

오토 배틀러 게임 설계에 관한 연구

박세훈¹, 권민수¹, 임재현¹, 이상현¹, 류지수², 김동주³

¹대구가톨릭대학교 컴퓨터소프트웨어학부 학부생 ²(주)드림아이디어소프트 이사

³대구가톨릭대학교 컴퓨터소프트웨어학부 교수

pane7979@naver.com, kd0615@naver.com, aminez4466@gmail.com,

tkdgus2282@naver.com, deekim@cu.ac.kr

A Study on Design for Auto Battler game System Development

Sehoon Park¹, Minsu Gwon¹, Jaehyeon Lim¹, Sanghyeon Lee¹, Dongju Kim³

¹School of Computer Software, Daegu Catholic University

²DreamideasoftCo. Ltd.

³School of Computer Software, Daegu Catholic University

요 약

오토배틀러 게임은 타워 디펜스 장르에 PVP 요소를 랜덤 이동과 랜덤 전투 요소를 결합한 게임 장르이다. 본 논문에서는 유니티 엔진을 활용하여 오토배틀러 장르의 게임을 어떻게 설계했는지에 관한 연구이다.

1. 서론

도타(Dota)의 유즈맵(UseMap)으로 시작한 오토 체스(AutoChess)가 오토체스 라이크(AutoChessLike)라는 하나의 장르로 탄생하고 다양한 오토체스 장르 앱스토어나 플레이스토어에서 볼 수 있다.



(그림1 라이엇 게임즈의 TFT게임 플레이화면)

하지만 기존의 오토배틀 장르의 게임들은 공통적으로 상위권의 성적을 기록하려면 최소 20분 길면 30분까지의 시간이 걸린다. 또한 조합 효과 같은 장기간 직접 해보지 않으면 외우기 힘들고 직관적이지 않다는 단점이 있었다. 이러한 점들을 해소하고자 플레이타임을 최대 10분으로 설계함과 동시에 진입장벽을 낮추고 새로운 형식의 오토 장르의 게임을 개발하고 설계한다.

본 논문에서는 유니티 엔진으로 C# 언어를 이용해 모바일 플랫폼 기반의 게임의 설계를 제안하고, Mirror 엔진과 AWS EC2 서비스를 사용하여 멀티플레이 게임을 구현한다.

본 논문의 구성은 2장에서는 관련 연구 및 연구의 필요성을 설명하고, 3장에서는 게임의 기획 및 설계에 대해 다룬다. 마지막으로 4장에서는 논문의 결론을 짓는다.

2. 관련연구 및 연구의 필요성

(1) 관련 기술 설명

Mirror 엔진은 Unity 기반 오픈소스 네트워킹 엔진으로, 멀티플레이를 쉽게 구현하고 관리할 수 있도록 도와주는 프레임워크이다.

AWS(Amazon Web Services)는 가상 컴퓨팅 자원을 서비스 해주는 플랫폼이다. AWS는 다양한 클라우드 기반 서비스를 제공하여 기업이나 개발자가 사용량에 따른 요금을 청구하여 물리적 컴퓨팅 자원을 유연하게 조절 할 수 있도록 한다. AWS EC2(Amazon Elastic Computer Cloud) 가상 서버 서비스와 RDS(Amazon Relational Database Service) 서비스를 사용하여 가상 서버 공간에서 모든 연산 처리를 다루도록 한다.

(2) 연구의 필요성

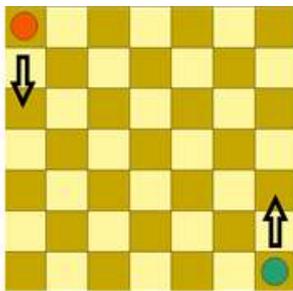
이 연구를 통해 서버와 클라이언트를 분리하여 클라이언트에서 데이터를 변질시키는 행위를 막을 수 있도록 하고 서버에서 모든 연산을 담당하는 만큼 로직과 데이터베이스를 효율적으로 설계 함으로써 서버를 이해할 수 있다. 게임을 개발하는 전반적인 흐름과 객체 관리 기술 및 트러블슈팅을 통해 게임의 서버와 클라이언트 간의 코드 작성 기법을 학습 및 연구할 수 있다. 또한 출시 이후, 사용자 피드백과 함께 향후 프로그램 개발 중 생각해 볼 수 있는 기초 자료가 될 수 있다.

3. 게임의 기획 및 설계

3.1 게임 기획

본 논문에서 기획한 게임은 기존의 오토배틀게임과는 달리 플레이 타임을 대폭 줄이는데 초점을 맞추었다. 동시에 게임이 각각의 강화 아이템 효과를 명확하게 함으로써 자신이 얼마나 강해졌는지 알기 쉽게 한다.

게임 진행 방식은 아래의 그림을 참고하여 설명한다.



(그림2 게임 진행 방식)

게임이 시작하면 가장 기본이 되는 유닛 한 개와 기초 골드가 부여된다. 골드로 상점에서 다양한 강화 아이템을 구매하여 기본 유닛에게 공격, 이동, 특수 아이템을 장착 시켜준다.

유닛들을 전장에 배치하여 배치 턴이 끝나면 적과 아군 유닛이 하나씩 선택되어 번갈아가며 전투 유닛에 장착되어 있는 아이템의 효과에 따라 이동, 공격을 반복한다. 각 유닛들이 체스판의 끝에 도달하면 유닛의 출발지점에서 바로 왼쪽 한칸에서 재출발하는 형식으로 적 유닛을 찾는다. 그러다가 유닛의 공격거리 내 적대적 유닛이 없을 때까지 공격을 시행한다. 더 이상의 적대적 유닛이 존재하지 않을 경우 해당 경기에서 승리하고 상대 플레이어에게 피해를 줄 수 있다. 플레이어 체력은 해당 게임에서의

본 체력으로써 해당 체력을 전부 소진시 게임에서 탈락하게 된다.

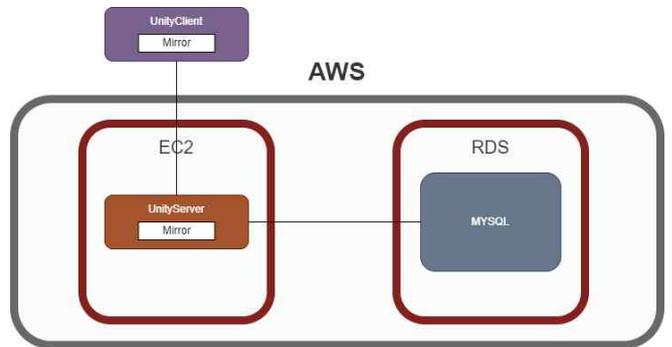
3.2 게임 설계

서비스 구성은 크게 클라이언트와 서버로 구성되며 클라이언트에서 서버로 정보 조회를 요청하고 서버로 올바르게 요청이 들어오면 RDS로 데이터를 요청한다음 해당 클라이언트로 응답해주도록 설계한다. 그림3의 서버와 클라이언트 구성도를 보면 클라이언트와 AWS의 가상서버 서비스와 데이터베이스 서비스를 사용하는 구성임을 볼 수 있다.

(그림3 게임서버 구성도)

4. 결론

이 논문은 유니티 엔진과 C# 언어를 사용하여 오토



배틀 장르의 게임을 설계한 내용이다. 게임의 핵심은 기존의 오토배틀 장르의 재미는 그대로 유지되 플레이타임을 최대 10분안에 끝나도록 설계하여 현대인들의 스낵타임에 맞게 설계하고 멀티플레이어 게임을 위한 기술적인 측면, 특히 AWS 클라우드 서비스와 미러 네트워킹 프레임을 사용하여 격리형 서버(dedicated server)구현에도 중점을 두었다.

5. 향후 연구

본 논문은 과학기술정보통신부 및 정보통신기획평가원에서 주관하여 진행하는 ‘SW중심대학사업’의 결과물입니다.(2019-0-01056)