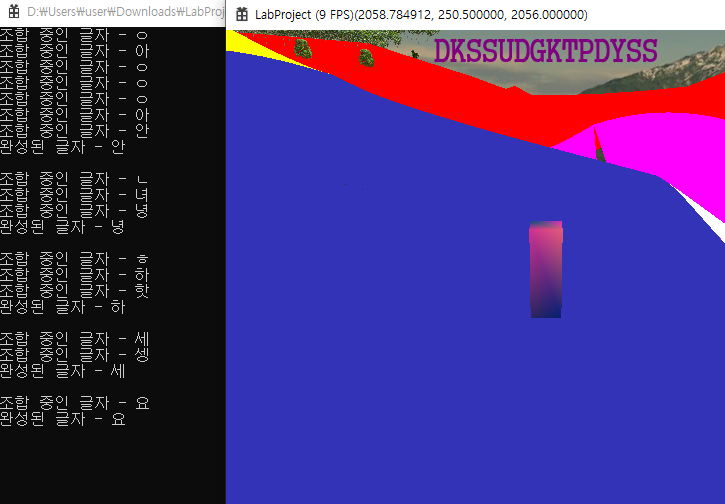
1. 한/영 Text 조합을 위하기 전 문자 버퍼를 분리하여 저장

텍스트, 실내, 오렌지, 가까이이(가) 표시된 사진

자동 생성된 설명

한글 조합중인 버퍼, 한글 조합이 완성된 버퍼, 영문 버퍼를 일단 따로 두어 저장하였고,

이를 적절히 합쳐 채팅처럼 보이는 알고리즘을 생각해야 한다.



* 버퍼 확인을 위한 실행 화면

이를 응용하여 같은 문자열 컨테이너 안에 한/영키 상태에 따라 적절히 붙여야 한다.

1. 클라이언트 프레임 워크 MGR 추가

기존의 클라이언트는 GameScene 클래스에서 static GameScene\* 으로 현재 관리하는 MainScene을 나타냈다. 이제는 서버로부터 받아오는 데이터를 Scene에 반영하는 등의 일도 수행할 필요가 있기 때문에 싱글톤 패턴의 관리자 ‘SceneMGR’ 클래스에서 Scene에 반영해야 하는 일들을 Tick() 함수로 묶어 처리한다.



(개인 작품) static Scene\* -> (졸업 작품)static GameScene\*

텍스트이(가) 표시된 사진

자동 생성된 설명

한 프레임마다 Tick() 함수 호출

텍스트이(가) 표시된 사진

자동 생성된 설명

GameFramework 생성때 싱글톤 패턴의 MGR 클래스 start() 수행



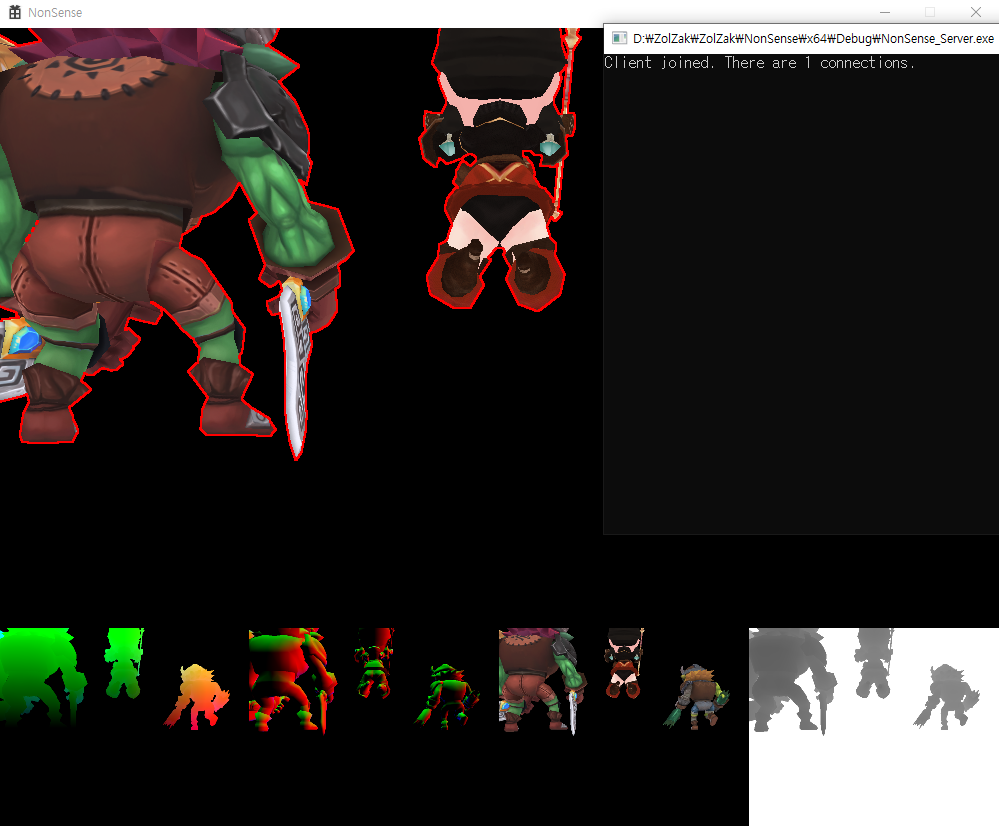


GameFramework::BuildObjects 수행시 GameScene 하나 생성과 SceneMGR에 setScene()

1. 서버와 클라이언트 연결

클라이언트의 디퍼드 렌더링, 외곽선, 애니메이션이 결합 됨에 따라 기존에 콘솔 프로젝트에서 이용하던 서버를 이용하여 본 게임의 서버를 작성했다.

지금은 가변 길이 패킷, 패킷 재조립 등의 작업도 수행해야 해서 데이터 송수신은 제외시켜 놓았고, 서버와 클라이언트의 연결만 확인되는 모습이다.



다음 작업은 가변 패킷, 패킷 재조립을 해결한 뒤, 클라이언트의 입력에 따라 서버에서 처리하고 다시 클라이언트로 결과를 보내는 데이터 송수신을 본격적으로 해야 한다.