

# 포트폴리오

이민섭

# MinNetIOCP

GitHub : <https://github.com/dat0613/MinNetIOCP>

사용 언어 : C++

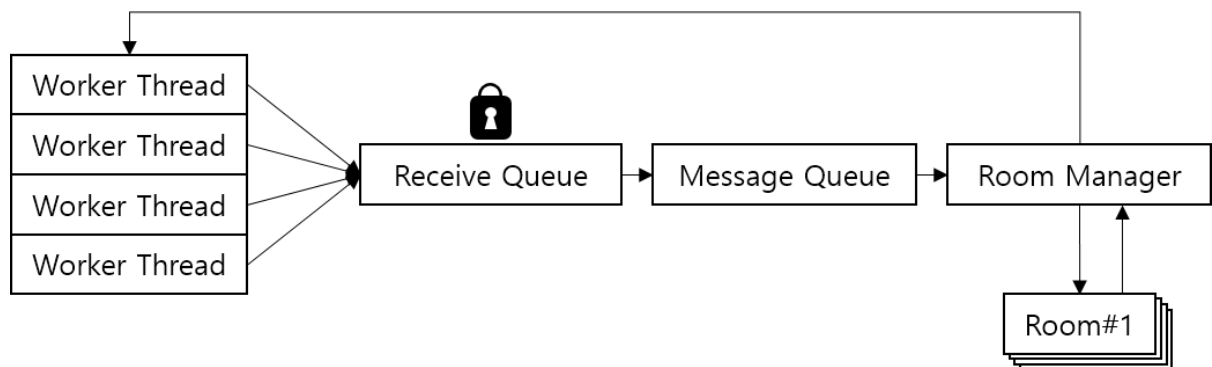
개발 기간 : 2019 / 08 / 21 ~ 개발 중

유니티 엔진과의 연동을 위한 네트워크 라이브러리를 구현했습니다.

기존에 만들어 두었던 라이브러리인 MinNet에 IOCP를 더하여 기능을 확장 시켰습니다.

MinNet은 서버 열기, 접속, 패킷 전송 등의 작업을 간단하게 처리해 주는 자체개발  
네트워크 라이브러리입니다.

MinNetIOCP는 다음과 같은 구조로 구현했습니다.

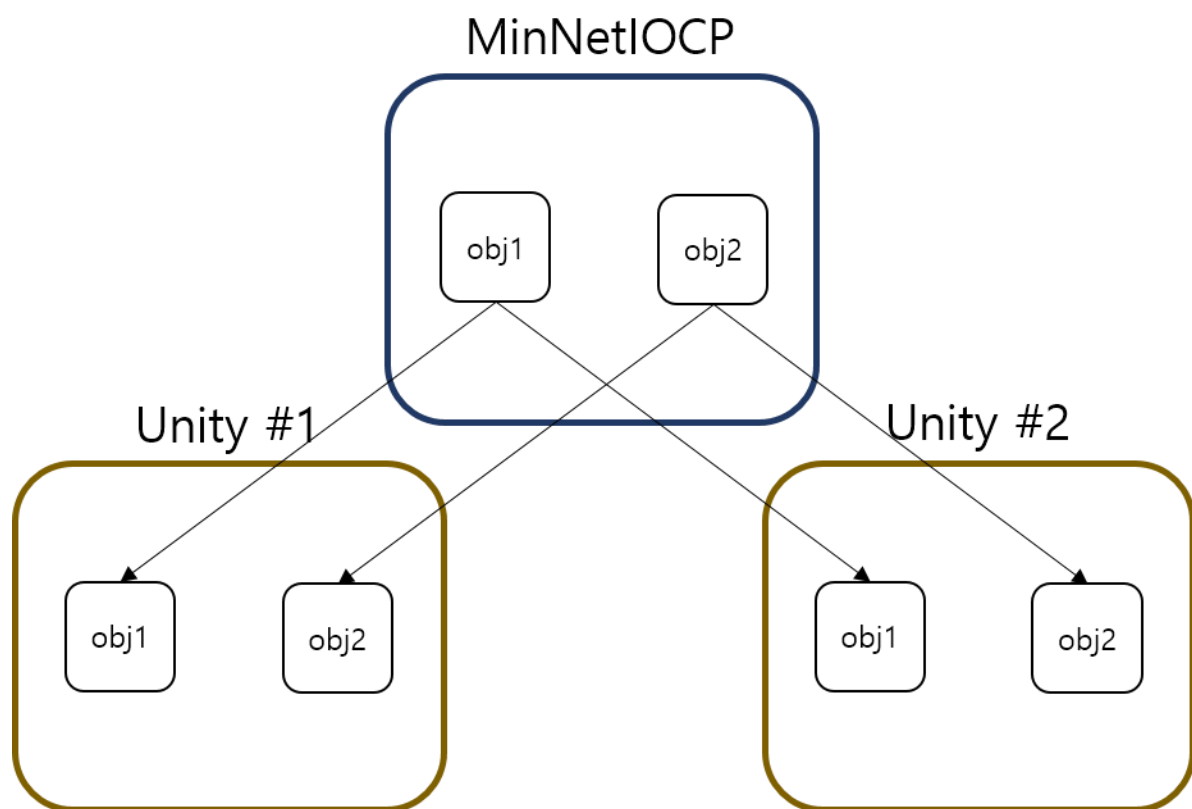


Worker Thread와 Main Thread가 동시에 접근하는 Receive Queue는 스핀락을 사용하여 스레드 세이프 하게 동작하도록 했고, 컨텍스트 스위칭으로 인한 성능저하를 줄였습니다.

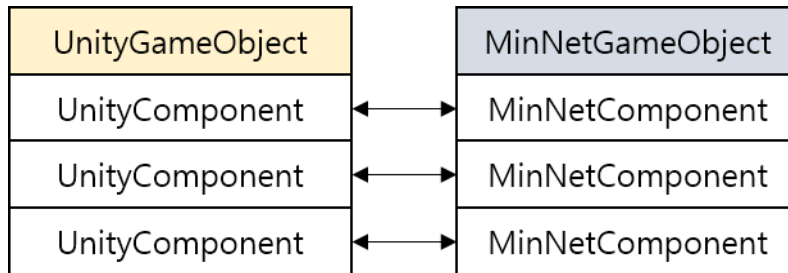
패킷 클래스, overlapped, 클라이언트 정보 등과 같이 생성과 삭제가 빈번한 객체들은 오브젝트 풀을 사용하여 new와 delete를 최소화했습니다.

새로운 클라이언트가 룸에 들어오면 기존 룸에 있던 오브젝트를 자동으로 동기화하고 다양한 콜백 함수를 통해 세부적인 동기화를 진행합니다.

유니티 엔진의 GameObject와 1대1 대응하는 MinNetGameObject 클래스를 통해 더욱 직관적으로 유니티 엔진과 상호작용할 수 있게 구현 했습니다.

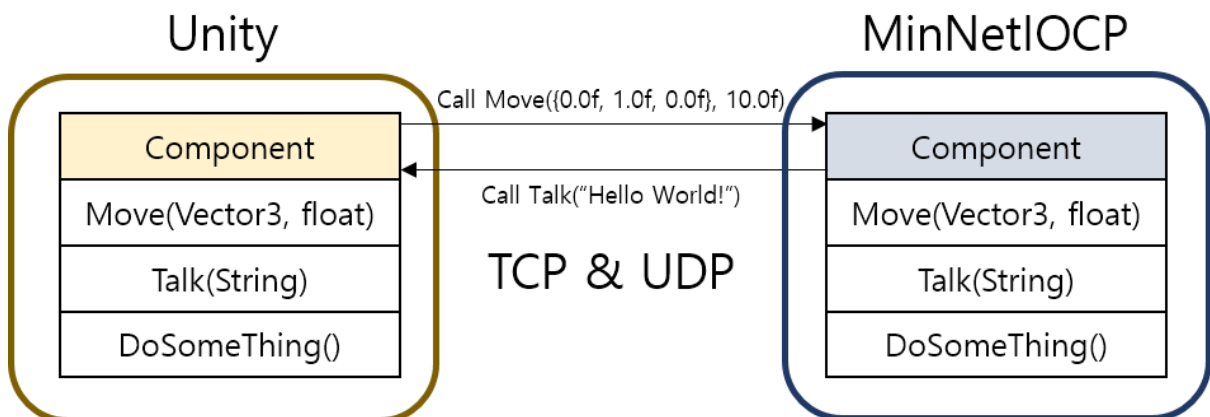


유니티 엔진처럼 컴포넌트 패턴을 도입해 하나의 오브젝트에 여러 개의 컴포넌트가 독립적으로 통신하도록 설계하였습니다.



이러한 설계를 통해 유니티에서 구현한 게임 오브젝트의 구조를 변경하지 않고 서버에 추가할 수 있습니다.

컴포넌트들의 통신에는 RPC(Remote Procedure Call) 를 사용합니다.



RPC는 TCP와 UDP를 모두 지원해 정보의 중요도에 따라 통신 방식을 선택할 수 있습니다.

RPC대상에 대한 다양한 옵션을 제공해 해당 정보가 필요한 대상을 지정하여 통신할 수 있습니다.

C++ 11의 가변 인자 템플릿을 이용해 RPC에 필요한 매개변수를 간단하게 추가할 수 있습니다.

# MinNetforUnity

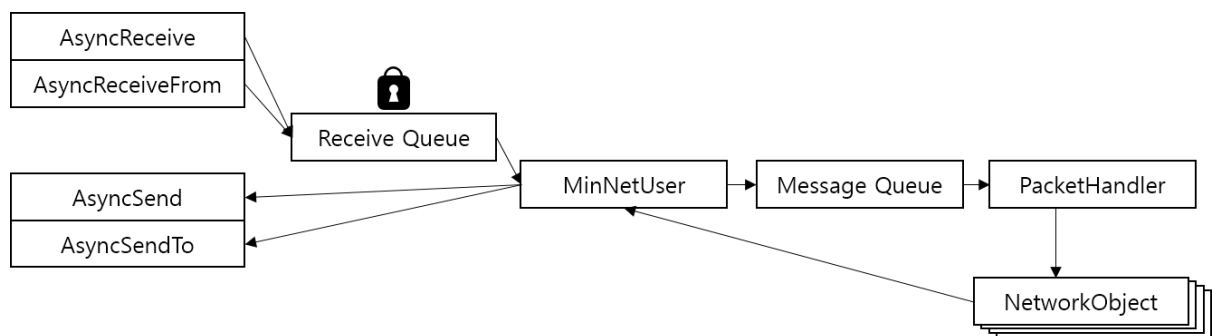
GitHub : <https://github.com/dat0613/MinNetforUnity>

사용 언어 : C# / Unity

개발 기간 : 2019 / 08 / 21 ~ 개발 중

유니티 엔진에서 MinNetIOCP와 통신하기 위해 만든 네트워크 라이브러리입니다.

MinNetforUnity는 다음과 같은 구조로 구현했습니다.



비동기 IO함수를 사용해 게임을 멈추지 않고도 서버와 통신을 할 수 있게끔 했습니다.

비동기 함수에서 바로 유니티 오브젝트에 접근하는 것이 허용되지 않아 **PacketHandler** 오브젝트를 생성하여 패킷을 처리했습니다.

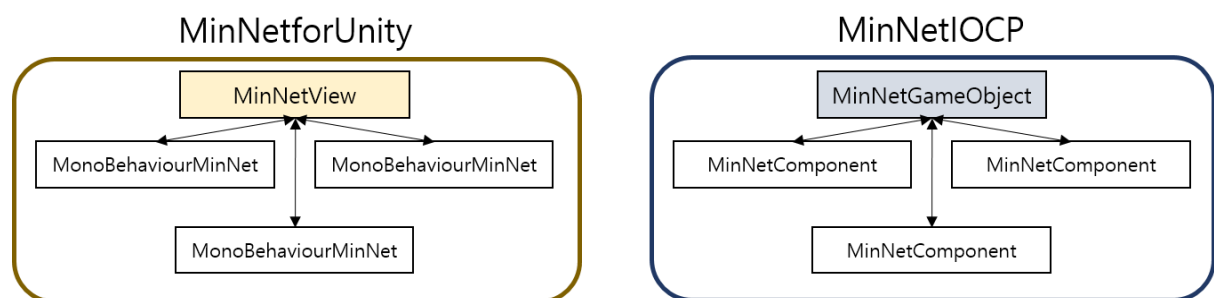
생성이 빈번한 패킷 클래스는 오브젝트 풀을 사용하여 GC로 인한 성능 저하를 최소화했습니다.

유니티의 Resources.Load 함수로 찾은 프리팹과 RPC에 필요한 리플렉션 객체는 캐시를 사용해 미리 저장해 두어 같은 객체를 찾는 작업이 반복되지 않게 만들었습니다.

유니티의 MonoBehaviour 클래스를 상속받는 MonoBehaviourMinNet 클래스를 통해 MinNetIOCP의 MinNetComponent와 통신합니다.

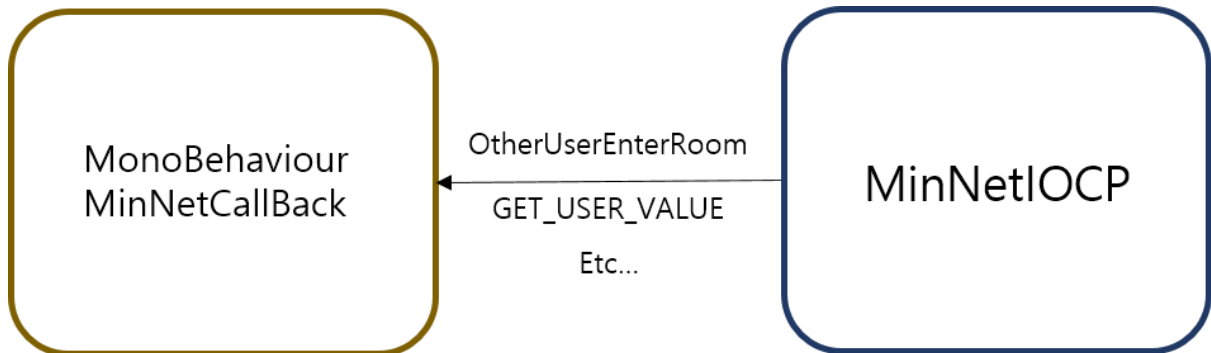
MinNetIOCP의 컴포넌트와 통신할 때는 TCP와 UDP를 모두 지원하는 RPC를 사용합니다.

MonoBehaviourMinNet컴포넌트를 추가할 게임오브젝트 에는 MinNetView 컴포넌트도 함께 추가해야 네트워크 오브젝트로서 동작합니다.

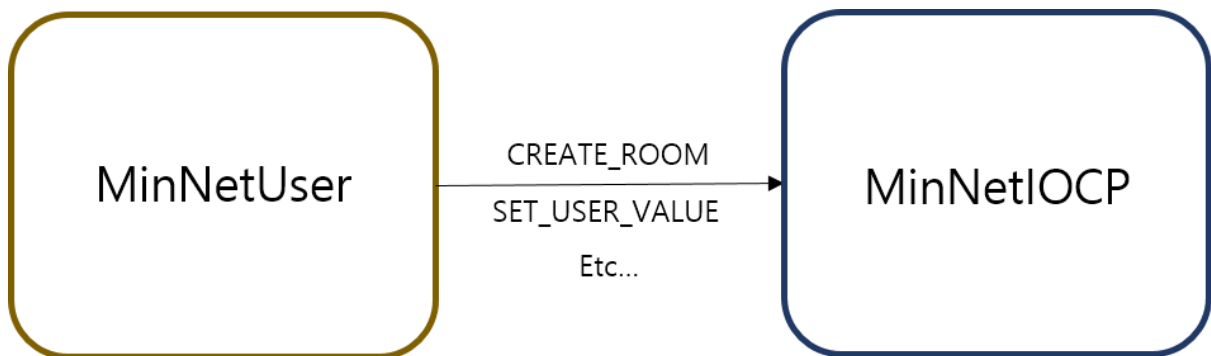


MinNetView와 MinNetGameObject는 네트워크 상에서 1대 1로 대응하는 관계를 가집니다.

서버와 상호작용하며 다양한 콜백함수를 호출하는 MonoBehaviourMinNetCallBack 클래스를 통하여 서버 접속, 방 접속, 다른 클라이언트의 접속 등의 다양한 이벤트를 처리할 수 있습니다.



MinNetUser클래스를 통해 서버에 접속, 접속 끊기, 게임 방 생성 등의 기본적인 동작들을 구사할 수 있습니다.



MinNetUser 클래스를 통한 동작의 결과는 MonoBehaviourMinNetCallBack 클래스가 받아 콜백 함수를 호출 합니다.

# IOCP TPS 프로젝트

---

Client GitHub : <https://github.com/dat0613/UnityNetworkGame>

Server GitHub : <https://github.com/dat0613/UnityNetworkGameServer>

YouTube : <https://www.youtube.com/watch?v=0U6TDQt4SBQ>

사용 언어 : C++ / C# / Unity

개발 기간 : 2019 / 10 / 28 ~ 2019 / 12 / 11

---

1인 개발 작품입니다.

MinNetIOCP와 MinNetforUnity를 사용하여 만든 네트워크 게임입니다.

최대 12명의 플레이어가 6 VS 6으로 전투를 벌이고 상대방의 남은 리스폰 카운트를 0으로 만들면 승리하는 TPS 게임 입니다.

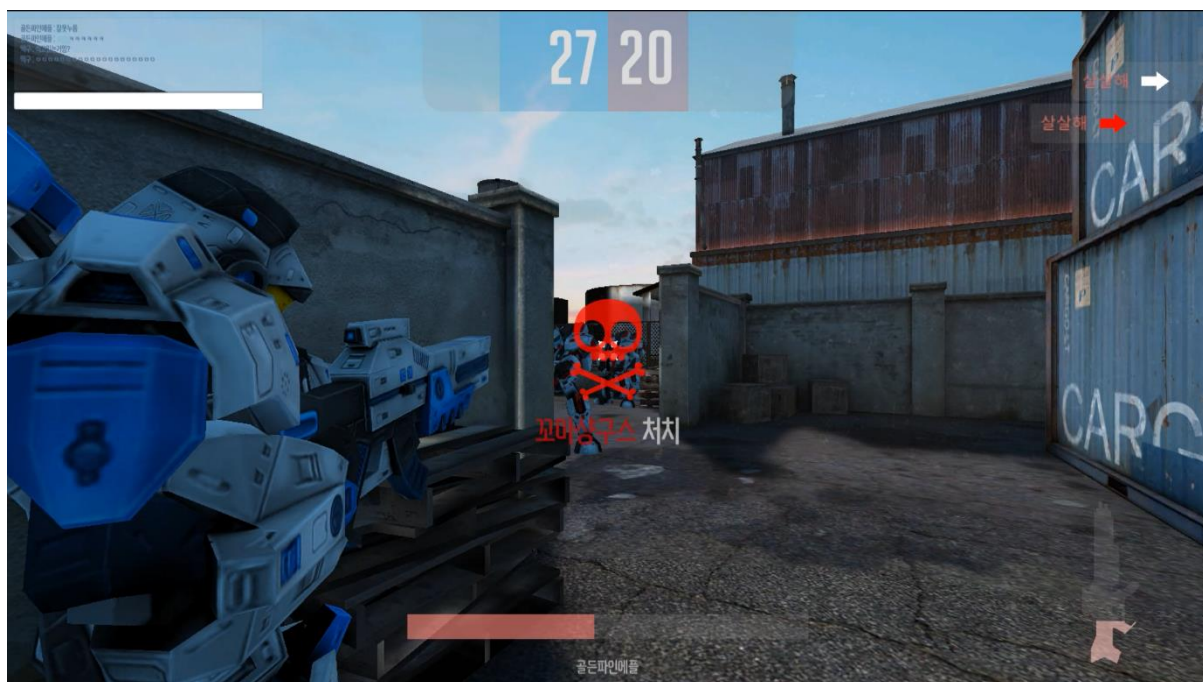
플레이어가 게임 방을 열고 게임 커스터마이징 할 수 있게 구현하여 게임의 자유도를 높였습니다.

게임 탐색기를 통해 플레이어가 원하는 방을 골라 플레이 할 수 있습니다.

플레이어의 사격, 피격, 사망, 리스폰 등과 같이 변조되면 안될 중요한 정보의 처리는 모두 서버가 담당 합니다.



## 인게임 스크린샷



# TCP/IP 2D 슈팅게임

---

GitHub : <https://github.com/dat0613/RunNGun>

YouTube : <https://youtu.be/VRXrBzVfQWQ>

사용 언어 : C++ / DirectX 9

개발 기간 : 2018 / 11 / 25 ~ 2018 / 12 / 25

---

5인 개발 작품입니다.

서버와 클라이언트의 모든 프로그래밍 파트를 담당했습니다.

여러 명의 플레이어가 필드에 놓여진 상자를 열어 무작위 무기를 얻고 다른 플레이어를 사격해 맵 밖으로 밀어내 생존하는 게임입니다.

기존에 만들어 두었던 MinNet라이브러리를 이용해 간단하게 게임을 만들 수 있었습니다.

날아가는 총알의 정교한 판정을 위해 OBB충돌을 구현하였고 공정한 피격 판정을 위해 사격한 플레이어와 피격된 플레이어의 결과가 같을 때 만 판정이 되도록 했습니다.

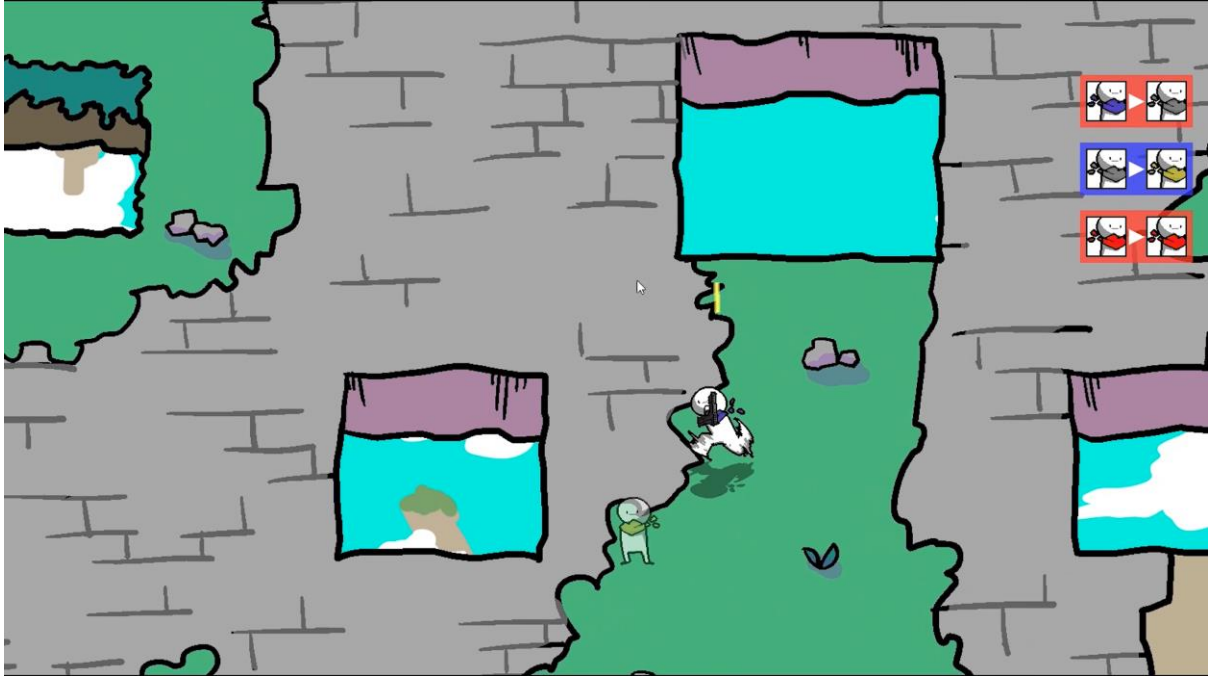
select를 사용하여 다수의 클라이언트를 처리하였습니다.

다수의 인원이 참여하는 팀 프로젝트에서 팀장과 메인 프로그래머 역할을 하며 새로운 경험을 쌓을 수 있는 프로젝트였습니다.

이 프로젝트는 전공선생님께 인정받아 2학년이지만 3학년 선배님들이 졸업작품을 발표하는 연말 행사인 전공 시사회에서 작품을 발표할 수 있었습니다.

## 인게임 스크린샷

---



여기까지 읽어주셔서 감사합니다!

혹시 더 궁금하신 내용이 있으시면

010-3376-1152, [dat0613@naver.com](mailto:dat0613@naver.com) 으로

연락 부탁드립니다!