

STEP 4. CLASS DIFINITION

: STEP 3의 최종 결과물인 Class Diagram을 구현에 적합하게 재검토하고,
최종적으로 도출된 Class들의 핵심 Specification을 명세하는 과정.

요약: 멀티플레이를 고려한 클라이언트 - 게임 모델 - 서버의 서브시스템 흐름은 유지되었고
더욱 필요하다 생각되는 클래스와 메소드가 추가됨

- Client Side

1. Core Class : 특정 기능의 수행이 목적인 클래스

클래스명	Main	
이 프로젝트의 main 클래스이자 패킷을 제외한 모든 이벤트의 Listener인 컨트롤러 생성자가 private이고 모든 Attribute를 바로 접근할 수 있도록 static Attribute를 가진 singleton 형태 를 띄고 있다.		
Super Class	JFrame	
interface	KeyListener, ActionListener, ListSelectionListener	
변수명	데이터타입	내 용
mctrl	static Remocon	싱글 플레이에서 사용하는 메인 Remocon 클래스
camera	static Camera	인게임 화면을 출력하는 컨테이너
login	static LoginPage	로그인 화면 컨테이너
waitr1	static WaitingR1	대기실 1 컨테이너
waitr2	static WaitingR2	대기실 2 컨테이너
build	static BuildPage	조립실 컨테이너
result	static Result	결과창 컨테이너
cTextout	static JTextField	Enter 키 입력에 포커스를 인식받는 채팅 전용 컴포넌트
gate	static Sock	소켓 처리 클래스
pressLock	static BitSet	멀티플레이에서 한번의 KeyPress로 중첩된 소켓을 보내지 않도록 중재하기 위한 BitSet
wt	WorkerThread	워커 스레드
WIDTH	static final int	프레임 고정 가로크기
HEIGHT	static final int	프레임 고정 세로크기
메소드명	리턴타입	내 용
Main	생성자	생성자
run	void	<Override> 스레드 루프
keyPressed(KeyEvent e)	void	<Override> 키 입력
KeyReleased(KeyEvent e)	void	<Override> 키 입력 해제
actionPerformed(ActionEvent e)	void	<Override> 액션 이벤트 입력
valueChanged(ListSelectionEvent e)	void	<Override> 리스트 값 선택 이벤트 입력
keyOperate(char type, boolean flag)	void	입력받은 키를 싱글플레이와 멀티플레이에 맞게 컨트롤

클래스명	Sock	
멀티플레이 네트워크 서버와 연결되는 소켓 클래스.		
Super Class		
interface		
변수명	데이터타입	내 용
recvBuffer	ByteBuffer	수신한 내용을 처리하기까지 스트림으로 보관하기 위한 바

		이트 버퍼
sockChannel	SocketChannel	서버와 연결되는 소켓 채널
selector	Selector	시그널 처리 셀렉터
myIndex	int	멀티플레이에서 Actor의 인덱스
myTeam	byte	멀티플레이에서 Actor의 팀
master	Main	메인 클래스
activated	boolean	소켓 활성화 상태
TNONE	static final byte	중립 팀 넘버
TRED	static final byte	레드 팀 넘버
TBLUE	static final byte	블루 팀 넘버
메소드명	리턴타입	내 용
Sock(Main master)	생성자	생성자
connect()	void	서버와 연결
send(ByteBuffer byteBuffer)	void	서버로 데이터 전송
recv()	void	서버로 부터 데이터 수신
recvProc()	void	서버로 부터 데이터를 완전히 수신하면 처리하는 과정
cut(int length, ByteBuffer buffer)	ByteBuffer	버퍼를 compact해서 반환
encodeString(String str)	ByteBuffer	문자열을 인코딩하여 인코딩된 데이터 길이와 함께 버퍼로 만들어 반환
cutString(ByteBuffer String)	String	버퍼에서 인코딩된 문자열 데이터를 잘라내어 디코 딩하여 반환
completeRecv(ByteBuffer buffer)	byte	패킷의 헤더를 디코딩하여 모든 데이터가 들어왔는지 확인
이하 메소드 21개는 패킷을 주고받기 위한 메소드이며, 이에 대한 자세한 명세는 Multiplay Analysis의 Packet Graph를 참조하세요!		
sendPak0~9	Client -> Server 패킷 전송	
reactProc, reactPak0~5	보낸 패킷에 대한 응답으로 온 패킷을 처리하는 메소드	
roomProc	대기실의 게임방 리스트에 관한 패킷을 처리하는 메소드	
channelProc	게임방에 관한 패킷을 처리하는 메소드	
chatProc	채팅에 관한 패킷을 처리하는 메소드	
gameProc	게임에 관한 패킷을 처리하는 메소드	

클래스명	WorkerThread		
Main Thread가 유저의 Action을 Listen하고 컨트롤 하기 때문에 실시간으로 갱신과 처리를 해야하는 기능 수행을 위한 Thread 클래스			
Super Class			
interface	Runnable		
변수명	데이터타입	내 용	
master	Main	메인 클래스	
thread	Thread	본인 스레드	
delay	int	프레임 딜레이	
userCtrl	Remocon[]	멀티플레이 기능 지원을 위한 유저 리모콘 배열	
메소드명		리턴타입	내 용
WorkerThread(Main master)		생성자	생성자
run		void	<Override> 스레드 루프
userControl(int index, byte type, byte flag, int syncX, int syncY, int angle)		void	멀티플레이 유저 컨트롤을 갱신
public void userAdd(byte team, int index, int x, int y, int topNum, int bottomNum, int weaponNum)		void	멀티플레이에서 새로운 유저 등록
public void objAdd(byte team, byte foward, int id, int obiType, int x, int y)		void	멀티플레이에서 새로운 객체 등록

objRemove(int id)	void	멀티플레이에서 객체 제거
projAdd(int id, int shotX, int shotY, int angle, int remain)	void	멀티플레이에서 새로운 투사체 등록
eventSync(int actor,int reactor, int point, int vib, byte isUp)	void	멀티플레이 게임에서 발생한 이벤트(공격 피해, 회복 등)의 상태를 갱신

2. Client GUI Class : GUI 제공이 목적인 클래스

::Component

클래스명	Logo	
게임 로고를 출력하는 컴포넌트		
Super Class	Canvas	
interface	Runnable	
변수명	데이터타입	내 용
buff	BufferedImage	더블 버퍼링을 위한 BufferedImage 객체
g2d	Graphics2D	draw하는 Graphics2D 객체
thread	Thread	스레드
delay	int	프레임 딜레이
pretime	int	시간 조절용 변수
ad2	ArrayList<ArcDesign>	원형 디자인
title	Image	타이틀 이미지
activated	boolean	페이지 활성 상태
메소드명		리턴타입
Logo		생성자
work()		void
stop()		void
run()		void
paint(Graphics g)		void
update(Graphics g)		void
dblpaint()		void

클래스명	ArcDesign		
원형 디자인 클래스			
Super Class			
interface			
변수명	데이터타입	내 용	
메소드명	리턴타입	내 용	

클래스명	idCard	
게임 방의 유저 한 슬롯의 정보를 표시하는데 이용되는 컴포넌트		
Super Class		
interface		
변수명	데이터타입	내 용
icon	JLabel	팀 등의 정보를 표시하는 라벨
id	JLabel	아이디를 표시하는 라벨
ready	JLabel	준비상태를 표시하는 라벨
메소드명	리턴타입	내 용
IdCard	생성자	생성자

add(JPanel master,int x,int y)	void	IdCard를 master 패널의 x,y좌표에 추가
setId(String name)	void	
setReady(boolean isReady)	void	
setTeam(byte team)	void	

클래스명	resultCard	
결과창의 유저 한	슬롯의 정보를 표시하는데 이용되는 컴포넌트	
Super Class		
interface		
변수명	데이터타입	내 용
team	JLabel	팀 정보 표시 라벨
id	JLabel	아이디 표시 라벨
score	JLabel	스코어 표시 라벨
deal	JLabel	누적 데미지 표시 라벨
메소드명	리턴타입	내 용
resultCard	생성자	생성자
add(JPanel master,int x,int y)	void	IdCard를 master 패널의 x,y좌표에 추가
setId(String name)	void	
setDeal(int eDmg)		
setScore(int kill, int death)	void	
setTeam(byte team)	void	

::Container

클래스명	LoginPage	
대기실 1 기능을	가시적으로 제공하는 GUI 패널 클래스	
Super Class	JPanel	
interface		
변수명	데이터타입	내 용
logo	Logo	게임 로고를 출력할 커스텀 컴포넌트
id	JTextField	id 입력 컴포넌트
pw	JTextField	pw 입력 컴포넌트
btnLogin	JButton	로그인 버튼
btnSign	JButton	회원 가입 버튼
btnSingle	JButton	싱글 모드 버튼
master	Main	메인 클래스
activated	boolean	페이지 활성화 상태
메소드명	리턴타입	내 용
LoginPage(Main master)	생성자	생성자
KeyProcess(KeyEvent e)	void	키 이벤트 실행 (Listener는 Main)
active()	void	페이지 활성화 지시
inactive()	void	페이지 비활성 지시
actionPerformed(ActionEvent e)	void	액션 이벤트 실행 (Listener는 Main)
isActivated()	boolean	활성화 상태인지 확인

클래스명	WaitingR2	
대기실 2 기능을	가시적으로 제공하는 GUI 패널 클래스	
Super Class	JPanel	
interface		
변수명	데이터타입	내 용
labelrid	JLabel	게임 방 번호를 표시하는 컴포넌트
labelname	JLabel	게임 방 이름을 표시하는 컴포넌트
idCard	IdCard	방 안의 유저 6명의 IDCARD 컴포넌트
cTextin	JTextArea	서버로부터 채팅 내용이 입력되는 컴포넌트

rnameout	TextField	생성할 방 이름을 입력할 컴포넌트
btnTRed	Button	방 참여 버튼
btnTBlue	Button	방 생성 버튼
btnReady	Button	준비 버튼
btnBuild	Button	조립실 들어가는 버튼
btnOut	Button	로그인 화면으로 돌아가는 버튼
master	Main	메인 클래스
activated	boolean	페이지 활성화 상태
메소드명	리턴타입	내 용
WaitingR2(Main master)	생성자	생성자
addChatMsg(String msg)	void	새로운 채팅 내용을 컴포넌트에 추가
setReady(int index, boolean ready)	void	index번째 idCard의 ready상태 변경
setId(int index, String name)	void	index번째 idCard의 id 변경
setTeam(int index, byte team)	void	index번째 idCard의 팀 변경
active()	void	페이지 활성화 지시
inactive()	void	페이지 비활성 지시
actionPerformed(ActionEvent e)	void	액션 이벤트 실행 (Listener는 Main)
valueChanged(ListSelectionEvent e)	void	리스트 선택 이벤트 실행 (Listener는 Main)
isActivated()	boolean	활성화 상태인지 확인

클래스명	WaitingR1	
대기실 1 기능을	가시적으로 제공하는 GUI 패널 클래스	
Super Class	JPanel	
interface		
변수명	데이터타입	내 용
rJlist	JList	게임 방을 나열하는 리스트 컴포넌트
selRindex	int	선택한 방 index
roomList	ArrayList<String>	방 정보 목록
cTextin	JTextArea	서버로부터 채팅 내용이 입력되는 컴포넌트
rnameout	JTextField	생성할 방 이름을 입력할 컴포넌트
btnJoin	JButton	방 참여 버튼
btnCreate	JButton	방 생성 버튼
btnBuild	JButton	조립실 들어가는 버튼
btnOut	JButton	로그인 화면으로 돌아가는 버튼
master	Main	메인 클래스
activated	boolean	페이지 활성 상태
메소드명	리턴타입	내 용
WaitingR1(Main master)	생성자	생성자
addChatMsg(String msg)	void	새로운 채팅 내용을 컴포넌트에 추가
clearrJlist()	void	roomList 를 클리어
removerJlist(int id)	void	roomList에서 방 번호가 id인 방을 찾아 제거
refreshrJlist()	void	rJlist 컴포넌트를 새로고침
active()	void	페이지 활성 지시
inactive()	void	페이지 비활성 지시
actionPerformed(ActionEvent e)	void	액션 이벤트 실행 (Listener는 Main)
valueChanged(ListSelectionEvent e)	void	리스트 선택 이벤트 실행 (Listener는 Main)
isActivated()	boolean	활성화 상태인지 확인

클래스명	ResultPage
게임 결과를 제공하는 GUI 패널 클래스	
Super Class	JPanel
interface	

변수명	데이터타입	내 용
idCard	resultCard[]	resultCard 배열
btnOut	JButton	퇴장 버튼
master	Main	메인 클래스
activated	boolean	페이지 활성 상태
메소드명	리턴타입	내 용
ResultPage(Main master)	생성자	생성자
setResult(int index, int kill, int death, int deal)	void	결과 갱신
active()	void	페이지 활성 지시
inactive()	void	페이지 비활성 지시
actionPerformed(ActionEvent e)	void	액션 이벤트 실행 (Listener는 Main)
isActivated()	boolean	활성화 상태인지 확인

클래스명	BuildPage	
로봇 조립 기능을	가시적으로 제공하는 GUI 패널 클래스	
Super Class	JPanel	
interface		
변수명	데이터타입	내 용
list	JList	모델 명을 나열하는 리스트 컴포넌트
BotImage	JLabel	조립되는 예상 이미지를 보여줄 라벨 컴포넌트
btnBot	JButton []	로봇 번호 버튼 컴포넌트
btnPart	JButton []	파트 종류 버튼 컴포넌트
btnOut	JButton	이전 페이지로 돌아가는 버튼 컴포넌트
btnBuild	JButton	로봇 조립을 요청하는 버튼 컴포넌트
wList	ArrayList<String>	무기 모델 목록
tList	ArrayList<String>	상체 모델 목록
bList	ArrayList<String>	하체 모델 목록
robot	int	조립중인 로봇 번호
model	int	리스트 컴포넌트 모델 종류 번호(상체/하체/무기)
weapon	int	선택한 무기 모델 번호
top	int	선택한 상체 모델 번호
bottom	int	선택한 하체 모델 번호
master	Main	메인 클래스
prevPage	int	이전 페이지 넘버
activated	boolean	페이지 활성 상태
WAIT1	static final int	대기실 1 페이지 넘버
WAIT2	static final int	대기실 2 페이지 넘버
메소드명	리턴타입	내 용
BuildPage(Main master)	생성자	생성자
selectButton(String type, int index)	void	버튼 선택에 따른 변화 갱신
active(int prevPage)	void	페이지 활성 지시
inactive()	void	페이지 비활성 지시
actionPerformed(ActionEvent e)	void	액션 이벤트 실행 (Listener는 Main)
valueChanged(ListSelectionEvent e)	void	리스트 선택 이벤트 실행 (Listener는 Main)
isActivated()	boolean	활성화 상태인지 확인

클래스명	Camera	
객체들의 리스트를 받아 viewPoint에 따라 draw해주는 클래스		
Super Class	Canvas	
interface		
변수명	데이터타입	내 용
buff	BufferedImage	더블 버퍼링을 위한 BufferedImage 객체
g2d	Graphics2D	draw하는 Graphics2D 객체
viewPoint	GameObject	Camera가 draw할 중심의 Object
projectileList	ArrayList<>	게임 내 모든 발사된 총알들이 있는 리스트
objectList	ArrayList<>	게임 내 모든 GameObject들이 있는 리스트
vib	int	Camera를 흔드는 정도
master	Main	메인 클래스
activated	boolean	게임 활성화 상태
메소드명	리턴타입	내 용
Camera()	생성자	생성자
paint(Graphics g)	void	update() 실행, Canvas override
update(Graphics g)	void	updateAll(), dblpaint() 실행
dblpaint()	void	뷰포인트 설정, 배경그리기, 총알, 객체그리기 실행
setViewP(GameObject obj)	void	뷰포인트 설정
getViewP()	GameObject	뷰포인트 반환
getObjList()	ArrayList	객체 리스트 반환
getProjList()	ArrayList	총알 객체 리스트 반환
updateForSingle()	void	싱글모드에서 사용하는 update
updateAll()	void	리스트의 정보를 업데이트한다.
active()	void	게임을 활성화함
inactive()	void	게임을 비활성화

3. Game Finite State Model Class : 게임에 등장하는 객체들의 상태와 행동이 추상화된 클래스들

<!> 이 클래스들은 Client와 Server가 공유하는 서브 시스템입니다.

<!> 이 문서에서 명세는 Client Side에서 사용되는 이미지 처리 기능을 포함합니다.
 <!> Server Side에서는 이미지 처리 기능만 제외하여 동일한 형태로 활용됩니다.

클래스명	abstract GameObject	
게임 내에 존재하는 모든 객체들의 조상 클래스		
Super Class		
interface		
변수명	데이터타입	내 용
id	int	GameObject가 갖는 고유의 id값
x	int	x좌표값
y	int	y좌표값
width	int	가로 길이값
height	int	세로 길이값
life	boolean	true인 동안 리스트에 존재한다
HP	int	현재 체력값
maxHP	int	최대 체력값
team	int	team 설정값
메소드명	리턴타입	내 용
GameObject()	생성자	생성자
draw(Graphics2D g, int cameraX, int cameraY)	void	<Abstract> Graphics2D 객체를 받아 Camera의 ViewPoint 상대좌표로 이 객체를 그린다.
update()	boolean	<Abstract> 이 객체의 정보를 업데이트한다.
genAftObj()	void	<Abstract> 이 객체의 수명이 다하면 다음 객체를 생성
setLocation(int x, int y)	void	이 객체의 x,y좌표를 설정한다.
getBound()	Rectangle2D	이 객체의 Bound를 Rectangle2D 객체로 반환한다.
getWidth()	int	이 객체의 가로 길이값을 반환한다.
getHeight()	int	이 객체의 세로 길이값을 반환한다.
getX()	int	이 객체의 x좌표 값을 반환한다.
getY()	int	이 객체의 y좌표 값을 반환한다.
hurtSync(int point, boolean isUp)	void	서버에서 객체의 생명력을 조정

클래스명	Land	
맵에 존재하는 땅을 표현한 클래스		
SuperClass	GameObject	
interface		
변수명	데이터타입	내 용
		protected로 GameObject의 변수를 공유
메소드명	리턴타입	내 용
Land(int x, int y, int width, int height)	생성자	생성자
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 땅객체를 draw한다.
update()	boolean	<Override> 객체의 정보를 업데이트한다.
getTopLine()	Line2D	객체의 가장 위쪽 Line을 리턴한다.(착지정보)
genAftObj()	void	<Abstract> 이 객체의 수명이 다하면 다음 객체를 생성

클래스명	Container	
로봇의 파츠들을 래핑하고 명령을 전달해 주는 클래스		
Super Class	GameObject	
Interface	Hieable	
변수명	데이터타입	내 용
top	Top	상체 파츠 객체
bot	Bottom	하체 파츠 객체
weapon	Weapon	무기 파츠 객체
foward	boolean	로봇이 바라보고 있는 방향
메소드명	리턴타입	내 용
Container(int x, int y, int id, byte team, Top top, Bottom bot, Weapon weapon)	생성자	생성자
move(boolean foward)	void	하체 파츠에게 움직임 명령 수행
jump(boolean dirjump)	void	하체 파츠에게 점프 명령 수행
isJumping()	boolean	하체 파츠가 JUMP 상태인지 반환
shot()	void	무기 파츠에게 발사 명령 수행
reload()	void	무기 파츠에게 재장전 명령 수행
skill()	void	상체 파츠에게 스킬발동 명령 수행
dimensionSync(int x, int y, int angle)	void	무기 파츠의 그림 좌표 동기화
hurtSync(int point, boolean isUp)	void	<Override> 서버에서 객체의 생명력을 조정
getBound()	Rectangle2D	<Override> 로봇의 바운드 반환
isCollision(GameObject other)	boolean	<Override> 로봇과 다른 객체와의 충돌여부 반환
isShoot(Projectile other)	void	<Override> 로봇이 총알에 맞으면, 데미지를 입힘
isHeal(HealPack heal)	void	로봇이 힐팩과 충돌하면, 체력을 높여줌
genAftObj()	void	<Override> 이 객체의 수명이 다하면 다음 객체를 생성
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 로봇의 파츠들을 그려줌
update()	boolean	<Override> 로봇의 정보들을 업데이트함
getAngle()	int	로봇의 발사각도를 반환
setDir(boolean foward)	void	로봇의 바라보는 방향을 설정
setId(int id)	void	로봇의 id를 설정
setTeam(byte team)	void	로봇의 팀을 설정

클래스명	Weapon	
무기 파츠의 조상 클래스, 탄환 발사를 담당한다.		
Super class	GameObject	
interface		
변수명	데이터타입	내 용
shotStart	long	탄환을 발사한 시간 기록
shotDelay	long	탄환마다 발사되는 사이의 시간
reloadDelay	long	재장전에 걸리는 시간
reloadStartTime	long	재장전을 시작한 시간 기록
maxAngle	int	최대 발사 각도
minAngle	int	최소 발사 각도
curAngle	int	현재 발사 각도
weight	int	무게
damage	int	데미지
shotX	int	발사 x좌표
shotY	int	발사 y좌표
L	BufferedImage[]	왼쪽을 바라보는 이미지의 배열
R	BufferedImage[]	오른쪽을 바라보는 이미지의 배열
shotLock	boolean	재장전시 발사를 할 수 없게 하는 flag
foward	boolean	무기가 바라보는 방향
index	int	발사 각도마다 무기 이미지를 설정
drawX	int	무기 이미지를 그리는 x좌표
drawY	int	무기 이미지를 그리는 y좌표
range	int	무기가 발사하는 탄환의 사정거리
메소드명	리턴타입	내 용
Weapon()	생성자	생성자
genAftObj()	void	<Override> 이 객체의 수명이 다하면 다음 객체를 생성
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 무기 객체를 그려줌
update()	boolean	<Override> 무기 객체 정보 업데이트
reload()	void	재장전
spinAngle(boolean pos)	void	발사 각도 조정
canShot()	boolean	발사가 가능한 상태인지 확인
shot()	void	탄환에 무기의 정보를 주어 탄환생성(발사)
getAngle()	int	발사 각도 반환
setAngle(int angle)	void	발사 각도 설정
setDir(boolean foward)	void	바라보는 방향 설정
setReload()	void	재장전 상태 설정
setShotLocation(int x, int y)	void	발사 좌표 설정
enforce(boolean on)	void	무기 강화 스킬에 사용되는 무기 공격력 강화

클래스명	MachineGun	
짧은 사거리, 빠른 연사력, 낮은 데미지, 많은 장탄수의 특징을 가진 머신건 클래스		
Super class	Weapon	
interface		
변수명	데이터타입	내 용
runout	int[]	반동으로 인한 총구의 흔들림을 표현하기 위한 정수배열
runoutIndex	int	배열에 사용되는 인덱스
메소드명	리턴타입	내 용
MachineGun()	생성자	생성자
spinAngle(boolean pos)	void	<Override> 머신건의 발사 각도 조정
canShot()	boolean	<Override> 머신건이 발사가 가능한 상태인지 확인
shot()	void	<Override> 머신건의 정보를 탄환객체에 주어 탄환생성(발사)
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 머신건 객체를 그려줌
update()	boolean	<Override> 머신건의 정보를 업데이트
reload()	void	<Override> 머신건 재장전

클래스명	Rifle		
중간 사거리, 중간 연사력, 중간 데미지, 중간 장탄수의 특징을 가진 라이플 클래스			
Super class	Weapon		
interface			
메소드명	리턴타입	내 용	
Rifle()	생성자	생성자	
spinAngle(boolean pos)	void	<Override> 라이플 발사 각도 조정	
canShot()	boolean	<Override> 라이플이 발사가 가능한 상태인지 확인	
shot()	void	<Override> 라이플 정보를 탄환객체에 주어 탄환생성(발사)	
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 라이플 객체를 그려줌	
update()	boolean	<Override> 라이플 정보를 업데이트	
reload()	void	<Override> 라이플 재장전	

클래스명	Rocket		
긴 사거리, 낮은 연사력, 높은 데미지, 적은 장탄수의 특징을 가진 로켓 클래스			
Super class	Weapon		
interface			
메소드명	리턴타입	내 용	
MachineGun()	생성자	생성자	
spinAngle(boolean pos)	void	<Override> 로켓 발사 각도 조정	
canShot()	boolean	<Override> 로켓이 발사가 가능한 상태인지 확인	
shot()	void	<Override> 로켓 정보를 탄환객체에 주어 탄환생성(발사)	
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 로켓 객체를 그려줌	
update()	boolean	<Override> 로켓 정보를 업데이트	
reload()	void	<Override> 로켓 재장전	

클래스명	Top	
상체 파츠의 조상	클래스, 체력관리, 스킬기능을 가지고 있다.	
Super class	GameObject	
Interface	Hitable	
변수명	데이터타입	내 용
img	BufferedImage[]	상체 이미지를 담은 이미지 배열
weight	int	무게
foward	boolean	객체가 바라보는 방향
skillDelay	int	스킬을 재사용하는데 걸리는 시간
skillStart	int	스킬 대기시간을 재는데 사용되는 변수
maxSkill	int	최대 스킬 사용가능 횟수
curSkill	int	남은 스킬 사용가능 횟수
메소드명	리턴타입	내 용
Top()	생성자	생성자
setDir(boolean foward)	void	객체가 바라보는 방향 설정
genAftObj()	void	<Override> 객체의 수명이 다하면 다음 객체를 생성
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 상체 파츠를 그려줌
update()	boolean	<Override> 상체 파츠의 정보를 업데이트
drawHPBar(Graphics2D g, int cameraX, int cameraY)	void	현재 체력을 체력바로 표시
getDmg()	int	
isCollision(GameObject other)	boolean	<Override> 로봇과 다른 객체와의 충돌여부 반환
isShoot(Projectile other)	void	<Override> 로봇이 총알에 맞으면, 데미지를 입힘
isHeal(HealPack heal)	void	로봇이 힐팩과 충돌하면, 체력을 높여줌
skill()	void	상체 파츠 고유의 스킬을 사용함

클래스명	LightArmor	
힐팩 스킬을 가진	경갑옷 클래스	
Super class	Top	
interface		
메소드명	리턴타입	내 용
LightArmor()		생성자
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 경갑옷 파츠를 그려줌
update()	boolean	<Override> 경갑옷 파츠의 정보를 업데이트
skill()	void	<Override> 경갑옷 고유의 스킬인 힐팩 생성 명령 수행

클래스명	HeavyArmor	
방벽 생성 스킬을 가진	중갑옷 클래스	
Super class	Top	
interface		
메소드명	리턴타입	내 용
HeavyArmor()		생성자
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 중갑옷 파츠를 그려줌
update()	boolean	<Override> 중갑옷 파츠의 정보를 업데이트
skill()	void	<Override> 중갑옷 고유의 스킬인 방벽 생성 명령 수행

클래스명	FlateArmor	
무기 공격력 강화 스킬을 가진	초합금갑옷 클래스	
Super class	Top	
interface		
변수명	데이터타입	내 용
isFlame	boolean	강화 스킬이 발동중인지 체크하는 flag
Weapon	weapon	강화 스킬을 적용시킬 무기 파츠
메소드명	리턴타입	내 용
FlateArmor ()		생성자
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 초합금갑옷 파츠를 그려줌
update()	boolean	<Override> 초합금갑옷 파츠의 정보를 업데이트
skill()	void	<Override> 초합금갑옷 고유의 스킬인 힐팩 생성 명령 수행
setWeapon (Weapon weapon)	void	강화할 대상인 무기 파츠를 설정

클래스명	Bottom	
로봇의 이동기능을 담당하는 하체 파츠의 조상 클래스		
SuperClass	GameObject	
Interface	Dropable	
변수명	데이터타입	내 용
img	BufferedImage[]	하체 파츠의 이미지 배열
foward	boolean	객체가 바라보는 방향
jumpLock	boolean	점프가 가능한 상황인지 체크하는 flag
mvSpeed	int	객체가 움직이는 속도
jumpPower	int	객체가 최대 점프할 수 있는 높이
jumpCur	int	현재 객체가 뛰어오른 높이
vX	int	객체가 x축으로 움직이는 방향 벡터
vY	int	객체가 y축으로 움직이는 방향 벡터
weight	int	무게
maxWeight	int	최대 하중
IDLE	int	상태변환에 사용, 1
MOVE	int	상태변환에 사용, 2
JUMP	int	상태변환에 사용, 3
DIR_JUMP	int	상태변환에 사용, 4
JUPM_DOWN	int	상태변환에 사용, 5
메소드명	리턴타입	내 용
Bottom()	생성자	생성자
getJumpLock()	boolean	jumpLock flag를 반환
setDir(boolean foward)	void	객체가 바라보는 방향을 설정
setStIdle()	void	상태를 IDLE로 바꿈
setStMove()	void	상태를 MOVE로 바꿈
setStJump (boolean dirjump)	void	dirjump값에 따라 DIR_JUMP JUMP상태로 바꿈
setStJumpDown()	void	상태를 JUMP_DOWN로 바꿈
idle()	void	idle 명령 수행
move()	void	move 명령 수행
jump(boolean dirjump)	void	jump, dirjump 명령 수행
jumpDown()	void	<Override> jumpdown 명령 수행
genAftObj()	void	<Override> 객체의 수명이 다하면 다음 객체를 생성
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 하체 파츠를 그려줌
update()	boolean	<Override> 하체 파츠의 정보를 업데이트
isFloating()	boolean	<Override> 객체가 떠있는지 아닌지 체크
getLandingPoint()	int	<Override> 객체가 착지할 y좌표를 얻어온다.

클래스명	Biwalk		
빠른 이동속도, 적은 최대하중 특징을 가진 이족보행 클래스			
Super class	Bottom		
interface			
메소드명	리턴타입	내 용	
Biwalk()	생성자	생성자	
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 이족 보행 파츠를 그려줌	

클래스명	Quad		
느린 이동속도, 많은 최대하중 특징을 가진 사족보행 클래스			
Super class	Bottom		
interface			
메소드명	리턴타입	내 용	
Quad()	생성자	생성자	
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 사족 보행 파츠를 그려줌	

클래스명	Copter		
중간 이동속도, 적은 최대하중, 공중에 떠다니는 특징을 가진 쿵터 클래스			
Super class	Bottom		
interface			
메소드명	리턴타입	내 용	
Copter()	생성자	생성자	
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 쿵터 파츠를 그려줌	
setStMove()	void	<Override> 하체 파츠와는 다른 기능을 수행 변화된 기능이 오버라이딩된 메소드	
setStJump (boolean dirjump)	void		
setStJumpDown()	void		
move()	void		
jumpDown()	void		
update()	void		

클래스명	Projectile	
무기 파츠에서 발사되는 탄환들의 조상클래스		
Super class	GameObject	
Interface	Hitable	
변수명	데이터타입	내 용
radical	double	PI / 180한 값
angle	int	탄환의 발사각도
damage	int	탄환의 데미지
life	int	탄환의 생명주기
range	int	탄환의 사거리
dstX	int	HitScan 방식에서 쓰이는 맞은점의 X
dstY	int	HitScan 방식에서 쓰이는 맞은점의 Y
scan	Line2D	HitScan 방식에서 쓰이는 탄환의 발사 궤적
endPoint	Point2D	HitScan 방식에서 쓰이는 탄환의 끝점
foward	boolean	탄환이 발사되는 방향 flag
preload	int	기관총 탄환에서 쓰이는 3발 데미지를 위한 변수
tail	int	탄환 그리기에 사용되는 변수
c	Colors	탄환 그리기에 사용되는 색
메소드명	리턴타입	내 용
Projectile(int angle, byte team, int damage)	생성자	생성자
genAftObj()	void	<Override> 객체의 수명이 다하면 다음 객체를 생성
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 투사체 객체를 그려줌
update()	boolean	<Override> 투사체 객체의 정보를 업데이트
getDmg()	int	탄환의 데미지 반환
attack()	void	<abstract> 탄환이 때렸을 때 수행되는 메소드

클래스명	Ammo		
짧은 간격으로 충돌이 판정되는 반 HitScan 방식의 탄환			
Super class	Projectile		
interface			
메소드명	리턴타입	내 용	
Ammo(Weapon mother, int id,int range, int angle, int startX, int startY, byte team, int damage, boolean foward, int preload)	생성자	생성자	
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 투사체 객체를 그려줌	
update()	boolean	<Override> 투사체 객체의 정보를 업데이트	
getCollisionPoint(GameObject other)	Point2D	다른 객체와 충돌한 좌표를 반환해줌	
getDistance(Point2D p)	int	p점과의 거리와 startX,Y와의 거리를 반환해줌	
setEndPoint(Point2D p)	Point2D	맞은점을 끝점으로 설정	
attack()	void	<Override> 맞았을 때 수행되는 메소드	
isCollision(GameObject other)	boolean	<Override> 충돌여부를 판단하는 메소드	
isShoot(Projectile other)	void	<Override> 이 객체에선 무기기능 없음	

클래스명	MG Ammo		
한 순간으로 충돌이 판정되는 HitScan 방식의 탄환			
Super class	Projectile		
interface			
메소드명	리턴타입	내 용	
MG Ammo(Weapon mother, int id,int range, int angle, int startX, int startY, byte team, int damage, boolean foward, int preload)	생성자	생성자	
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 투사체 객체를 그려줌	
update()	boolean	<Override> 투사체 객체의 정보를 업데이트	
getCollisionPoint(GameObject other)	Point2D	다른 객체와 충돌한 좌표를 반환해줌	
getDistance(Point2D p)	int	p점과의 거리와 startX,Y와의 거리를 반환해줌	
setEndPoint(Point2D p)	Point2D	맞은점을 끝점으로 설정	
attack()	void	<Override> 맞았을 때 수행되는 메소드	
isCollision(GameObject other)	boolean	<Override> 충돌여부를 판단하는 메소드	
isShoot(Projectile other)	void	<Override> 이 객체에선 무기기능 없음	

클래스명	Missle		
발사하고 날아가면서 충돌을 체크하는 투사체 방식의 탄환			
Super class	Projectile		
interface			
변수명	데이터타입	내 용	
xpos	int	미사일을 그릴 때 사용되는 값(draw)	
ypos	int		
r	int		
r2	int		
r3	int		
stroke	int	미사일을 그리는 선의 굵기	
length	int	미사일의 길이	
메소드명		리턴타입	내 용
Missle(Weapon mother, int id,int range, int angle, int startX, int startY, byte team, int damage, int preload)		생성자	생성자
draw(Graphics2D g, int cameraX, int cameraY)		void	<Override> 미사일 객체를 그려줌
update()		boolean	<Override> 미사일 객체의 정보를 업데이트
attack()		void	<Override> 맞았을 때 수행되는 메소드
isCollision(GameObject other)		boolean	<Override> 충돌여부를 판단하는 메소드
isShoot(Projectile other)		void	<Override> 이 객체에선 아무기능 없음

클래스명	Item	
게임내 존재하는	아이템의 조상 클래스	
Super class	GameObject	
Interface	Hitable, Dropable	
interface		
변수명	데이터타입	내 용
vY	int	y좌표 벡터값
img	BufferedImage[]	이미지 배열
메소드명	리턴타입	내 용
Item(int id,int x, int y)	생성자	생성자
jumpDown()	void	<Override> jumpdown 명령 수행
genAftObj()	void	<Override> 객체의 수명이 다하면 다음 객체를 생성
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 아이템 객체를 그려줌
update()	boolean	<Override> 아이템 객체의 정보를 업데이트
isFloating()	boolean	<Override> 객체가 떠있는지 아닌지 체크
getLandingPoint()	int	<Override> 객체가 착지할 y좌표를 얻어온다.
isCollision(GameObject other)	boolean	<Override> 충돌여부를 판단하는 메소드
isShoot(Projectile other)	void	<Override> 이 객체에선 아무기능 없음
drawHPBar(Graphics2D g, int cameraX, int cameraY)	void	이 객체의 체력바를 표시

클래스명	HealPack	
먹으면 체력을 회복시켜주는 힐팩 클래스		
Super class	Item	
interface		
변수명	데이터타입	내 용
life	int	이 객체의 생명주기
heal	int	회복량
isEatable	boolean	완전히 생성되어 땅에 착지한 후 인지를 체크하는 flag
메소드명		리턴타입
HealPack(int id ,int x, int y)		생성자
draw(Graphics2D g, int cameraX, int cameraY)		void
update()		boolean
isCollision(GameObject other)		boolean
getHeal()		int

클래스명	Barrier	
일정량의 데미지를 대신 맞아주는 방벽 클래스		
Super class	Item	
interface		
변수명	데이터타입	내 용
foward	boolean	객체가 바라보는 방향
메소드명	리턴타입	내 용
public Barrier(byte team, int id, int x, int y, boolean foward)	생성자	생성자
genAftObj()	void	<Override> 객체의 수명이 다하면 다음 객체를 생성
draw(Graphics2D g, int cameraX, int cameraY)	void	<Override> 방벽 객체를 그려줌
update()	boolean	<Override> 방벽 객체의 정보를 업데이트
drawHPBar(Graphics2D g, int cameraX, int cameraY)	void	<Override> 방벽 객체의 체력바를 표시

인터페이스명	Hitable	
충돌을 일으키는 오브젝트를 위한 인터페이스		
메소드명	리턴타입	내 용
isCollision(GameObject other)	boolean	다른 객체와의 충돌을 검사하여 여부를 반환
isShoot(Projectile other)	void	총알 객체에게 맞았을 때 수행된다.

인터페이스명	Dropable	
중력의 영향을 받아 떨어지는 기능이 있는 오브젝트를 위한 클래스		
메소드명	리턴타입	내 용
isFloating()	boolean	객체가 땅을 밟고 있지 않은지 판단한다.
getLandingPoint()	int	객체가 밟을 땅의 y좌표를 반환한다
jumpDown()	void	땅에 떨어지는 기능수행

클래스명	ContainerFactory	
로봇을 생성하기 위한 과정을 비즈니스 로직과 분리하기 위해 팩토리 패턴을 적용한 블랙박스 클래스.		
메소드명	리턴타입	내 용
genContainer(byte team, int x, int y, int id, int wnum, int tnum, int bnum)	Container	로봇의 팀, 좌표, id값, 각 파트 번호를 받아 로봇 객체를 반환함

클래스명	ResourceManager	
리소스를 관리하기	위해 싱글톤 패턴을 적용한 클래스	
변수명	데이터타입	내 용
instance	ResourceManager	한번만 생성하여 사용하기 위한 객체변수
...IMG	BufferedImage[]	각자 필요한 이미지 배열을 변수로 가짐
메소드명	리턴타입	내 용
ResourceManager()	생성자	생성자
getInstance()	ResourceManager	instance를 반환
get...IMG()	BufferedImage[]	객체 내의 이미지 배열을 반환

Server Side

클래스명	Main	
서버 메인 클래스. 게임 채널들의 업데이트를 관여한다.		
Super Class		
interface		
변수명	데이터타입	내 용
roomNum	static int	다음에 생성될 방 번호.
gameChannel	static HashMap<Integer, GameChannel>	인게임 화면을 출력하는 컴포넌트
메소드명		리턴타입
Main		생성자
makeRoom(Client c, String name)	void	새로운 채널을 생성하고 그 채널에 Client 객체를 넣음.
deleteRoom(Client c, GameChannel gch)	void	해당 채널을 제거
joinRoom(Client c, int id)	void	기존의 채널에 파라미터 Client를 참여

클래스명	Server	
서버 싱글톤 클래스. 각 Client 객체들과 Server 본인의 소켓 채널의 입출력을 관리한다.		
Super Class		
interface		
변수명	데이터타입	내 용
instance	Server	싱글톤 객체
selector	Selector	인게임 화면을 출력하는 컴포넌트
servChannel	ServerSocketChannel	서버의 소켓 채널
clntList	ArrayList<Client>	접속중인 Client 객체 리스트
port	int	포트번호
thread	Thread	스레드
dbmgr	DBManager	데이터베이스 관리 클래스
메소드명	리턴타입	내 용
Server	생성자	생성자
init()	void	객체를 초기화하고 기능을 활성화한다.
run()	void	스레드 루프
accept(SelectionKey selectionKey)	void	새로운 연결을 허용하여 Client 객체를 등록

cut(int length, ByteBuffer buffer)	ByteBuffer	버퍼를 잘라 compact해서 반환
encodeString(String str)	ByteBuffer	문자열을 인코딩하여 인코딩된 데이터 길이와 함께 버퍼로 만들어 반환
cutString(ByteBuffer buffer)	String	바이트 버퍼에서 인코딩된 문자열 데이터를 잘라내어 디코딩하고 반환
completeRecv(ByteBuffer buffer)	byte	패킷의 헤더를 디코딩하여 모든 데이터가 들어왔는지 확인

클래스명	Client	
서버에 접속한 유저들을 추상화한 클래스. Server의 스레드에서 소켓 입출력 메시지 패싱을 받고 Main의 스레드에서 본인이 속한 GameChannel의 진행상황에 대한 메시지 패싱을 받는다.		
Super Class		
interface		
변수명	데이터타입	내 용
id	String	접속한 계정 ID
ready	boolean	게임 채널 안에서 준비 상태
socketChannel	SocketChannel	소켓 채널
recvBuffer	ByteBuffer	소켓으로 수신한 데이터를 정리하여 패킷단위로 소비하기 위한 버퍼
ctrl	Remocon	게임 플레이에서 로봇 객체를 컨트롤하는 컨트롤러 객체
mybot	Container	유저가 소유한 3종류의 로봇을 저장하는 배열
myChannel	GameChannel	유저가 속한 채널
index	int	게임 채널에서 유저의 인덱스
team	byte	게임 채널에서 유저의 팀
kill	int	게임 실적 (처치)
death	int	게임 실적 (사망)
deal	int	게임 실적 (피해량)
메소드명	리턴타입	내 용
Client(SocketChannel socketChannel)	생성자	생성자
recv(SelectionKey selectionKey)	void	소켓으로부터 데이터를 수신하여 recvBuffer에 쌓음
recvProc()	void	하나의 패킷이 완성될때마다 처리하는 메소드
castToSelf(ByteBuffer byteBuffer)	void	본인의 소켓채널에게 패킷을 전송
castToChannel(ByteBuffer byteBuffer)	void	내가 속한 게임 채널의 Client 객체들의 소켓 채널에 패킷을 전송
castToAwaiters(ByteBuffer byteBuffer)	void	게임 채널에 속하지 않은 대기자들의 소켓 채널에 패킷을 전송
castToAll(ByteBuffer byteBuffer)	void	모든 Client 객체의 소켓 채널에게 패킷을 전송
이하 메소드 21개는 패킷을 주고받기 위한 메소드이며, 이에 대한 자세한 명세는 Multiplay Analysis의 Packet Graph를 참조하세요!		
reactPak0~5		
sendPak0~4		
syncRoom/Channel		

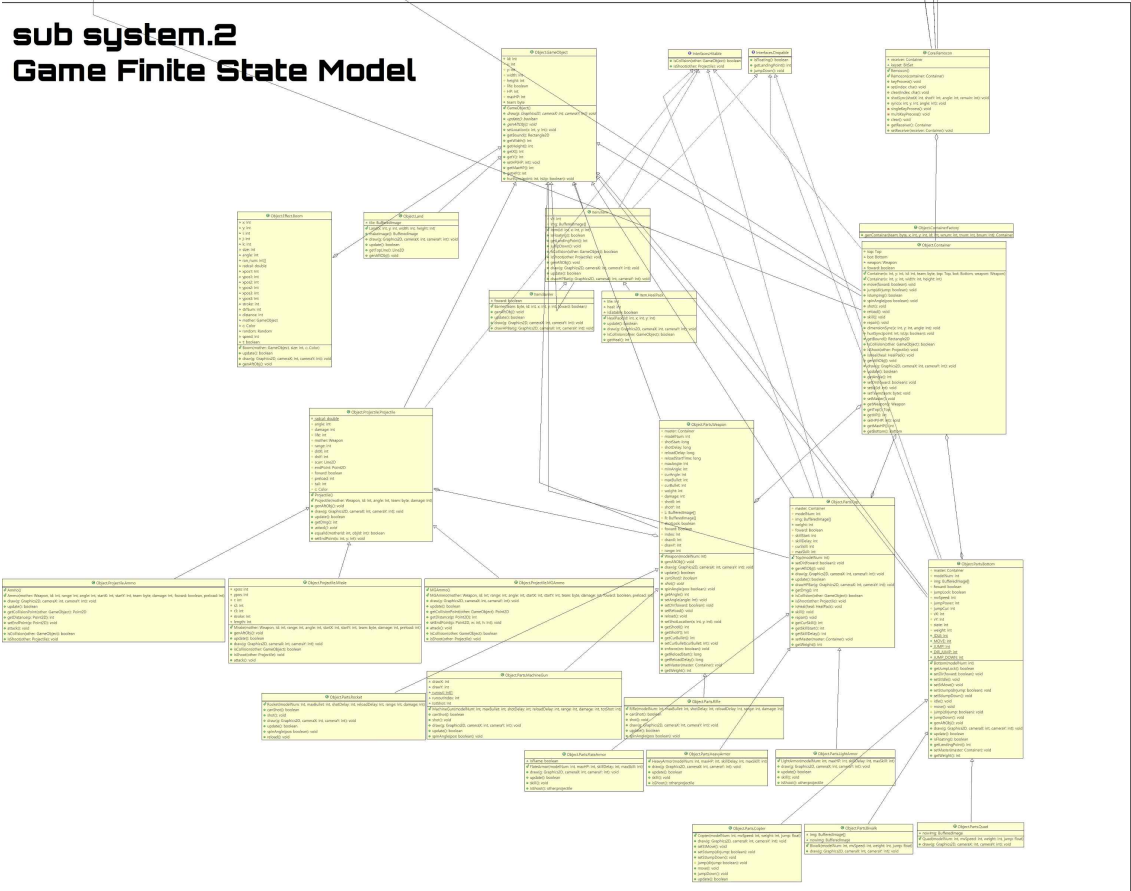
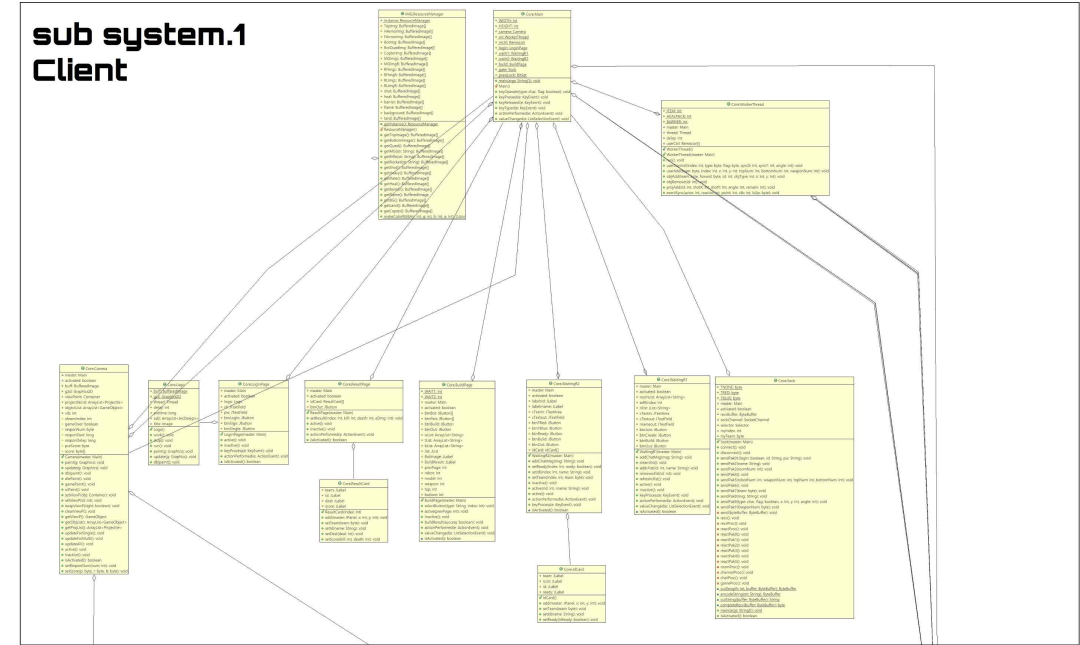
클래스명	DBManager	
데이터 베이스 관리 클래스		
Super Class		
interface		
변수명	데이터타입	내 용
DRIVER	static final String	jdbc 드라이버 패키지 경로
URL	static final String	database url
ID	String	database host
PW	String	database password

SSL	String	SSL
conn	Connection	Connection 객체
stmt	Statement	DB Statement
rs	ResultSet	DB ResultSet
메소드명	리턴타입	내 용
Client(SocketChannel socketChannel)	생성자	생성자
recv(SelectionKey selectionKey)	void	소켓으로부터 데이터를 수신하여 recvBuffer에 쌓음
recvProc()	void	하나의 패킷이 완성될때마다 처리하는 메소드
castToSelf(ByteBuffer byteBuffer)	void	본인의 소켓채널에게 패킷을 전송
castToChannel(ByteBuffer byteBuffer)	void	내가 속한 게임 채널의 Client 객체들의 소켓 채널에 패킷을 전송
castToAwaiters(ByteBuffer byteBuffer)	void	게임 채널에 속하지 않은 대기자들의 소켓 채널에 패킷을 전송
castToAll(ByteBuffer byteBuffer)	void	모든 Client 객체의 소켓 채널에게 패킷을 전송
이하 메소드 21개는 패킷을 주고받기 위한 메소드이며, 이에 대한 자세한 명세는 Multiplay Analysis의 Packet Graph를 참조하세요!		
reactPak0~5		
sendPak0~4		
syncRoom/Channel		

클래스명	GameChannel	
게임 서버로써 기능을 수행하기위한 게임 채널 클래스		
Super Class		
interface		
변수명	데이터타입	내 용
name	String	게임 채널 이름
id	int	게임 채널 번호
teamNum	byte	각 팀 인원수 배열
score	byte	각 팀 스코어 배열
purScore	byte	목표 스코어
userList	Client[]	유저 Client 객체 배열
caster	Client	유저들에게 채널단위 통지용 Client객체
userNum	int	인원수
activated	boolean	게임 활성화 상태
objId	int	게임내 생성되며 누적되는 instance objId값
delay	long	딜레이시간
count	long	카운트시간
elapsed	long	누적시간
objectList	ArrayList<GameObject>	객체리스트
projectileList	ArrayList<Projectile>	투사체리스트
TNONE	static final byte	중립팀
TRED	static final byte	레드팀
TBLUE	static final byte	블루팀
메소드명	리턴타입	내 용
GameChannel(String name, int id)		
gameInit()	void	게임 활성화
gameClose()	void	게임 비활성화
update(long diffTime)	void	update
update_objpf()	ByteBuffer	update object per frame
accountContainer(int I, int robotNum)	ByteBuffer	로봇 상태
accountObj(byte team, int objType, int x, int y, int z)	String	객체 상태
controlKey(int index, byte key)	byte	소켓을 통해 입력받은 클라이언트 측의 컨트롤을 Client 객체로 전달

keyType, int x, int y, int angle, byte flag)		체의 Remocon 객체, 그리고 그 객체와 연결된 Client 객체에 동기화 후 다른 클라이언트들에게 해당 신호를 전달하여 동기화작업을 지시
joinUser(Client c)	boolean	유저 입장 처리
outUser(Client c)	void	유저 퇴장 처리
isTeam(GameObject g, byte team)	boolean	객체 팀 식별
changeTeam(Client c, byte team)	boolean	유저 팀 변경
equalId(int id)	boolean	아이디가 동일한지 확인

최종 클래스 다이어그램



sub system.3
Server

