

[Toy Ground(토이 그라운드)]

22 주	2020. 5. 23 ~ 2021. 5. 29	작성자	김동석
이번 주 한 일	<p>[0] 공동 05.24 주간 회의 - 일주일간 계획공유 - 앞으로의 계획</p> <p>[1] 김영준(서버) 주간목표: - 피드백 기반으로 todo list 작성</p> <p>05.24 기존에 정리하던 Effective C++ 정리</p> <div> <ul style="list-style-type: none"> ▶ #35: 가상함수 대응의 개발 방식도 염두에 두자 ▶ #36: 상속받은 비 가상 함수를 파생 클래스에서 재정의하지 말자 ▶ #37: 상속받은 기본 매개변수 값은 재정의 하지 말자 ▶ #38: has-a(~는 ~을 가짐), is_implemented_in_terms_of(~는 ~을 써서 구현)을 모형화 할 때는 객체 합성을 사용 ▶ #39: private 상속은 심사숙고해서 구현 ▶ #40: 다중상속(MI)는 심사숙고해서 사용 ▶ #41: 템플릿 프로그래밍의 중요한 것은 암시적 인터페이스와 컴파일 타임 다형성 ▶ #42: typename의 두 가지 의미 ▶ #43: 템플릿으로 만들어진 기본 클래스 안에 접근하는 방법 ▼ #44: 매개변수에 독립적인 코드는 템플릿으로부터 분리하자 </div> <p>05.25 todo list 작성 한 것, 할 것으로 분류하여 리스트를 작성</p>		

한거

▼ 서버

서버 구조를 나눠 여러 CPU에서 사용 가능하도록 구현 (Lobby, Battle, Client)

Lobby → Client 연동, Lobby → Battle Room 연동 (Client 접속을 위한 정보), Lobby → Client Room
정보 전달, Client → Battle 연동 (게임 플레이) (패킷 구조는 머메이드로 정리)

priority_queue을 통한 Timer 클라이언트 패킷 Event 관리

Timer EVENT을 통한 게임 시작 전 딜레이, 보석 생성 로직, 게임 종료

매칭된 유저의 정보를 받아 하나의 Room에 포함하여 게임 로직 실행

먼저 들어온 유저가 방장이 되어, 모두가 준비 상태가 되었을 때 게임 시작 가능 패킷을 전달

클라 패킷 입력 처리를 통한 캐릭터 상태를 받아오고, 해당 상태를 다른 클라이언트들에게 전달
입력을 받은 클라 내용을 해당 클라가 포함된 Room에 전달(msg)

atomic한 msg 처리를 위해 flushMsg 구현

(msg를 바로 처리하지 않고 새로운 msg로 복사한 뒤 처리)

▼ 클라

NetCore, Service의 Event 단위의 유지 보수에 용이한 게임 클라이언트 ~ 서버 연동 시스템

패킷이 들어오면 Service에 전달하여 클라에서 전달받은 패킷 처리

클라에서 보낼 패킷이 있으면 Service을 통해 netCore의 Send 호출

이런 구조를 통해 Server ~ Client 연동에 있어 패킷을 주고 받는 내용을 한눈에 볼 수 있음

캐릭터 상태에 따른 애니메이션 key frame을 전달하기 위해 animation_state 전달

할거

DB연동을 통한 MMR 기반 Matching Server 구현

2인 이상 플레이어 방 처리(매칭)

총알 구현

점프

죽었을때 벨는거

HP

UI

로비씬

게임 종료 씬

게임 방 리셋 패킷 처리

애니메이션 고쳐야함

소리

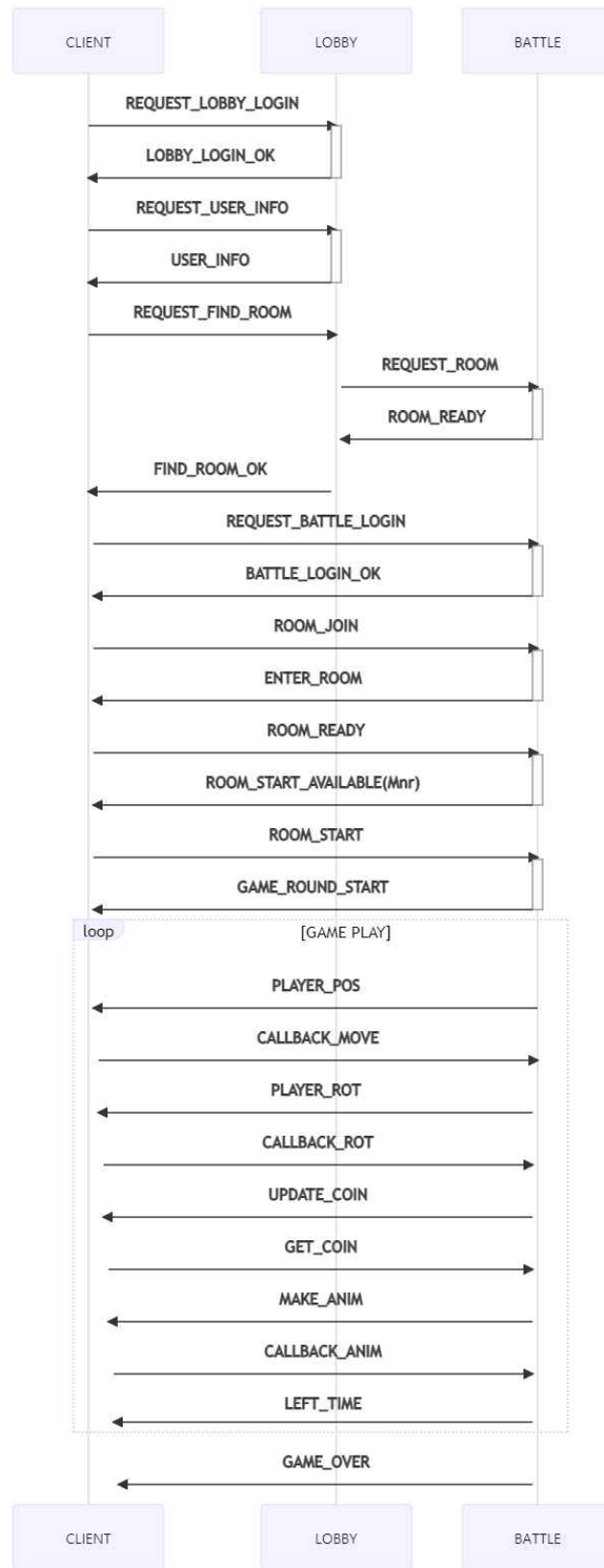
로그인하는거

그림자

맵충돌

리스폰하는거

05.26 패킷 전달 과정을 도식화(머메이드)



05.26 ~ 27 게임서버 과제 수행

GetQueuedCompletionStatus을 통한 Timer Event처리
AI Thread를 사용한 AI 개발
시야처리, 섹터링 등 다양한 실습을 진행할 수 있었음

05.28 귀가

[2] 김동석(클라이언트)

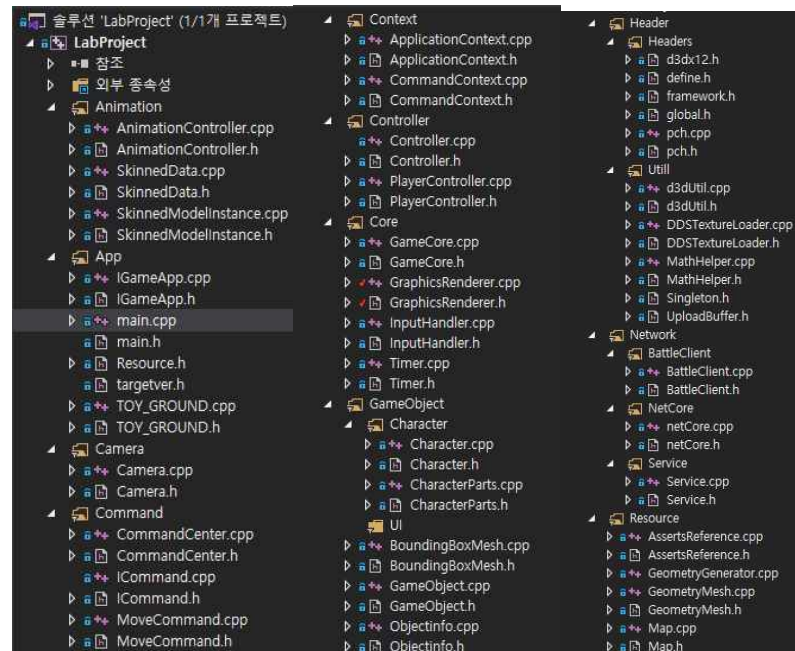
주간 목표 :

- UI를 위한 빌보드 구현
- 파일, 코드 정리

05.24 파일, 코드 정리

- 파일 정리

프로젝트 파일들 솔루션 탐색기에서 필터를 이용해 분류

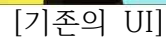


[솔루션 탐색기에서 분류한 모습]

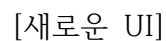
- 코드 정리

지저분하게 짜여진 코드들 정리, 함수로 분리, 사용하지 않는 함수 파일 삭제, 클래스 정리

- UI 재설계



게임 내에서 상대방의 위치를 파악 하는게 중요하기 때문에 최대한 화면을 가리지 않도록 간단하게 수정



	<p>- UIInterface UI를 관리하는 클래스인 UIInterface 추가 마우스를 이용해 클릭할 수 있고 스프라이트 애니메이션 형식으로 그릴 수 있도록 프레임을 관리하는 기능 추가</p> <p>- Button, ImageView UIInterface 클래스를 상속 받는 Button과 ImageView 추가 Button - 클릭 가능(Lobby Scene에서 사용 예정) ImageView - 이미지만 그림(UI, 타이틀 등에 사용 예정)</p> <p>- SpriteAnimation 이미지에 적용되는 애니메이션을 관리하는 클래스 현재 프레임, 애니메이션 스피드 등을 관리</p> <p>- UI PipeineState 파이프라인에 UI 관련 추가</p> <p>- 빌보드(20%) 상대방의 머리 위에 표시되는 이름과 체력을 빌보드로 처리하려 하는데 블록 뒤에 숨었을 경우에는 UI 때문에 위치가 들키기 때문에 상대방이 보일 때만 표시하도록 설계했고 코딩 중이다. 그리고 크기가 너무 작아지면 체력 수치가 보이지 않기 때문에 일정 크기를 유지하도록 설정해야겠다.</p>
다음 주 할 일	<p>[0] 공동 05.31 주간 회의 (23주차) 안건 : 일일계획 공유</p> <p>[1] 김영준 (서버) 주간목표 - Match make server 구현 시작</p> <p>[2] 김동석 (클라이언트) 주간 목표 - Bullet 구현 - Bullet 충돌처리(수정), 서버와 연동</p>

문제점	<p>[1] 김영준 (서버)</p> <ul style="list-style-type: none"> - 중간발표가 끝나고 미뤄둔 과제를 해결했다. - 다른 팀의 발표를 보고난 뒤 어떤 방식으로 개발했는지 궁금한 팀이 많았다. 나를 포함한 많은 팀이 통과했으면 좋겠다는 마음이 들었다. - 팀원들의 각자의 사정으로 피드백을 기반으로 todo list 제작이 늦어졌다. - 중간발표 이전 보고서 제출 순서를 헷갈려 정상적인 시간에 보고서를 메일로 전송하지 못했다. 죄송합니다. <p>[2] 김동석 (클라이언트)</p> <ul style="list-style-type: none"> - 코드를 정리했다. 중간발표를 준비하면서 급하게 추가한 코드들을 함수로 분리하고 변수명 등을 수정했다. 새로운 코드를 추가할 때 내가 설계한 프레임워크 틀에서 벗어나지 않도록 주의해야겠다. - 상대 플레이어의 체력 등을 표시할 때 어떨 때 표시해야 할지 고민을 좀 더 해봐야겠다. 브롤스타즈의 경우 탭뷰이기 때문에 무조건 같은 크기로 표시하면 되지만 ToyGround는 그렇지 않기 때문에 적합한 방법을 찾아야겠다.
-----	---

[추가]