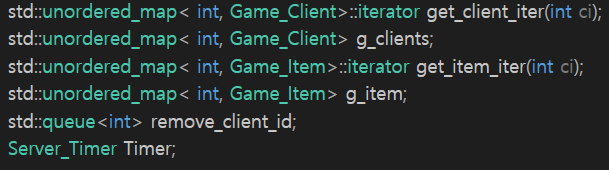
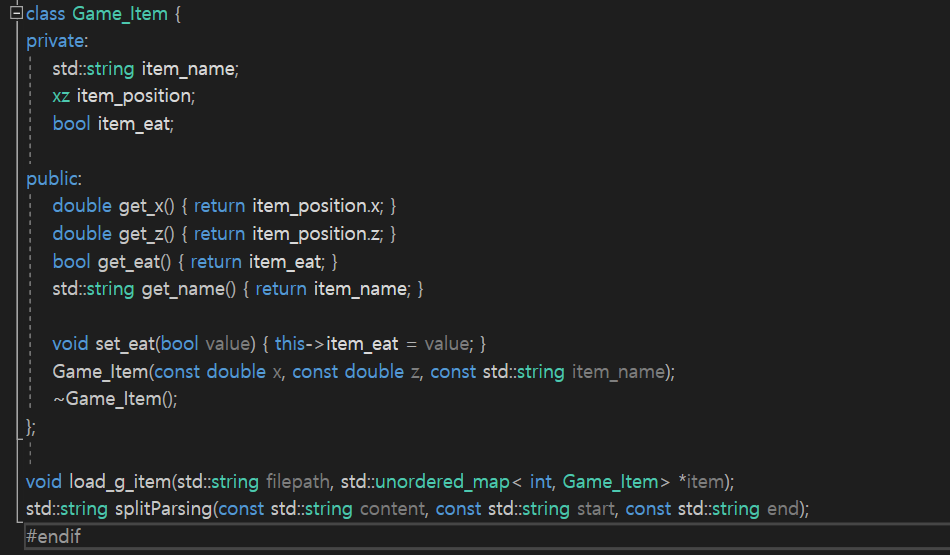
04월 04일 ~ 08일 작업일지:

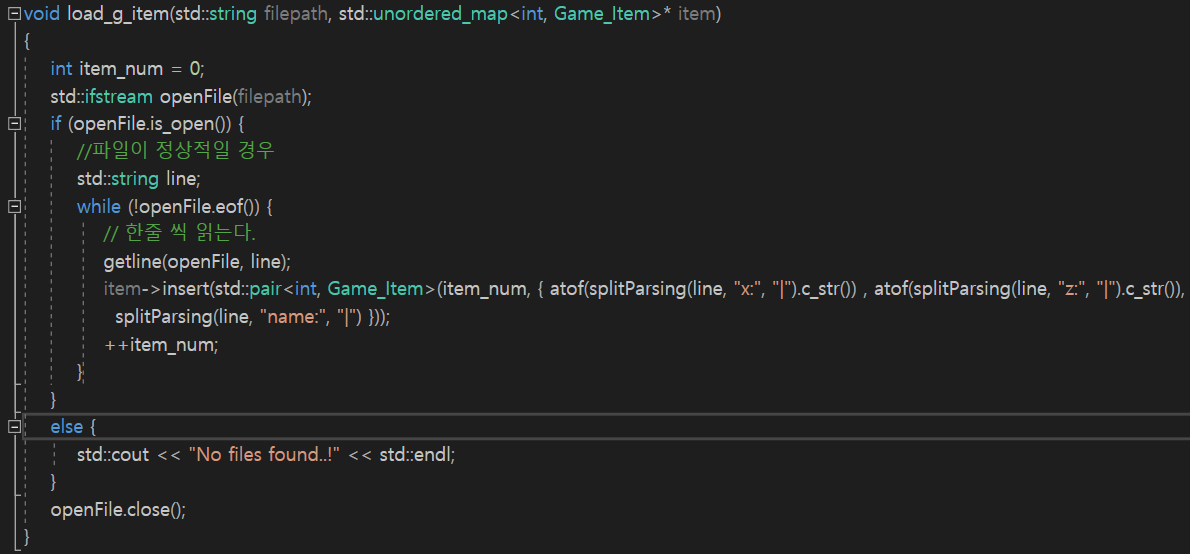
**서버에서 클라이언트 고유번호를 Stack 에서 Queue로 변경**

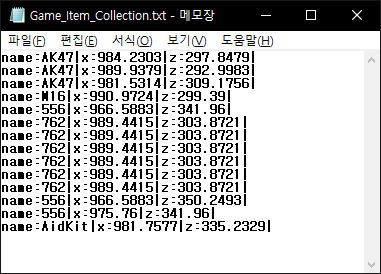


* **기존 get\_person 으로 고유번호를 주는 것을 Queue로 변환하여 클라이언트 고유번호 부여시 에 대한 오버헤드 최소화.**

**서버에서 클라이언트 에게 아이템을 뿌려 주기 위한 아이템 데이터 및 클래스 작성**

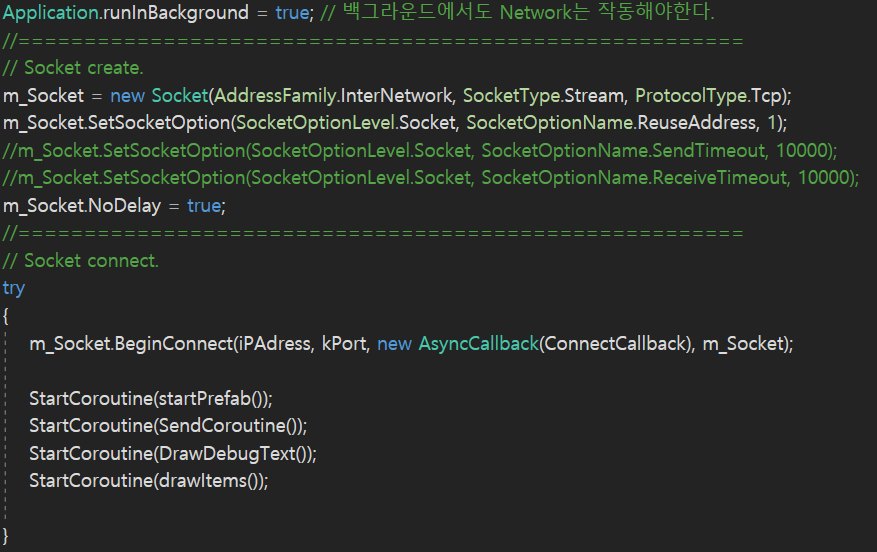


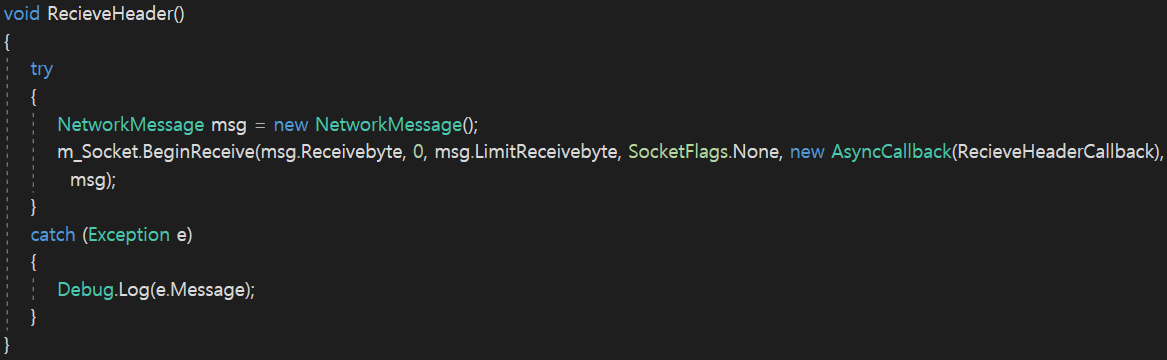




* **txt 파일을 서버에서 읽은 뒤 서버에서는 클라이언트 들에게 아이템이 여기 있다고 전송을 하여 아이템이 생성되게 작업을 완료.**
* **다른 클라이언트가 해당 아이템을 먹을 시 다른 클라이언트에서 먹지 못한 아이템은 자동으로 사라지게 작업 완료.**
* **서버에서 txt 파일을 파싱 하기 위하여, load\_g\_item 함수를 제작, txt 파일을 한 줄 씩 읽어서 Item\_Class에 넣어준다.**

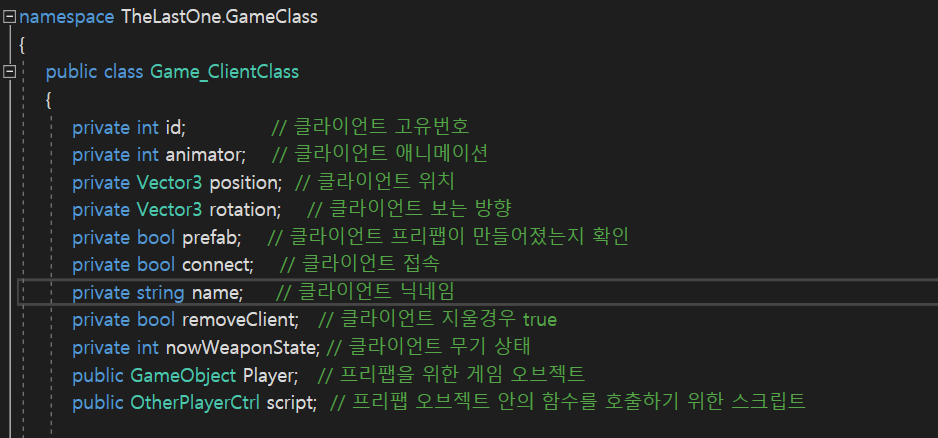
**클라이언트 Socket Recv 방식 수정**

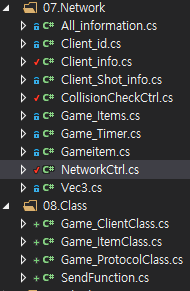
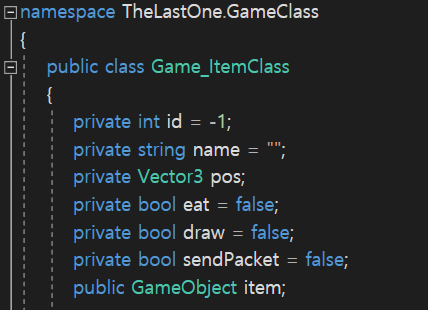


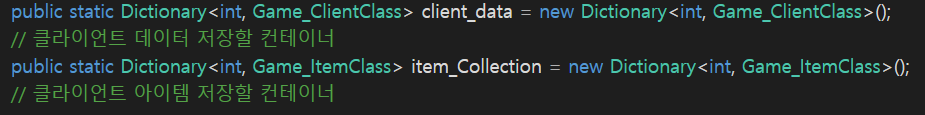


* **기존 방식의 경우 코루틴을 이용하여 계속 Recv를 받게 하였지만, 현재 방식은 Recv가 이루어 진 이후 다시 Recv를 할 수 있게 작업을 하였다.**
* **기존 패킷 처리 방식의 경우 보낸 패킷과 실제 사이즈가 맞을 경우에만 프로세스를 해주게 하였지만, 변경된 방식은 기존 패킷 사이즈와 실제 패킷 사이즈가 다를 경우 다시 한번 Recv를 하게 만들거나, 오버 사이즈로 받아 왔을 경우 패킷을 잘라서 프로세스 하게 만들었다.**

**클라이언트 Client, Item 클래스로 변경 및 Dictionary컨테이너 사용**







* **클라이언트 데이터를 기존 배열로 관리 할 경우 서버에서 최대 인원을 수정할 경우 클라이언트도 똑같이 최대인원을 수정해야하는 번거로움을 수정**
* **배열 에서의 for문을 통한 찾는 시간보다 Dictionary를 사용하여 찾는 시간이 더 빠르므로 찾는 부분에 오버헤드를 최소화.**
* **Class를 사용하여 기존 NetworkCtrl 에 무수히 많은 struct 코드들을 파일 분할 하여 깔끔하게 정리.**