2. 개발 내용

2.5 게임의 특성

2.5.1 게임의 특성을 고려한 중점 연구 과제 설정

던전형 4인 레이드 게임으로 iocp 서버의 장점을 살리기 위해서 하나의 게임 서버에서 동시에 여러 개의 방이 돌아가는 구조를 고려하여서 게임 서버 구조를 기획하였다. 또한 속도감 있는 액션 게임을 지향하였고 일반 npc와 보스 npc와의 전투를 게임의 기본 흐름으로 두었기 때문에 대부분의 충돌을 섬세하지 않아도 되면서 동시에 빠르게 감지되고 처리되어야 한다고 생각해서 클라이언트에서 감지하고 서버에 패킷을 전송하면 서버가 이를 검증하는 방식을 채택하였다.

또한 동시에 서버 개발자가 아니더라도 서버에서 관리하는 여러 파라미터들에 쉽게 관여하여 게임의 전반적인 밸런스 조절 및 초기 단계의 스테이지 기획이 원활하게 끔 설계하기 위해서 스크립트 언어를 사용하기로 결정하였다.

위의 내용을 중점 연구 과제로 설정하여서 게임 개발을 진행하였다.

2.5.2 개발 진행 과정

하나의 게임 서버에서 여러 개의 방이 돌아가는 것을 목표로 가장 작은 단위를 방으로 설정하여 개발을 진행하려 하였으나 초기 플레이어의 움직임 동기화를 최초 목적으로 개발을 진행하다 보니 코드의 최소 단위를 방에서 플레이어로 변경하여서 프로그램을 개발하였고 이에 따라서 각각의 플레이어를 배열로 관리하는 방향으로 전환하게 되었다.

프로그램 개발 방향이 방 단위에서 유저 단위로 변경됨에 따라서 플레이어를 하나의 클래스로 정의하고 이에 필요한 변수들을 설정한 뒤에 동기화 작업을 진행하였다. 이때 패킷의 구조들이 설계되었고 이를 통해서 동기화를 진행하였다.

플레이어 동기화가 이뤄진 다음 npc를 생성하고 동기화 하는 것을 목표로 진행하였고 npc 역시 하나의 더미 플레이어 객체로 생각하여 유저와 같은 클래스로 설정한 뒤 세부 파라미터를 변경하는 것으로 npc 객체를 설정하였다. 이에 따라 빠른 npc 동기화가 이뤄졌다.

이후 스크립트 언어를 사용하기 위해서 Lua 스크립트를 서버에 탑재하였고 이를 주로 보스 npc의 패턴 구현에 사용하기 위해서 보스 패턴에 대한 스크립트를 우선적을 제작하였다.

또한 플레이어 단위로 이루어진 게임 서버의 안전성과 더 많은 유저의 수용을 위해서 로그인 과정을 담당하는 로비 서버를 분리하여서 개발 진행하였다.

위의 과정으로 mo형식 보다는 mmo 형식에 가까운 게임 서버를 개발했고 몬스터 및 플레이어의 여러 수치를 스크립트로 변경 가능하게끔 프로그램을 개발했고 로비 서버와 게임 서버 두 개의 서버로 이루어진 서버를 제작하였다. 또한 스크립트를 이용하여 서버가 라이브 상태일때도 여러 게임 내 요소들을 수정 할 수 있게 설정하였다.