



멀티플랫폼 지원 스트리밍 기반 게임 서비스

이동춘* · 배수영* · 김경일* · 조창식*

*한국전자통신연구원 스마트게임플랫폼연구실

목 차

I. 서론	IV. 구현
II. 관련 연구	V. 결론
III. 스트리밍 기반 게임 서비스 요소기술	

I. 서론

스마트폰, 태블릿 PC의 대중화와 더불어, WCDMA, HSDPA 등 3세대 이동통신과 LTE 통신 기술의 고도화에 따라, 시간과 장소에 제약 없이 언제 어디서나 사용자가 원하는 서비스를 받을 수 있는 유비쿼터스 컴퓨팅 환경이 도래하였다. 게임 시장에 있어서도 이러한 변화로 인해 기존 PC 중심의 게임시장은 정체되고 있는 반면, 모바일 게임 시장은 점차적으로 확대되어 가고 있는 추세이다. 하지만 이러한 모바일 게임 시장의 성장세에도 불구하고, 모바일 디바이스의 성능 한계로 사용자가 원하는 고품질의 3D 게임보다는 2D 게임 위주의 콘텐츠 성장이 이루어지고 있다. 이러한 문제를 해결하기 위한 대안으로 고성능의 게임 실행을 PC에서 수행하고 실행된 결과 영상을 모바일 단말기로 전송하여 게임 플레이를 하는 스트리밍 기반 게임 서비스가 시도되고 있다.

본 논문에서는 저사양의 모바일 단말에서도 고품질의 3D 게임 콘텐츠를 플레이 할 수 있는 스트리밍 기반 게임 서비스 기술에 대해 소개한다. 스트리밍 기반 게임은 게이머가 자신의 하드웨어 장치에 게임을 설치하지 않고도 게임을 플레이할 수 있는 서비스로, 게이머는 네트워크 통신을 통해 스트리밍 서버에 접속한 다음 스트리밍 방식으로 전송되는 게임 영상을 통해 게임을 즐길 수 있다. 게임 실행 및 게임 화면을 디스플레이하는 하드웨어 자원이 원격의 서버 단말기에 있고, 게임 플레이를 통해 서버 단말기에 만들어진 화면

을 실시간 캡처 및 인코딩 작업을 통해 H.264 등 동영상 형태로 실시간 압축한 후 네트워크 통신을 통해 게이머에게 전달해준다. 이러한 스트리밍 기반 게임 서비스 기술은 모든 연산이 서버단에서 일어나기 때문에 사용자 단말의 성능에 무관하게 고성능의 게임을 즐길 수 있다. 또한 실제 게임 플레이가 스트리밍 서버단에서 이루어지기 때문에, 서버단의 환경에 맞추어 개발된 게임 콘텐츠를 스마트폰, 태블릿, 스마트TV 등 다양한 클라이언트 단말에서도 게임을 즐길 수 있어 게임 콘텐츠에 대한 멀티플랫폼 서비스가 가능하게 된다.

본 논문의 2절에서 스트리밍 기반 게임 서비스 기술에 대해 살펴보고, OnLive, Gaikai, C-games 같은 기존 기술 현황을 소개한다. 3절에서는 멀티플랫폼 지원 스트리밍 기반 게임 요소기술에 대해 설명하고, 4절에서 구현을 보이며, 5절에서 결론을 맺는다.

II. 관련 연구

2.1. 스트리밍 게임 서비스

스트리밍 기반 게임 서비스 기술은 대규모의 스트리밍 서버를 구성하고, 게임 콘솔이나 고성능 PC에서 수행하던 컴퓨터 게임을 스트리밍 서버상에서 구동하는 기술이다. 스트리밍 서버에서 실행된 게임 영상과 소리는 유무선 네트워크를 통해서 스트리밍 방식으로 사용자의 단말에 전송되어 재생되며, 사용자 단말기에

서 발생하는 사용자 입력은 스트리밍 서버에 전송되어 게임의 다음영상 재생에 반영된다.^[1]

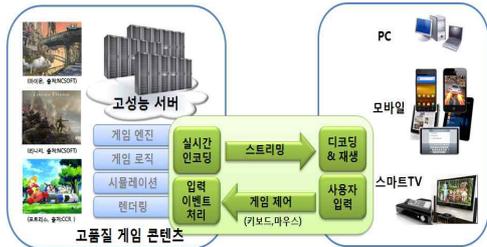


그림 1. 스트리밍 기반 게임 서비스 구성도

스트리밍 기반 게임 서비스 기술은 클라이언트단에 게임 설치 없이 게임 플레이가 가능하여 게임에 대한 사용자의 초기 진입 장벽을 낮출 수 있는 기술이다. 무엇보다도 모든 게임의 리소스 처리가 서버단에서 이루어지기 때문에 저사양의 모바일 단말기에서도 고품질의 PC게임을 즐길 수 있으며, 하나의 게임 콘텐츠를 스마트폰, 태블릿, 스마트TV 등 이기종 단말기에서 동시에 즐길 수 있는 멀티플랫폼 게임환경을 제공하는 장점을 가진다.

그러나 스트리밍 게임 서비스는 사용자가 게임을 플레이 하기 위해서는 중간에 스트리밍 서버를 항상 거쳐야 하기 때문에 게임의 실시간 반응성이 기존 게임 방식에 비해 떨어진다는 단점을 가진다. 또한 게임 영상과 소리를 네트워크 통신을 통해 클라이언트 단으로 전송하기 때문에, 많은량의 네트워크 트래픽이 발생한다. 네트워크 트래픽을 줄이기 위하여 게임 영상과 소리 데이터를 실시간 압축 후 클라이언트 단으로 전송하는 방법을 사용하지만, 압축시간으로 인하여 게임의 실시간 반응성은 좀 더 떨어지게 된다.

2.2. OnLive

OnLive^[2] 기술은 2009년 GDC(Game Developers Conference)에서 소개되었으며 2010년 미국에서 윈도우 PC와 Mac 등의 테스트탑을 대상으로 서비스를 시작하였다. 마이크로콘솔(Microconsole)이라 불리는 어댑터를 추가 발매하여 현재는 TV와 태블릿을 통해서도 게임 플레이를 지원한다. PC의 입력은 키보드와 마우스를 이용하고, TV와 태블릿 단말의 경우, 마이크로 콘솔과 연동되는 전용 무선 컨트롤러가 필요하다.

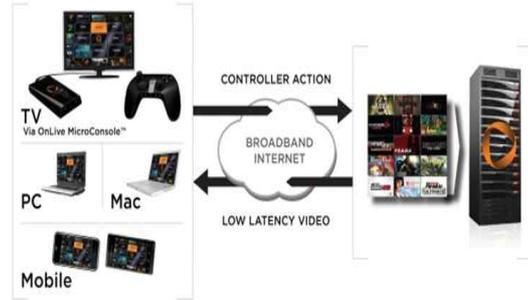


그림 2. OnLive 게임 스트리밍 아키텍처 (자료:OnLive)

OnLive는 여러곳에 데이터센터를 구축하여, 그림 2와 같은 서비스 플랫폼을 운영하고 있으며, 각 데이터 센터내의 장비는 CPU와 GPU외에 OnLive가 자체 개발한 비디오 압축 칩을 탑재하고 있다. 720P의 고품질 영상의 서비스를 지원하기 위해 5Mbps 이상의 네트워크 환경을 기반으로 하고, 각 데이터 센터에서 1천 마일 이내의 거리에 있는 사용자에게도 고품질의 서비스를 제공하고 있다. 서버는 2개의 영상 스트림을 생성하여, 게임을 수행하는 사용자와 게임 플레이를 관전하는 사용자 모드를 지원한다.

영상 인코딩 지연을 최소화하기 위해 자체적인 영상 인코딩, 디코딩 코덱을 사용하며, 인터넷 연결 최적화를 통해 네트워크 전송 시간을 포함 인코딩, 스트리밍, 디코딩, 1개의 프레임 재생 시간을 합하여 67.7ms가 가능하다.^[3] 게이머의 반응 속도에 아주 민감한 1인칭 슈팅 게임의 경우가 100 ms 이하의 지연 시간을 만족하면 서비스 가능하다는 연구 결과를 볼 때^[4], 지연에 따른 반응 속도면에서 여러 종류의 게임을 스트리밍 기반으로 서비스할 수 있는 가능성을 보여주고 있다. 하지만, 이는 이상적인 네트워크 상태에서 실시한 지연 테스트이며, 게임 타이틀에 따라 150~200ms의 지연 시간이 있음을 지적한 조사 결과도 있다.^[5]

2.3. Gaikai

Gaikai^[6]의 스트리밍 기반 게임 서비스는 사용자 단말에서 별도의 콘솔 없이 웹 브라우저와 플래시 프로그램으로 스트리밍 게임을 제공한다. 서버에서 비디오 게임을 실행하여 생성된 영상과 음향을 인터넷으로 전달하면, 사용자는 단말에 설치된 웹브라우저의 Java 또

는 Adobe Flash, Silverlight 플러그인을 이용해 전송된 스트림을 재생한다. 게임 영상과 음향 압축은 업계의 표준 동영상 코덱(MPEG-4와 H.264 등)을 사용한다.

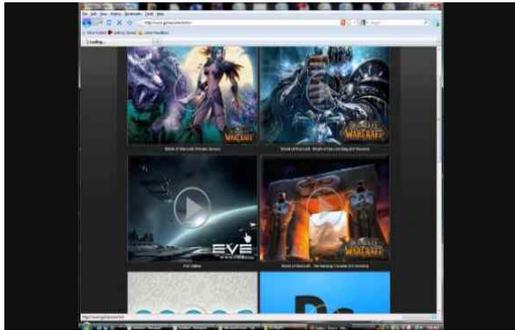


그림 3. 웹 브라우저를 통한 Gaikai 서비스
(자료: Massively)^[7]

Gaikai는 사용자 단말기에서 어떤 추가적인 동작이 필요없는 기술을 목표로 하며, 웹브라우저만 실행하면 바로 게임을 시작할 수 있는 간편함을 내세우고 있다. 또한, 적용 분야도 기존의 게임 서비스를 대체하기 위한 수단이 아니라, 게임 타이틀에 대한 홍보 시장을 목표로 하여 개발되고 있다. 사용자는 추가적인 하드웨어 장비나 소프트웨어의 다운로드, 인스톨 등의 설치 과정이 필요없이 데모 게임을 즉시 실행해 볼 수 있고, 게임 개발자 입장에서는 게임 코드가 외부에 유출되지 않을 뿐만 아니라, 패치나 업그레이드가 용이하게 된다.

2010년 E3에서 공식적인 발표 및 데모 동영상을 소개한 이후, 2010년 하반기 클로즈 베타테스트를 실시하고, 2011년 2월 미국내에서 공식적인 서비스를 시작했다. 지난해 12월 유럽 서비스를 개시하여 북미와 유럽내 12개국에 서비스를 제공하고 있다. 올해 초 CES2012에서 LG전자의 시네마 3D TV와 전략적 제휴를 맺고, 스마트 TV를 통한 클라우드 서비스를 발표했으며^[8], 페이스북 진출을 선언하는 등 활발한 활동을 벌였다. 그러나 2012년 8월 Gaikai는 3억 8천만달러에 소니컴퓨터엔터테인먼트(SCE)에게 인수되어진다. 소니는 앞으로 Gaikai가 가진 클라우드 게임 서비스 기술을 이용해 소니 플레이스테이션에서만 가능했던 게임을 스마트폰, 태블릿, 스마트TV, 노트북 등의 기기에서 서비스 할 예정에 있으며, 특히 소니의 차세대 플레이스테이션 게임기기인 PS4에 게임 스트리밍 기술이

들어갈 것으로 알려졌다.

2.4. C-games

C-games^[9]는 2012년 7월 LG유플러스에 의해 국내 최초로 상용 서비스를 시작한 스트리밍 기반 클라우드 게임 서비스로 초기에는 LG유플러스 통신 가입자만을 대상으로 하였다. LG유플러스, 엔비디아, 유비투스 3사의 협력으로 서비스가 시작되었으며, 엔비디아는 서비스를 위한 게임이 저장될 서버와 게임 정보를 실시간으로 전송해줄 시스템을 제공하고, LG유플러스는 게임 정보를 전송해주는 초고속 네트워크를 제공하며, 유비투스는 클라우드 게임을 개발하기 위한 개발 킷을 제공한다.

유저들은 특정 게임을 구입하기 이전에 미리 서비스 게임을 10분간 무료로 체험할 수 있으며, 1일/7일/30일/무제한 이용권 등을 구입할 수 있다. 게임 이용권을 구입하면 특정 기기뿐만 아니라 플레이할 수 있는 것이 아니라 PC, IPTV, 스마트폰 등 모든 기종에서 추가 요금 없이 플레이할 수 있다. 영상의 질적인 면에 있어서 움직임이 적은 게임에서는 체감 그럴 듯한 화면으로 플레이가 가능하지만, 대전 격투 게임과 같이 화면의 변화가 많은 게임을 플레이하면 네트워크 전송 문제로 인해 화면이 깨지기도 한다.

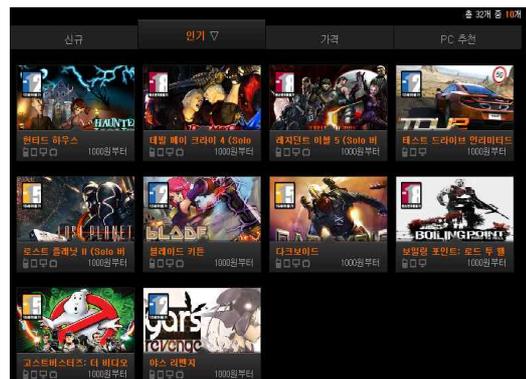


그림 4. C-games 게임 타이틀(자료:C-games)

LG유플러스는 2013년 5월 C-games를 타사 이동통신 고객들에게 전면 개방하여, KT, SKT 고객들도 모바일, 태블릿, PC 등에서 C-games서비스를 즐길 수 있게 되었다. 현재 C-games에서 제공하는 게임 콘텐츠는 '야

구의 신', '스트리트파이터 X 철권' 등 40여 종이 있다.

Ⅲ. 스트리밍 기반 게임 서비스 요소기술

스트리밍 기반 게임 서비스를 위해서는 클라이언트 단말의 개발과 서버 단말의 개발이 필요하다. 이중 클라이언트 단말의 개발은 비교적 단순한 작업으로, 사용자로부터 게임 입력을 전달 받는 기능이 추가된 동영상 디코더 개발이 추가 된다. 클라이언트 디코더에서 사용자의 입력을 받을 때, 서버 단말 입력장치와 클라이언트 단말 입력장치가 다를 경우 키 입력의 매핑 기능이 필요하다. 이러한 이기종 단말간의 키 매핑을 위한 방법으로 게임 영상위에 게임 입력을 위한 독자적인 버튼 모양의 스킨을 입힌 후, 스킨의 각 버튼을 누를 경우 각 버튼에 대응하는 키값을 서버로 전송하는 방법을 사용할 수 있다.

서버 단말 개발에 있어서 중요한 부분은 하나의 서버 시스템에 여러명의 클라이언트를 지원하기 위한 서버 자원의 가상화이다. 서버 자원의 가상화는 OS 레벨의 가상화와 응용 레벨의 가상화가 있다. OS 레벨 가상화는 하나의 호스트 OS 위에 여러개의 게스트 OS를 구동하는 방법으로, 호스트 OS와 게스트 OS는 서로 다른 종류의 OS가 실행될 수 있다. OS 레벨 가상화 솔루션으로는 VirtualBox, VMWare, KVM(Kernel Virtual Machine) 등이 있으며, 하나의 게스트 OS에서 실행되는 응용프로그램은 다른 게스트 OS에서 구동되는 응용프로그램과 독립적인 환경에서 실행되어지기 때문에 하나의 호스트 자원을 상호 독립적으로 사용할 수 있다. OS 레벨의 가상화 방법은 서버 자원 가상화를 위해 응용 프로그램에서의 별도 작업이 없어서 편리하나, 스트리밍 서비스를 위한 응용 프로그램을 실행하기 위해서는 가상화 하고자 하는 개수 만큼의 게스트 OS를 실행해야 하기 때문에 시스템 부하가 많은 단점을 가진다.

스트리밍 기반 게임 서비스를 위한 응용 레벨의 가상화를 위해서는 서버의 영상, 소리, 사용자 입력에 대한 가상화가 필요하다. 여러명의 클라이언트가 하나의 서버에 접속해서 동시에 게임 플레이를 할 경우, 하나의 서버에는 게임 영상과 소리가 한꺼번에 재생되고, 재생 데이터를 각각의 클라이언트로 전송하기 위해서

는 개별 사용자에 대한 영상과 소리를 독립적으로 관리하는 기술이 필요하다. 또한 각각의 클라이언트 단말로부터 입력되는 사용자 입력을 서버에서 실행되는 각각의 게임에 입력으로 전달하기 위한 사용자 입력에 대한 독립적인 관리 기술도 필요하다.



그림 5. 윈도우 GDI를 사용한 캡처 결과

게임 영상 가상화에서 중요한 것은 클라이언트 단말로 전송되어질 게임 영상을 획득하는 일이다. 게임 영상을 획득하는 방법으로는 크게 윈도우 GDI를 사용하는 방법과 DirectX를 사용하는 방법이 있다.^[10] 윈도우 GDI를 이용하는 방법은 실행되는 게임의 윈도우 핸들을 사용하여 윈도우별로 독립적으로 캡처하는 방법으로, 게임 화면이 다른 윈도우에 가려지더라도 원본 게임 화면을 캡처할 수 있는 장점을 가지지만, 캡처에 걸리는 시간이 길다는 단점이 있다. 반면 DirectX를 사용하여 캡처하는 방법은 그래픽 카드 내부의 비디오 메모리를 직접적으로 접근하여 화면을 캡처하는 방법으로 게임 영상의 빠른 캡처가 가능하지만, 게임 영상이 다른 윈도우에 가려져있을 경우 가려진 게임 영상의 획득이 불가능하다는 단점을 가진다. 이 때문에 DirectX를 이용하여 여러개의 게임 영상을 캡처할 때는 게임 영상들이 서로 겹쳐지지 않도록 화면을 분할 배치하여 사용하여야 한다.



그림 6. DirectX를 사용한 캡처 결과

게임에서 플레이되는 사운드 데이터들은 사운드 드라이버를 통해서 실제 소리로 재생되어지는데, 게임 사운드 가상화를 위해서는 사운드 드라이버의 기능을 수정하여 가상화가 가능하다. 일반적인 사운드 드라이버는 입력되는 소리 신호를 스피크로 출력하는 역할을 수행하지만, 가상화를 위한 사운드 드라이버에서는 입력되는 소리 신호를 게임별 고유의 메모리 주소 공간에 저장하고, 저장된 데이터를 인코딩한 후 각각의 클라이언트 단말에 네트워크 통신을 통해 보내주는 역할을 수행한다.

일반적으로 게임에서 사용하는 사용자 입력 처리 방법에는 윈도우 메시지를 이용한 방법과 DirectInput을 이용한 방법이 있으며, 사용자 입력의 가상화를 위해서는 이 두가지 방법에 대한 처리가 필요하다. 윈도우 메시지 기반의 입력 처리 방법은 여러 클라이언트 단말에서 사용자 입력이 동시에 들어오더라도, 서버측에서는 수신된 입력 순서대로 윈도우 메시지를 생성한 후 해당 게임에 전달해 주면 되기 때문에 입력 신호의 가상화는 어렵지 않다. 그렇지만 DirectInput을 사용한 게임의 경우, 게임에 사용자 입력을 전달하기 위해서는 해당 게임 윈도우가 활성화되어야 하고, 윈도우의 포커스가 해당 게임 윈도우에 있어야 한다. 그렇지만 윈도우 운영체제에서는 하나의 윈도우만이 윈도우의 포커스를 가질 수 있기 때문에, 일정시간 동안 두 개의 게임에 사용자 입력을 동시에 지속적으로 입력하는 것이 불가능하다. 그림 7은 DirectInput을 사용하는 두 게임에서 하나의 게임에는 'A'키가 계속해서 입력되고, 다른 게임에는 'B'키가 계속해서 입력될 때, 키 입력에 대한 게임의 반응 결과를 나타낸 것이다. 시간 t0에 실제 입력 키 값은 'A'키, 'B'키 두 개이지만, 게임의 반응 결과를 보면 'B'키 값만을 인식함을 알 수 있다.

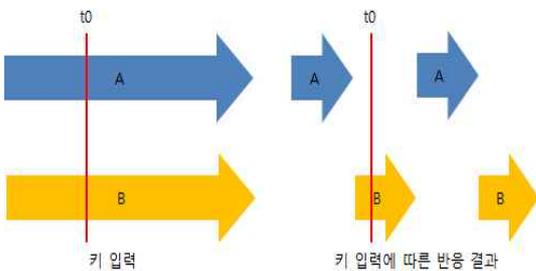


그림 7. 동시 키 입력에 대한 게임의 반응 결과

이러한 입력과 반응 결과의 차이는 DirectInput이 하나의 시스템에 여러개의 게임이 동시에 플레이되는 것을 고려치 않았기 때문에 발생하는 현상으로, 이러한 문제를 해결하기 위해서는 DirectInput의 입력 처리와 관련 있는 함수들을 DLL Injection 작업을 통해 원하는 함수로 대체해 줘야 한다.

IV. 구현

멀티플랫폼 지원 스트리밍 기반 게임 서비스 기술은 서버 플랫폼과 클라이언트 플레이어로 구성된다. 서버 플랫폼은 다양한 화질의 클라이언트 플랫폼에 스트리밍 게임 서비스를 제공하기 위한 스트리밍 게임 서버군과, 서버군을 관리하고 클라이언트 서비스 요청 시 적절한 게임 스트리밍 서버를 할당해 주는 게이트웨이 서버로 구성된다.

그림 8은 스트리밍 게임 서버와 클라이언트 단말이 연동하여 게임 서비스 실행 중인 화면을 보여준다. 스트리밍 게임 서버는 응용 레벨의 가상화를 제공하여 서버 1대당 640*480@30fps 기준으로 최대 8개의 클라이언트에 대한 동시 접속을 지원한다. 다중 클라이언트에 대한 동시 서비스를 위해 필요한 그래픽 자원의 공유는 화면 분할 방식을 사용하고, 사운드 장치는 각 클라이언트별 가상 사운드 드라이버를 제공한다. 또한, 원격에서 전달되는 멀티 클라이언트의 입력 처리를 위해 실시간 입력 포커스 스케줄링을 제공한다.

게이트웨이 서버는 서버 플랫폼을 총괄하며, 서비스 사용자 정보 및 스트리밍 게임 서버 실행 정보, 클라이언트 연동 기록을 모두 DB로 구축하여 관리한다. 그림 9의 왼쪽 영역이 사용자 관리 화면이고, 가운데 영역이 스트리밍 게임 서버별 접속 사용자, 시스템 메모리, CPU 사용률, 네트워크 대역폭 사용량을 볼 수 있는 영역이다. 클라이언트 접속시 게이트웨이 서버는 클라이언트 해상도와 수집된 스트리밍 게임 서버군의 접속 사용자 수, CPU, 메모리, 네트워크 사용률을 고려하여 적절한 서버를 분배한다.



그림 8. 스트리밍 게임 서버 실행 화면
(640*480 해상도 지원 스마트 폰 연동 (좌), 800*600 해상도 지원 태블릿 단말 연동(우))

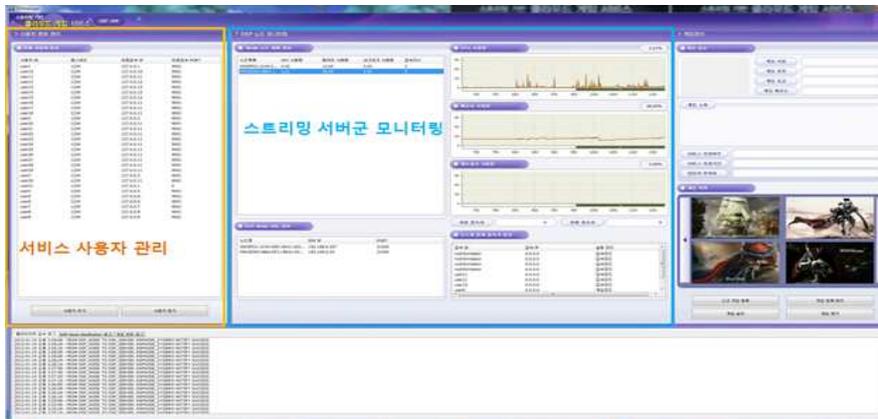


그림 9. 스트리밍 서비스를 위한 게이트웨이 서버



그림 10. 클라이언트 게임 플레이어

그림 10은 스마트폰에서 실행중인 클라이언트 게임 플레이어의 게임 실행 모습이다. 안드로이드 OS 기반 단말에 적용 되었으며, 스트리밍 프로토콜로 RTP(Realtime Transport Protocol)/ RTSP(Realtime Streaming Protocol)를 지원한다. 동영상 재생은 H.264 비디오와 AAC 오디오를 지원하며, H.264기준으로 640x480@30 fps의 성능을 보인다.

그림 11은 게임 입력 메뉴 스크립트를 다운 받아 구성된 스킨을 적용한 화면이다. 키보드, 마우스는 기본 모드로 제공되며, 별도의 게임 특화된 입력 메뉴를 스킨으로 구성한 예이다. 적용된 스킨은 포트리스 3D 게임용 입력을 간단하게 구성한 샘플이며, 캐릭터의 이동과 아이템의 각도 조절, 발사 버튼이 제공되고 있다.



그림 11. 게임 플레이어 게임 입력 메뉴 스킨

V. 결론

통신 기술이 급속도로 발달하고, 스마트TV, 스마트 기기, 모바일 단말기의 보급이 확대됨에 따라 사용자는 다양한 단말을 이용하여 유무선 인터넷이 연결되는 곳이면 장소에 구애를 받지 않고 원하는 게임을 즐길 수 있기를 원하나, 단말기의 성능상 고성능 PC에서와 같은 다양한 게임 콘텐츠를 무리 없이 실행하기에는 어려움이 따르고 있다. 또한, 저사양 PC에서 실행되도록 개발된 게임 콘텐츠들도 이들 단말에 이식하기 위해서는 이식 비용 및 추가 작업이 요구된다. 특히, 모바일 단말 및 스마트 단말과 같은 경량의 소형 단말일 경우, 하드웨어와 시스템 소프트웨어 등의 영향을 많이 받아, 이식하는 과정도 어렵고, 시스템 환경이 조금만 변해도 모든 단말마다 재작업해야 하는 어려움이 있다.

본 논문에서는 다양한 게임 콘텐츠를 제한된 하드웨어 및 소프트웨어 환경을 가지는 단말에서 손쉽게 제공하기 위한 스트리밍 기반 게임 서비스를 소개하였다. 본 서비스는 서버 플랫폼과 클라이언트 게임 플레이어로 구성된다. 서버 플랫폼은 다양한 화질을 제공하는 여러 스트리밍 게임 서버군으로 구성하여 다양한 단말의 사용자들에게 게임 스트리밍 서비스를 제공한다. 클라이언트 게임 플레이어는 PC, 스마트폰, IPTV 셋톱박스 등 다양한 하드웨어 플랫폼을 지원하며, 이들 단말에서 별도의 입출력 장치 연동없이 스트리밍 방식으로 고성능 그래픽과 렌더링을 필요로 하는 게임 콘텐츠를 서비스를 지원한다. 본 기술은 640*480 해상도와 800*600 해상도 기반으로 서버당 최대 8개의 클라이언트까지 연동할 수 있다.

참고문헌

- [1] 임충규, 김성수, 김경일, 원종호, 박창준, 클라우드 컴퓨팅 기반의 게임 스트리밍 기술 동향, 전자통신 동향분석 제 26권 제1호, pp47~56, 2011.
- [2] OnLive, www.onlive.com
- [3] Steve Perlman, The Process of Invention : OnLive Video Game Service, Columbia University(<http://tv.seas.columbia.edu/>), 2009.
- [4] Mark Claypool, Kajal Claypool, Latency and player actions in online games, Communications of the ACM - Entertainment networking, Volume

49, Issue 11, pp40~45, 2006.

- [5] Richard Leadbetter, OnLive Latency : The Reckoning, Eurogamer(www.eurogamer.net), 2010.
- [6] Gaikai Inc, www.Gaikai.com
- [7] James Egan, Gaikai's David Perry on solving latency issues with streaming games to your browser, MASSIVELY by joystiq (<http://massively.joystiq.com>), 2009.
- [8] 명진규, LG전자 스마트 TV에서 클라우드 게임 서비스, 아시아경제(<http://www.asiae.co.kr/>), 2012.
- [9] C-games, www.cgames.co.kr
- [10] Gopalakrishna Palem, Various methods for capturing the screen, Code Project(www.codeproject.com), 2006.

저자소개



이동춘(Dong-Chun Lee)

경북대학교 컴퓨터공학과 (석사)
한국전자통신연구원 선임연구원

※관심분야 : 게임입력 가상화, 실시간 스트리밍, GPU 기반 인코딩



배수영(Su-Young Bae)

경북대학교 컴퓨터공학과 (석사)
삼성전자 무선사업부
한국전자통신연구원 선임연구원

※관심분야 : 멀티미디어, 실시간 스트리밍, 게임 스트리밍



김경일(Kyoung-Il Kim)

고려대 경영정보대학원 (석사)
한국전자통신연구원 책임연구원

※관심분야 : 멀티미디어, 실시간 스트리밍, 게임 스트리밍



조창식(Chang-Sik Cho)

충남대학교 컴퓨터공학과 (박사)
한국전자통신연구원 책임연구원
"실시간 스트리밍 기반 게임서비스 기술" 과제책임자

※관심분야 : 클라우드게임 스트리밍, 온라인 게임 테스트, 멀티미디어, 임베디드 시스템