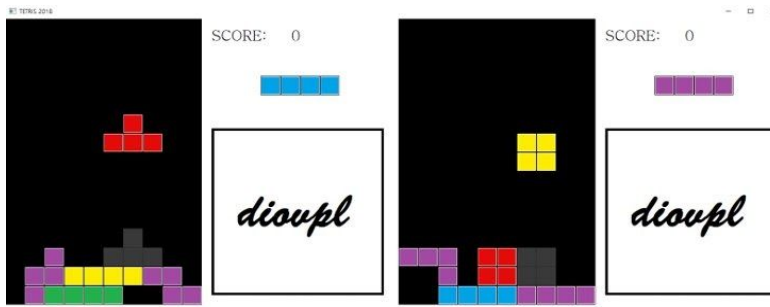


## 오픈소스와 소켓을 이용한 2인 테트리스 게임



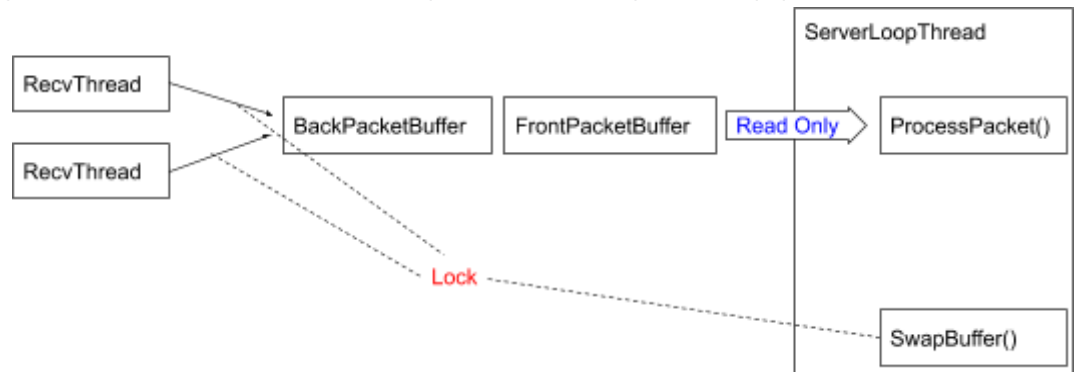
- 구현 목표 : 선착순으로 대기 중인 클라이언트 2명을 연결해 함께 테트리스 게임을 하게 하는 서버를 구현한다.
  - 개발 언어 : C++ 17
  - 서버 : IOCP 서버
  - 클라이언트 : 오픈소스 싱글 테트리스 게임 수정(WinAPI)  
(<https://github.com/diovgl248/cpp-tetris>)

- 클라이언트 : 클라이언트들의 입력을 공유하여 화면을 동기화한다.

```
g_GameScene.Input(); /* 유저 input 서버로 send() */
if(receivedPacket.UserNumber != g_MainPlayerUserNumber) {
    g_OtherGameScene.Input(receivedPacket); /* 서버 경유된 다른 유저 input */
}

g_GameScene.Update();
g_OtherGameScene.Update();
```

- IOCP 서버 : Back Buffer와 Front Buffer를 이용하여 Lock 없이 패킷을 처리한다.



- 문제점 : 클라이언트 1은 블록 이동 후 움직이지 않는 시간 이내에 key up을 하지만 네트워크 딜레이로 인하여 클라이언트 2에서는 블록 이동 후 움직이지 않는 시간 이후에 key up이 되어 클라이언트 2에서 블록이 한 칸 더 이동하는 경우가 생겼다.

- 문제를 수정하기 위하여 key와 블록의 위치를 같이 전송하여 움직이는 블록의 위치를 수정한다.
- 1.로 인해 블록이 한 칸 이동후 다시 원위치로 돌려지는 것이 그대로 보여 유저가 보기에 좋지 않다.
- 출동 후에는 움직이는 블록이 아니므로 수정할 수 없다.

- 다음 프로젝트에서 참고 사항

- key를 전송하여 동기화하는 것보다 게임 요소의 상태를 전송하여 동기화하는 것이 더 좋을 것 같다.(예 : 블록의 위치, 타입, 이벤트 등)
- Packet의 Data 부분은 직접 하나하나 포인터와 형 변환으로 저장하는 것보다 따로 정한 구조체로 저장하는 것이 수정 용이성, 가독성, 실수 방지에 더 좋다.
- Packet Buffer를 std::vector가 아닌 정적으로 크게 사용하여 write할 포인터를 얻을 때만 Lock하고 메모리 복사는 Lock없이 할 것.
- 3.을 할 때 false sharing을 방지할 것.