

지능형 개발 스튜디오 환경에서의 온라인 게임 제작

김동완*, 권지안*, 박용성**, 윤찬현*
 *KAIST 전기전자공학과
 **KAIST GRID 미들웨어연구센터
 e-mail : wannyk@kaist.ac.kr

Making of Online Game in Intelligent Development Studio

Dong-Wan Kim*, Ji-An Kwon*, Yong-Sung Park**, Chan-Hyun Youn*
 *Department of Electrical Engineering, KAIST
 **Grid Middleware Research Center, KAIST

요 약

지능형 응용 개발 스튜디오의 어플리케이션 디자이너와 블록 에디터를 활용하면, 개발 언어에 대한 지식이 없거나 개발 경험이 없는 사람도 코드 작성 없이 간단히 Drag&Drop 으로 기능을 개발할 수 있다. 본 연구에서는 지능형 스튜디오 환경에서 간단한 온라인 게임을 시험적으로 제작하고 평가한다.

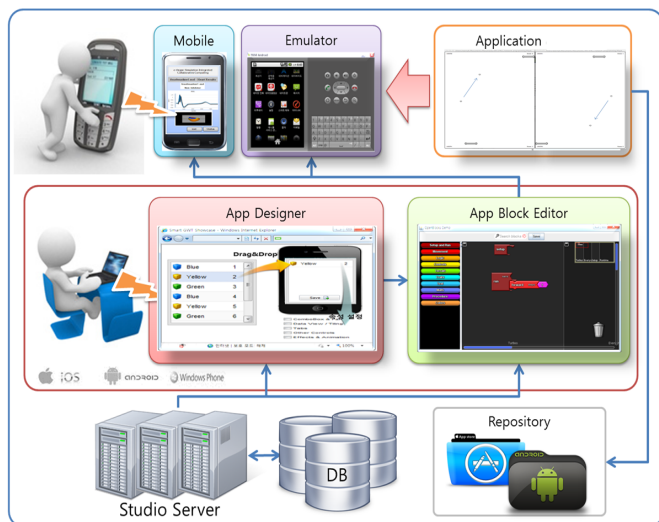
1. 서론

지능형 응용 개발 스튜디오는 단일소스로 이기종 OS 를 지원하는 개인화된 응용프로그램을 개발하고 실행 시 복잡한 처리를 클라우드 상의 개인가상단말에서 실행하는 지능적인 형태의 처리 로직 및 가상 테스트 환경을 제공하는 통합개발환경을 말한다. 이때 어플리케이션의 개발은 App Designer 와 App Block Editor 를 이용해 이루어진다.

2. 기본 개념 및 구현 방향

지능형 응용 개발 스튜디오의 어플리케이션 디자이너(App Designer)와 어플리케이션 블록 에디터(App Block Editor)를 활용하여, 간단한 게임을 제작한다.

어플리케이션 디자이너는 어플리케이션 개발 언어에 대한 지식이 없어도 지능적으로 어플리케이션을 디자인할 수 있는 기능을 가지고 있다. 또한 블록 에디터는 (그림 2)와 같은 개념으로, 역시 응용프로그램 개발 경험이 없는 사람이라도 간단히 Drag&Drop 으로 기능을 개발할 수 있도록 설계되어 있다. 레고 블록으로 집을 만들 듯이 어플리케이션의 기능을 블록을 쌓아서 개발하는 것이다.



(그림 1) 지능형 응용 개발 스튜디오 구조



(그림 2) App 블록 에디터

이를 활용하여, 네트워크로 진행하는 간단한 온라인 게임을 제작해본다. 그 과정은 Matching Server 의 제작과 Client Game Application 제작의 두 과정으로 나뉘어진다.

3. 어플리케이션 제작

3.1 Matching Server 제작

클라이언트 간의 TCP 통신을 중계하기 위한

Matching Server Application 을 제작한다. 클라이언트들이 1:1 로 대화해야 하므로, 이들을 매칭시켜줄 기능이 필요하며 이것이 Matching Server 의 핵심이다. 서버에 접속한 순서대로 클라이언트들을 1:1 매칭시켜주며, 둘 중 한 클라이언트가 접속을 종료했을 경우, 파트너 클라이언트의 접속을 자동으로 종료시켜준다. 다음 표는 유저 1~4 가 서버에 접속하고 종료하는 예시를 순서대로 설명하고 있다.

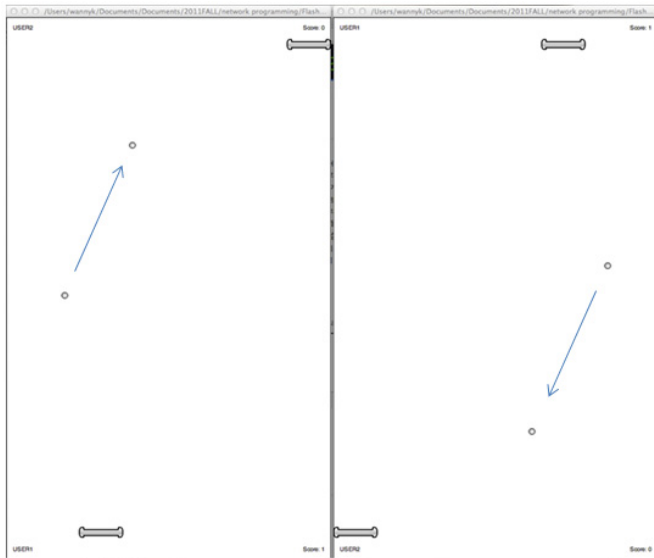
<표 1> 유저 매칭 프로세스

1	2	3	4	5
유저 1 접속	유저 2 접속	유저 3 접속	유저 4 접속	유저 3 종료
대기	1-2 매칭	대기	3-4 매칭	유저 4 강제종료

구현된 Matching Server 에 두 클라이언트가 접속하여 매칭이 완료되면, 서버는 한쪽의 클라이언트가 전송하는 데이터를 그대로 전부 다른 클라이언트에게 전송하는 단순한 기능만을 한다.

3.2 Tennis Game 제작

최종적으로 구현한 게임은 온라인으로 접속해 다른 유저와 1:1 로 진행하는 테니스 게임이다. 게임 플레이는 Real-time 으로 진행되며, TCP 전송의 지연이 일어나는 경우에도 잘 작동되도록 설계해야 한다. Real-time 으로 진행되는 게임인 만큼 Synchronization 이 중요하며, 이점에 중점을 두어 제작하였다.



(그림 3) 게임 실행 화면

(그림 3)은 구현된 테니스 게임을 한 컴퓨터에서 두 개 실행하여 서로 접속, 플레이 하는 모습입니다. 이와 같이, 두 클라이언트는 상하좌우가 반전된 모습으로 똑같이 작동한다. 이를 위해 두 클라이언트의 작동이 싱크되어야 하며 이는 게임 플레이를 위해 가장 중요한 요소라고 할 수 있다.

이를 구현하기 위해 게임의 구역을 가상으로 나눈 뒤

한쪽 클라이언트가 다른 쪽 클라이언트에게 좌표를 전송하는 방식으로 게임을 설계하였다. 즉, 좌표의 전송 상황에 지연이 생기더라도 두 클라이언트의 싱크가 맞지 않아 서로 다른 게임 상황이 표현되는 일은 일어나지 않게 된다. 다만 딜레이가 너무 심할 경우 공이 자꾸 뒤로 돌아가는 현상이 발생하여 정상적인 게임 플레이가 불가능하게 된다. 이를 방지하기 위해 좌표 전송량의 적절한 조절이 필요하다.

그리하여 가변 프레임 속도를 도입하였다. FPS(frame per seconds)를 가변적으로 설정하여, Client Application 의 실행 속도가 낮아지거나 네트워크 전송에 지연이 생기는 경우 프레임 속도를 떨어뜨려 원활한 플레이가 가능하도록 하는 것이다.

4. 결론

지금까지 지능형 응용 개발 스튜디오 환경에서 온라인 멀티 플레이가 가능한 간단한 게임을 제작하는 과정을 살펴보았다.

지능형 응용 개발 스튜디오의 어플리케이션 디자이너와 블록 에디터를 활용하면, 개발 언어에 대한 지식이 없거나 개발 경험이 없는 사람도 코드 작성 없이 간단히 Drag&Drop 으로 기능을 개발할 수 있다.

이를 활용해, 먼저 다수의 클라이언트들을 연결해 줄 Matching Server 를 구현하여 클라이언트들이 서버에 접속하는 순서대로 짝을 지어 매칭해주는 기능을 수행하도록 하였다. 또한 최종적으로 Real-time 방식의 테니스 게임을 구현하고 특성을 분석하였다.

ACKNOWLEDGEMENT

본 논문은 2012 년도 정부(지식경제부)의 재원으로 개인 및 기업 맞춤형 서비스를 위한 개방형 모바일 클라우드 용 통합개발환경 및 이기종 단말-서버 간 협업 기술 개발[2012-10039260]과 BK21 사업의 지원을 받아 수행된 연구임.

5. 참고자료

- [1] Andrew Begel, Eric Klopfer “StarLogo TNG: An Introduction to Game Development”, 2004
- [2] Ricarose Vallarta Roque “OpenBlocks: An Extendable Framework for Graphical Block Programming Systems”, 2007