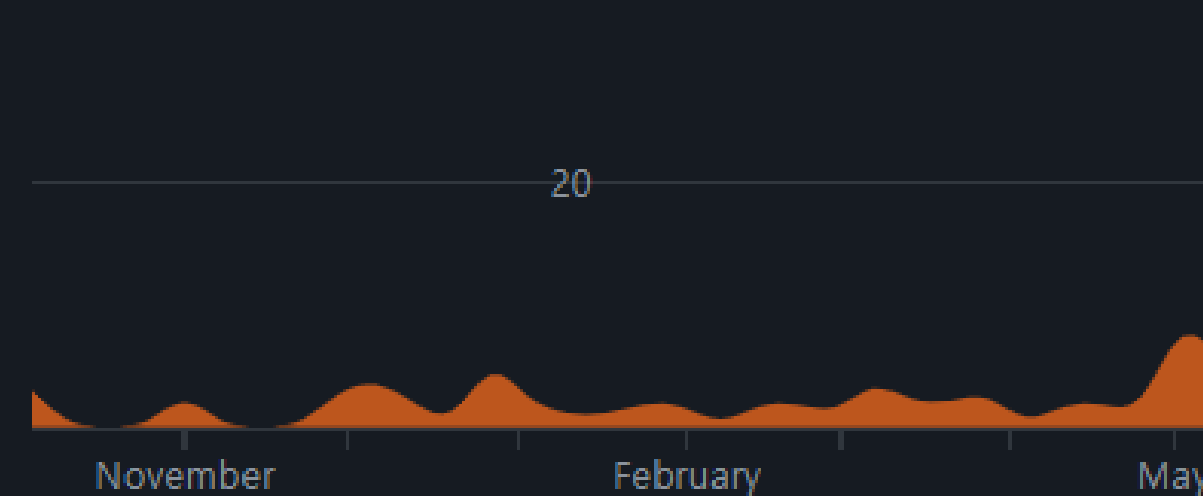
Contributions to main, excluding merge commits and bot accounts



김동석
DorongSeok #1
128 commits 4,311,572 ++ 3,119,891 --



김영준
rladudwns4643 #2
65 commits 596,467 ++ 19,816 --



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 (100%)
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

문제점 및 보완책

김동석 - 클라이언트

문제점	무료로 구한 애니메이션이 부자연스럽다. 많은 객체를 그리다 보니 FPS가 떨어진다.
보완책	애니메이션을 툴을 통해 직접 편집한다. 변하지 않는 값들은 미리 계산하여 불러오고 프레임워크 내에서는 계산하지 않는다.

김영준 - 서버

문제점	매칭 시스템 구현을 위해 새로운 프로젝트를 만들지, 기존 프로젝트에 새로운 쓰레드 추가로 해결할지 고민 DB, AWS, SQL 등 데이터에 대한 구현
보완책	매칭 시스템을 구현하기 위해 Matching Server 프로젝트를 새로 만들어 기존 Lobby -> Battle Server 형태에서 Lobby의 매칭 시스템을 분리해 Lobby -> Matching -> Battle의 구조를 가지도록 구현 Amazon RDS(SQL) 튜토리얼을 통해 DB 구축하여 데이터 관리 구현

06

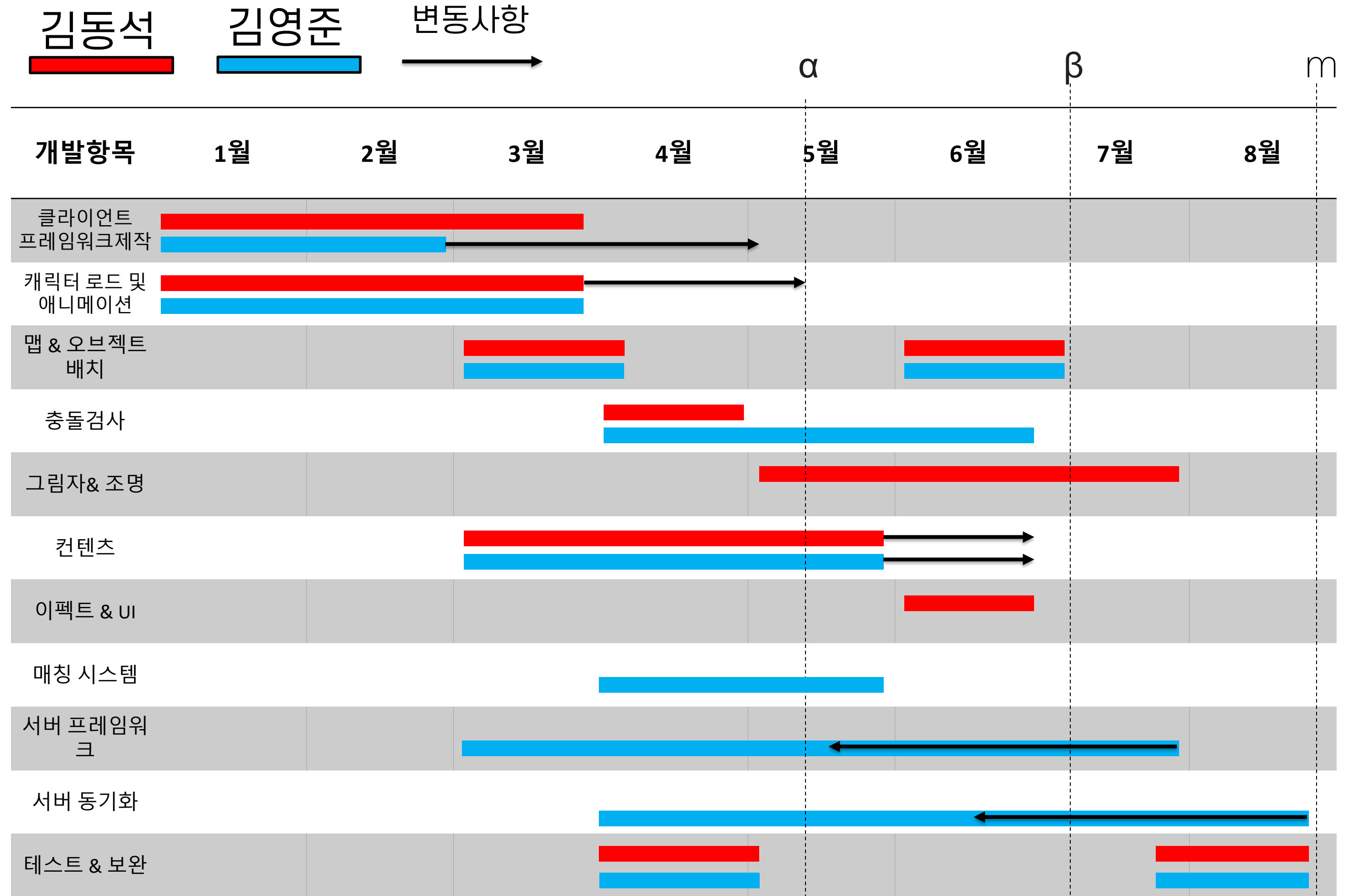
향후 개발 일정

향후 개발 일정

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

06

향후 개발 일정



06

데모 시연

데모 시연

QnA

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