

(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.
G06F 3/00 (2006.01)
G06Q 99/00 (2006.01)

(11) 공개번호 10-2006-0115227
(43) 공개일자 2006년11월08일

(21) 출원번호 10-2005-0037666
(22) 출원일자 2005년05월04일

(71) 출원인 삼성전자주식회사
경기도 수원시 영통구 매탄동 416

(72) 발명자 오정환
경기 수원시 영통구 영통동 964-5 신나무실 신안아파트 533동306호

(74) 대리인 김동진
정상빈

심사청구 : 있음

(54) 어플리케이션에 원거리에서 제어가능한 인터페이스를제공하는 방법 및 장치

요약

어플리케이션에 원거리에서 제어가능한 인터페이스를 제공하는 방법 및 장치에 관한 것이다.

본 발명의 일 실시예에 따른 어플리케이션의 인터페이스를 변환하는 방법은 마우스 또는 키보드로 제어가능한 인터페이스를 포함하는 제 1 모드로 어플리케이션을 디스플레이 하는 단계, 상기 어플리케이션의 외관을 구성하는 데이터를 수집하는 단계, 리모트 컨트롤로 제어가능한 인터페이스를 포함하는 제 2 모드로 상기 어플리케이션을 디스플레이 하도록 상기 데이터를 변환하는 단계, 및 상기 변환된 데이터를 저장하는 단계를 포함한다.

대표도

도 5

색인어

투풋 UI(2-foot UI), 토포 UI(10-foot UI), 리모트 컨트롤, 윈도우, 이벤트, 메시지

명세서

도면의 간단한 설명

도 1은 미디어 재생기를 구성하는 여러 개체를 보여주는 예시도이다.

도 2는 도 1의 미디어 재생기(100)의 리소스 정보를 살펴본 도면이다.

도 3은 본 발명의 일 실시예에 따른 투풋 UI를 텐풋 UI로 변경하는 과정을 나타내는 도면이다.

도 4는 본 발명의 일 실시예에 따른 텐풋 UI로 실행되는 어플리케이션의 실행시의 화면이다.

도 5는 본 발명의 일 실시예에 따른 투풋 UI와 텐풋 UI간의 연동 메커니즘을 보여주는 도면이다.

도 6은 본 발명의 일 실시예에 따른 어플리케이션의 인터페이스를 텐풋 UI로 변환시키는 과정을 나타내는 순서도이다.

도 7은 본 발명의 일 실시예에 따른 텐풋 UI를 통해 입력된 신호를 어플리케이션의 해당 기능에 맞는 메시지로 변환하여 실행시키는 과정을 나타내는 순서도이다.

도 8은 본 발명의 일 실시예에 따른 인터페이스 변환부의 구성을 나타내는 도면이다.

도 9는 본 발명의 일 실시예에 따른 인터페이스 처리부의 구성을 나타내는 도면이다.

<도면의 주요 부분에 대한 부호의 설명>

100 : 어플리케이션 320 : 미리보기

500 : 텐풋 UI 900 : 리모트 컨트롤

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 어플리케이션에 원거리에서 제어가능한 인터페이스를 제공하는 방법 및 장치에 관한 것으로 더욱 상세하게는 근거리에서 제어가능한 인터페이스로 구성된 어플리케이션에 원거리에서 제어가능한 인터페이스를 제공하는 방법 및 장치에 관한 것이다.

최근 홈네트워크가 발전하면서 기존에 분리된 영역을 가지던 컴퓨터와 디지털 티비, 비디오, 셋톱박스 등이 서로 융합하고 서로의 기능을 차용하여 복합된 기능을 가지게 되었다. 이러한 변화중 하나로는 컴퓨터의 인터페이스가 기존의 마우스와 키보드를 바탕으로 하는 인터페이스에서 리모트 컨트롤을 도입하여 신호를 입력받아 기능을 수행하는 방식으로 변화하고 있다.

컴퓨터에 적용되는 모니터가 대형화 되거나 혹은 디지털 티비가 컴퓨터의 모니터 기능을 대신하면서, 과거의 입출력 방식이 사용자에게 불편함을 가중하게 되었다. 키보드와 마우스는 입력해야 할 사항들이 많을 뿐 아니라 유선으로서의 한계를 가지고 있다. 물론, 최근 무선 마우스와 무선 키보드 제품이 다양하게 생산되고 있으나, 이는 기본적으로 텍스트 기반과 아이콘 기반의 입출력이라는 큰 구조를 벗어나지는 못하고 있다.

마이크로 소프트사에서 출시한 미디어 센터(Media Center)에서는 컴퓨터가 가지는 인터페이스의 한계를 극복하고, 가전으로의 융합을 도모하기 위해 텐풋 사용자 인터페이스(10-foot User Interface, 이하 '텐풋 UI'라 한다)라는 개념을 창안하였다. 텐풋 UI는 사용자와 모니터 또는 디지털 티비등의 출력 장치와의 거리를 약 10 피트(3m)의 거리가 있는 상태에서 제공되는 사용자 인터페이스를 의미한다. 과거, 키보드와 마우스를 가지고 컴퓨터를 사용하던 시기에는 약 60cm(2 feet) 정도의 거리를 두었기 때문에 투풋 사용자 인터페이스(2-foot User Interface, 이하 '투풋 UI'라 한다)라 부른다. 텐풋 UI를 사용할 경우, 사용자는 리모트 컨트롤을 가지고 쉽게 어플리케이션을 사용할 수 있다. 특히 디지털 티비, DVD 등의 기기에서 종래 제공하였던 리모트 컨트롤 기반의 상하좌우 또는 특정 버튼과 기능을 연결시켜서 실행하는 방식이 가능해지면서, 사용자는 컴퓨터 모니터에 가까이 있을 필요가 없어졌다.

최근 컴퓨터 관련 업체들은 이러한 텐풋UI를 지속적으로 개발하여 가전제품과의 융합을 통한 홈네트워크의 구축에 한걸음씩 다가가고 있다. 그러나, 투풋 UI가 대부분을 차지하는 어플리케이션들을 모두 텐풋 UI로 바꾸는 것은 많은 비용과 시간을 필요로 한다.

DVD 재생, 음악감상, 동영상/그림 감상등 멀티미디어 컨텐츠등을 조작시에는 특별히 많은 버튼을 사용하지 않고 정지, 재생, 볼륨조절 등의 기능에 국한된다. 그러나 이들 어플리케이션들이 텐핏 UI를 지원하지 않는 경우, 어플리케이션을 사용하기 위해서 화면 가까이 다가가서 키보드와 마우스로 어플리케이션을 실행하거나 멀티미디어 컨텐츠를 재생하거나 정지시켜야 하는 어려움이 있다.

컴퓨터 어플리케이션들도 이를 위해 텐핏UI가 출시되고 있으나, 이는 모두 신제품이며, 기존에 널리 이용되던 어플리케이션들의 활용은 불가능하였다. 따라서 어플리케이션의 전반적인 수정 없이도 투핏 UI를 텐핏 UI로 쉽게 변경할 수 있는 방안이 필요하다.

발명이 이루고자 하는 기술적 과제

본 발명의 기술적 과제는 기존의 어플리케이션이 텐핏 UI를 제공하도록 변경하는데 있다.

본 발명의 다른 기술적 과제는 기존의 어플리케이션에 텐핏 UI를 추가하여 손쉽게 디지털 가전과의 융합을 가능하게 하는 데 있다.

본 발명의 목적들은 이상에서 언급한 목적들로 제한되지 않으며, 언급되지 않은 또 다른 목적들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

발명의 구성 및 작용

어플리케이션에 원거리에서 제어가능한 인터페이스를 제공하는 방법 및 장치에 관한 것이다.

본 발명의 일 실시예에 따른 어플리케이션의 인터페이스를 변환하는 방법은 마우스 또는 키보드로 제어가능한 인터페이스를 포함하는 제 1 모드로 어플리케이션을 디스플레이 하는 단계, 상기 어플리케이션의 외관을 구성하는 데이터를 수집하는 단계, 리모트 컨트롤로 제어가능한 인터페이스를 포함하는 제 2 모드로 상기 어플리케이션을 디스플레이 하도록 상기 데이터를 변환하는 단계, 및 상기 변환된 데이터를 저장하는 단계를 포함한다.

본 발명의 일 실시예에 따른 어플리케이션을 디스플레이하는 방법은 리모트 컨트롤로 제어가능한 인터페이스를 포함하는 모드로 어플리케이션을 디스플레이하는데 필요한 외관을 구성하는 데이터를 수신하는 단계, 상기 모드로 상기 어플리케이션을 디스플레이하는 단계, 및 리모트 컨트롤로부터 상기 모드로 디스플레이된 인터페이스의 구성요소를 선택하는 입력 신호를 수신하는 단계를 포함한다.

본 발명의 일 실시예에 따른 컴퓨터 시스템은 마우스 또는 키보드로 제어가능한 인터페이스를 포함하는 제 1 모드로 어플리케이션을 실행하며, 상기 어플리케이션이 상기 제 1 모드로 실행되는 경우, 상기 어플리케이션의 외관을 구성하는 데이터를 수집하는 어플리케이션 분석부, 리모트 컨트롤로 제어가능한 인터페이스를 포함하는 제 2 모드로 상기 어플리케이션을 디스플레이할 수 있도록 상기 데이터를 변환하는 어플리케이션 변환부를 포함한다.

본 발명의 다른 실시예에 따른 컴퓨터 시스템은 리모트 컨트롤이 송신한 입력 신호를 수신하는 신호 수신부, 리모트 컨트롤로 제어가능한 인터페이스를 포함하는 모드로 어플리케이션이 실행될 수 있도록 상기 어플리케이션의 외관을 구성하는 데이터를 수신하는 데이터 수신부, 및 상기 어플리케이션을 상기 모드로 디스플레이하는 디스플레이 제어부를 포함하며, 상기 디스플레이 제어부는 상기 입력 신호에 따라 상기 모드로 실행중인 어플리케이션의 소정의 기능을 실행한다.

본 발명의 일 실시예에 따른 소프트웨어는 리모트 컨트롤이 송신한 입력 신호를 어플리케이션을 구성하는 소정의 구성 요소에 대한 이벤트 신호로 변환하는 이벤트 처리부, 상기 어플리케이션의 상기 구성요소가 상기 이벤트 신호를 받는 경우 제공하는 기능을 실행하는 제어부를 포함하며, 상기 어플리케이션은 리모트 컨트롤로 제어가능한 인터페이스를 포함하는 모드로 실행되며, 상기 모드는 상기 어플리케이션을 1m 이상의 거리에서 제어가 가능하도록 상기 어플리케이션을 디스플레이하고 리모트 컨트롤로부터 제어 신호를 수신한다.

본 발명의 일 실시예에 따른 저장매체는 리모트 컨트롤로 제어가능한 인터페이스를 포함하는 모드로 어플리케이션을 실행하도록 상기 어플리케이션의 외관을 구성하는 데이터, 및 상기 어플리케이션이 실행될 때, 상기 데이터로 상기 어플리케이션의 외관을 변환하여 제공하는 인터페이스 처리부를 포함하며, 상기 데이터는 상기 어플리케이션의 구성요소를 실행시킬 경우 제공하는 기능에 대한 정보를 포함한다.

본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나 본 발명은 이하에서 개시되는 실시예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 수 있으며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하고, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다. 명세서 전체에 걸쳐 동일 참조 부호는 동일 구성 요소를 지칭한다.

이하, 본 발명의 실시예들에 의하여 어플리케이션에 원거리에서 제어가능한 인터페이스를 제공하는 방법 및 장치를 설명하기 위한 블록도 또는 처리 흐름도에 대한 도면들을 참고하여 본 발명에 대해 설명하도록 한다. 이 때, 처리 흐름도 도면들의 각 블록과 흐름도 도면들의 조합들은 컴퓨터 프로그램 인스트럭션들에 의해 수행될 수 있음을 이해할 수 있을 것이다. 이들 컴퓨터 프로그램 인스트럭션들은 범용 컴퓨터, 특수용 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비의 프로세서에 탑재될 수 있으므로, 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비의 프로세서를 통해 수행되는 그 인스트럭션들이 흐름도 블록(들)에서 설명된 기능들을 수행하는 수단을 생성하게 된다. 이들 컴퓨터 프로그램 인스트럭션들은 특정 방식으로 기능을 구현하기 위해 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비를 지향할 수 있는 컴퓨터 이용 가능 또는 컴퓨터 판독 가능 메모리에 저장되는 것도 가능하므로, 그 컴퓨터 이용가능 또는 컴퓨터 판독 가능 메모리에 저장된 인스트럭션들은 흐름도 블록(들)에서 설명된 기능을 수행하는 인스트럭션 수단을 내포하는 제조 품목을 생산하는 것도 가능하다. 컴퓨터 프로그램 인스트럭션들은 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비 상에 탑재되는 것도 가능하므로, 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비 상에서 일련의 동작 단계들이 수행되어 컴퓨터로 실행되는 프로세스를 생성해서 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비를 수행하는 인스트럭션들은 흐름도 블록(들)에서 설명된 기능들을 실행하기 위한 단계들을 제공하는 것도 가능하다.

또한, 각 블록은 특정된 논리적 기능(들)을 실행하기 위한 하나 이상의 실행 가능한 인스트럭션들을 포함하는 모듈, 세그먼트 또는 코드의 일부를 나타낼 수 있다. 또, 몇 가지 대체 실행예들에서는 블록들에서 언급된 기능들이 순서를 벗어나서 발생하는 것도 가능함을 주목해야 한다. 예컨대, 잇달아 도시되어 있는 두 개의 블록들은 사실 실질적으로 동시에 수행되는 것도 가능하고 또는 그 블록들이 때때로 해당하는 기능에 따라 역순으로 수행되는 것도 가능하다.

이해를 돕기 위해, 마이크로 소프트사의 Windows 운영 시스템에서의 인터페이스를 바탕으로 한다. 그러나 이는 Windows 시스템에 국한되는 것은 아니다. 키보드 또는 마우스를 통해 특정 신호를 수신하고 각각의 개체가 해당 신호를 수신하여 필요한 작업을 수행하는 윈도우 기반의 모든 운영 시스템 또는 인터페이스에 적용가능하다. 예를 들어, 유닉스 또는 리눅스에서 사용하는 인터페이스의 하나인 X-Windows도 이에 해당하며, 매킨토시에서 제공하는 인터페이스에도 적용가능하다.

설명에 앞서 본 명세서에서 사용하는 용어의 의미를 간략히 설명한다. 그러나 용어의 설명은 본 명세서의 이해를 돕기 위한 것으로서 명시적으로 본 발명을 한정하는 사항으로 기재하지 않은 경우에 본 발명의 기술적 사상을 한정하는 의미로 사용하는 것이 아님을 주의해야 한다.

이벤트(Event)는 키보드, 마우스 등을 통해 들어오는 신호, 예를 들어, 키 스트로크(keystroke), 클릭(click), 드래그(drag) 등의 신호를 의미한다. 본 명세서에서는 어플리케이션에 전달되는 메시지의 일 실시예로 이벤트라 명명하고 설명하지만, 이에 국한되는 것은 아니다. 운영체제 또는 인터페이스 구현 방식에 따라 메시지, 이벤트, 인터럽트 등 다양하게 이름지어질 수 있다.

본 명세서에서는 기존의 인터페이스를 툴킷 UI로, 리모트 컨트롤 등으로 조정하는 인터페이스의 한 예로 툴킷 UI로 명명하고 있다. 그러나, 툴킷 UI는 키보드, 마우스 등으로 근거리에서 조정하는 인터페이스의 일 실시예이며, 툴킷 UI는 리모트 컨트롤과 같이 원거리에서 상하 또는 좌우와 같은 방향 버튼으로 제어가능한 인터페이스의 실시예이다.

Windows, X-Windows와 같은 윈도우 기반의 인터페이스에서 어플리케이션은 수많은 개체들로 이루어져 있다. 도 1은 미디어 재생기를 구성하는 여러 개체를 보여준다.

도 1에서 미디어 재생기(100)는 여러 버튼과 전체 틀로 구성된다. 버튼들과 전체 틀도 모두 각각의 개체로 존재하므로, 해당 개체에 특정한 이벤트가 일어날 경우 어떤 작업을 수행해야 하는지가 정해져 있다. 예를 들어 재생 버튼(10)을 마우스로 누를 경우, 마우스의 클릭 이벤트가 발생했음이 재생 버튼으로 전달된다. 그러면 재생 버튼(10)에 등록된 여러 기능 중에서 마우스 클릭에 대해 어떤 작업을 수행할지를 결정한다. 발생한 이벤트에 대해 어떤 기능을 수행할 지를 결정하는 것이

이벤트 핸들러(Event Handler)이다. 이 이벤트 핸들러에 통상 해당 이벤트에 대해 어떤 기능 또는 어떤 함수를 수행해야 하는지에 대한 정보가 등록되어 있다. 도 1의 여러 버튼들도 모두 마우스의 클릭 버튼이 발생할 경우 어떤 함수를 수행할 지 미리 정해져 있다.

한편, 개체는 어떤 기능을 수행할 것인가에 대한 정보 외에도, 개체가 어떤 형식으로 보여질 것인가에 대한 설정 정보도 가지고 있다. 예를 들어, 정지 버튼(20)의 크기가 가로 120 픽셀, 세로 100 픽셀인 경우, 이러한 정보도 개체의 상세 정보에 저장되어 있다. 따라서, 정보를 바꿀 경우에는 개체가 표현되는 방식도 달라진다. 상기의 정보를 통상 리소스(Resource)라 부른다. 리소스는 어플리케이션을 표현하거나 어플리케이션의 인터페이스를 구현하는데 필요한 정보로, 리소스 정보는 대부분 어플리케이션과 별개로 존재할 수 있다.

대부분의 윈도우 기반의 인터페이스는 수많은 개체들을 모두 윈도우로 구현하고 있다. 어플리케이션이 동작되는 하나의 윈도우 내에 여러 버튼, 캡션, 라디오버튼, 체크박스 등도 모두 윈도우의 일종으로 본다. 따라서 어플리케이션은 둘 이상의 윈도우들의 집합이라 볼 수 있다. 이러한 윈도우들을 하나의 어플리케이션내에서 체계적으로 보여주기 위한 큰 틀을 최상위 윈도우라 할 수 있다.

도 2는 도 1의 미디어 재생기(100)의 리소스 정보를 살펴본 도면이다. Spy++ 와 같이 어플리케이션의 구성 윈도우의 정보와 이벤트를 보여주는 프로그램을 실행할 경우 나타나는 리소스 정보를 간략히 나타내고 있다. 도 1의 미디어 재생기는 도 2와 같이 나타난다.

미디어 재생기(100)를 구성하는 개체들인 윈도우들에 대한 정보는 도 2의 201과 같다. 201에는 미디어 재생기(100)라는 어플리케이션을 구성하는 각각의 윈도우들의 윈도우 핸들값(Window Handle), 캡션, 타입에 대한 정보를 제공하고 있다. 미디어 재생기(100)는 "PLAYER"라는 이름을 가지고 있고 00000C1C라는 핸들 값을 가지는데, 이는 윈도우 내부에서의 고유한 식별자이다. 미디어 재생기(100)를 구성하는 요소로 전체 개체들을 포괄하는 윈도우인 PLAYER(핸들 값이 00000C20)이 존재한다. 이 윈도우 내에 존재하는 윈도우들은 미디어 재생기(100)에서 볼 수 있듯이 여러가지 버튼과 미디어에 대한 정보를 제공하는 판넬(Pannel)들이다. 각각에 대해서 윈도우 핸들, 캡션, 그리고 윈도우의 타입에 대한 정보를 제공하고 있다. 윈도우 핸들은 각 버튼 또는 각 판넬의 식별자가 되며, 캡션은 해당 판넬의 이름으로 볼 수 있다. 그리고 윈도우의 타입은 "Button" 또는 "Pannel"로 명시하고 있다. 이중에서 일시정지 버튼(30)에 대한 상세 정보(211)를 살펴보면 다음과 같다.

윈도우 캡션(Window Caption)은 "PAUSE"로 되어 있다. 윈도우에 다른 이미지가 결합되지 않을 때에는 윈도우 캡션이 윈도우에 나타난다. 도 2의 일시정지 버튼(30)은 이미지가 결합되어 있으므로 윈도우 캡션이 나타나지 않는다.

윈도우 핸들(Window Handle)은 000102F7로, 이는 일시정지 버튼(30)에 대한 윈도우 인터페이스 내부의 식별자이다. 윈도우 프로시저(Win Proc)는 이 윈도우에 대한 작업을 처리하는 프로세스의 번호를 보여준다. 그리고 영역(Rectangle)에서는 일시정지 버튼(30)이 차지하는 영역에 대한 정보를 제공한다. 이미지(Image)는 윈도우에 나타날 이미지에 대한 정보이다. RES_PAUSE_IMG는 이미지를 제공하는 리소스(Resource)에 대한 정보이다.

윈도우에 대한 정보중에 크기라든가 이미지, 위치에 대한 정보는 모두 리소스의 형식으로 제공된다. 리소스 정보는 윈도우와 같은 어플리케이션의 외관(외형)을 구성하기 위해 필요한 데이터들의 집합으로 볼 수 있다. 따라서 리소스 정보를 변환함으로써 투풋 UI를 텐풋 UI의 형식으로 변환할 수 있다.

도 3은 본 발명의 일 실시예에 따른 투풋 UI를 텐풋 UI로 변환하는 과정을 나타내는 도면이다. 텐풋 UI로 변환하기 위한 어플리케이션을 실행시킨 후, 각 버튼의 인터페이스를 대체할 UI를 선택한다. 예를 들어, 되감기 버튼(51)을 누른 후에 이 버튼에 대해 적용할 텐풋 UI 요소를 인터페이스 리스트(310)에서 선택한다. 그리고 선택한 요소(311)를 클릭하거나 드래그를 시켜서 미리보기 (320)에 설정할 수 있다. 미리보기(320)는 사용자가 설정한 텐풋 UI가 어떤 방식으로 사용자에게 보여질 것인가를 미리 보여주는 것으로, 컴퓨터 또는 노트북의 모니터, 디지털 TV와 같은 디스플레이 장치에 출력되는 형태를 축소시킨 것이라 볼 수 있다.

위치 선정은 자동 또는 수동으로 이루어질 수 있으며, 인터페이스 리스트에 나타나는 이미지들은 미리 보관, 제공되는 템플릿(template)을 사용하여 설정할 수 있다. 이 과정에서 사용자가 판단하기에 불필요한 요소에 대해서는 텐풋 UI의 인터페이스 요소를 선택하지 않을 수 있다. 이러한 과정을 통해, 사용자는 투풋 UI를 텐풋 UI로 변환할 수 있다. 또한 사용자에게 미리 저장된 템플릿을 보여주고 사용자가 이를 수락하여 적용할 것인지를 결정하도록 자동으로 제공하는 방식도 가능하다.

도 4는 본 발명의 일 실시예에 따른 텐풋 UI로 실행되는 어플리케이션의 실행시의 화면이다. 도 3의 미리보기(320)에 사용자가 설정한 버튼, 판넬등의 인터페이스가 화면에 나타나며, 사용자는 리모트 컨트롤을 통해 선택할 수 있다.

리모트 컨트롤의 상하좌우 방향키를 통해 버튼에 대한 포커스를 변경시킬 수 있다. 또한 사용자가 리모트 컨트롤을 통해 실행 또는 확인 버튼을 누르면 이에 해당하는 신호를 수신하여, 이를 해당 버튼에 대한 클릭 이벤트 신호로 변환시킨다. 그 결과, 사용자가 리모트 컨트롤로 실행 또는 확인 버튼을 누른 것이, 실질적으로 해당 어플리케이션에 대한 클릭 이벤트를 발생시켜서 해당 어플리케이션의 특정 버튼을 누른 것으로 인식하여 해당하는 기능을 수행할 수 있다.

예를 들어 사용자가 리모트 컨트롤의 상하좌우 버튼을 통해 도 4의 네 개의 버튼 중 하나(321)를 선택할 수 있다. 선택된 버튼의 배경색을 달리하거나 테두리를 굵게 하거나 또는 버튼에 나타나는 이미지를 동적으로 변화시키는 애니메이션 효과를 주어서 해당 버튼이 선택됨을 나타낼 수 있다. 그리고 사용자가 리모트 컨트롤의 실행 또는 확인 버튼을 누를 경우, 이 신호가 현재 포커스된 버튼에 대해 이루어진걸 알 수 있으므로, 이 신호를 클릭 이벤트로 바꾸어 발생시킨다. 그 결과, 실제 어플리케이션에서는 321 버튼에 대한 기능, 즉 되감기 기능을 수행한다. 리모트 컨트롤에서 생성하는 제어 신호 또는 입력 신호는 인터페이스를 구성하는 구성 요소에 전달되고, 이 신호는 어플리케이션에 전달된다. 이때, 어플리케이션이 구동되는 운영체계에 맞는 정보, 예를 들어 이벤트 메시지와 같은 정보가 어플리케이션으로 전달된다.

도 5는 본 발명의 일 실시예에 따른 투풋 UI와 텐풋 UI간의 연동 메커니즘을 보여주는 도면이다.

기존의 어플리케이션(100)에 대해 텐풋 UI를 설정하여 어플리케이션 실행시에 화면 전체에 대한 텐풋 UI(500)가 디스플레이된다. 이는 기존 어플리케이션에 대한 리소스 정보를 변경시키는 것을 통해 가능하다. 리소스 정보가 어플리케이션의 리소스 파일을 참조하는 경우, 텐풋 UI를 지원하는 리소스 파일을 새로이 생성하여 사용할 수 있다. 텐풋 UI(500)가 디스플레이되면 사용자는 리모트 컨트롤(900)을 통해 기능을 선택할 수 있다. 상하좌우 버튼을 통해 어플리케이션의 기능을 수행하는 버튼들 사이로 포커스를 이동시킬 수 있다. 이때, 사용자가 특정 버튼(321)에서 확인 버튼을 누를 경우, 텐풋 UI(500)는 확인 버튼이 눌러짐을 알리는 신호를 수신한다. 이 신호를 어플리케이션(100)의 되감기 버튼(51)에 대한 클릭 이벤트로 변경하여 어플리케이션(100)으로 보낸다. 어플리케이션의 되감기 버튼(51)에 대한 이벤트 핸들러는 수신된 클릭 이벤트에 따라 되감기 버튼(51)을 클릭했을 때 제공하는 함수를 호출한다.

지금까지 살펴본 텐풋 UI로 변경하고 리모트 컨트롤 신호를 수신하는 과정은 반드시 하나의 어플리케이션에 대해서만 수행되는 것은 아니다. 둘 이상의 어플리케이션을 선택하여, 이들 어플리케이션 전체를 포함하는 텐풋 UI를 생성할 수 있다.

도 6은 본 발명의 일 실시예에 따른 어플리케이션의 인터페이스를 텐풋 UI로 변환시키는 과정을 나타내는 순서도이다.

변환을 위해서는 변환을 수행하는 텐풋 UI를 구동시킨다. 그리고 투풋 UI를 제공하는 어플리케이션을 실행시킨다(S11). 어플리케이션을 실행할 경우, 어플리케이션을 구성하는 윈도우들의 리소스 정보를 알 수 있다. 분석 프로그램을 통해 어플리케이션의 구성 윈도우들을 분석하여, 각 윈도우에 대한 정보를 사용자가 알기 쉬운 형태로 출력한다(S12). 도 2에서 살펴본 바와 같이 크기, 윈도우 명칭, 이미지에 대한 리소스 정보 등을 제공한다. 그러나 프로그램 개발시 사용하는 SPY++ 와 같이 복잡한 정보를 제공할 필요는 없다. 사용자가 해당 윈도우의 외형 또는 위치를 자유로이 설정하는데 필요한 정보를 제공하면 된다.

제공된 정보를 바탕으로 버튼, 판넬 등을 구성하는 각각의 윈도우들을 텐풋 UI를 통해 디스플레이할 것인지를 결정한다(S13). 그리고 이들 윈도우들이 디스플레이될 경우 배치를 결정할 수 있다(S14). 그리고 이들 윈도우들의 형태 또는 윈도우들이 배치될 배경화면의 템플릿을 읽어와서 설정한다(S15). S13내지 S15 과정은 순서를 달리하여 적용가능하며, 도 3에서 살펴본 바와 같이 어플리케이션의 윈도우(버튼, 판넬)을 클릭한 후에, 변환된 이미지를 선택하여 드래그하여 배치하는 실시예가 가능하다. 또한, 사용자에게 미리 변환된 형태를 자동으로 생성하여 보여주고, 사용자는 이를 적용할 것인지 말 것인지를 판단하는 것으로 구현할 수 있다. 대부분의 어플리케이션은 재생, 정지, 되감기 등의 한정된 기능을 제공하므로, 어플리케이션을 텐풋 UI로 나타낸 결과는 어플리케이션에 따라 큰 차이가 없을 수도 있다. 따라서 미리 제공하는 템플릿 데이터를 사용하여 설정할 수 있다. 배치된 구성이 어떤 형식으로 나타날 것인지 미리보기 등의 기능을 통해 검토하고 변환 결과를 저장함으로써 텐풋 UI로 변경하는 과정이 완료한다(S16). 변환 결과는 파일의 형태로 저장가능하다. 또한 변환 결과를 다른 컴퓨터에 저장된 동일 어플리케이션에 대해서도 텐풋 UI를 제공하기 위해 파일의 형태로 저장하여 배포 가능하다. 통상 네트워크를 통해 사용자의 컴퓨터가 서버로 접속해서 다운로드 받거나, 어플리케이션을 개발한 회사 측에서 어플리케이션과 함께 배포하는 것이 가능하다.

도 7은 본 발명의 일 실시예에 따른 텐풋 UI를 통해 입력된 신호를 어플리케이션의 해당 기능에 맞는 메시지로 변환하여 실행시키는 과정을 나타내는 순서도이다. 어플리케이션을 실행시 함께 실행되어 어플리케이션에 텐풋 UI의 인터페이스가 실행되도록 할 수 있다. 또한 리모트 컨트롤로부터 소정의 제어 신호 또는 입력 신호를 받아서 이를 어플리케이션의 해당 기능이 활성화 또는 비활성화 되도록 이벤트 신호를 생성하여 보낼 수 있다.

어플리케이션을 실행시킨다(S21). 어플리케이션을 실행시키면 해당 텐풋 UI를 실행한다(S22). 텐풋 UI를 실행시킨다는 것은 도 6에서 변경한 리소스 정보를 보여준다는 것을 의미한다. 텐풋 UI가 실행되면 컴퓨터 모니터 또는 컴퓨터와 연결된 디지털 TV를 통해 어플리케이션을 리모트 컨트롤로 제어할 수 있다. 사용자가 리모트 컨트롤의 버튼을 누를 경우, 텐풋 UI는 리모트 컨트롤로부터 신호를 수신하여 분석한다(S25). 신호가 메뉴 또는 특정 기능을 실행시키거나 메뉴의 기능을 확인하는 신호인 경우(S30) 기능을 실행하기 위한 작업을 수행한다. 텐풋 UI는 클릭 이벤트를 생성한다(S41). 생성한 클릭 이벤트는 텐풋 UI에서 포커싱된 윈도우에 송신한다(S42). 어플리케이션의 이벤트 핸들러는 해당 이벤트에 대한 함수를 호출한다(S43). 그리고 호출된 함수가 기능을 수행한다(S44). 한편, S30 단계에서 상하좌우의 메뉴 이동을 요청하는 신호인 경우, 현재 포커싱된 메뉴로부터 포커스를 이동시킨다(S50). 포커싱된 메뉴 또는 버튼은 포커싱되었음을 나타내기 위해, 애니메이션 효과를 줄 수 있다.

도 8은 본 발명의 일 실시예에 따른 인터페이스 변환부의 구성을 나타내는 도면이다.

본 실시예에서 사용되는 '~부'라는 용어, 즉 '~모듈' 또는 '~테이블' 등은 소프트웨어 또는 Field Programmable Gate Array(FPGA) 또는 주문형 반도체(Application Specific Integrated Circuit, ASIC)과 같은 하드웨어 구성요소를 의미하며, 모듈은 어떤 역할을 수행한다. 그렇지만 모듈은 소프트웨어 또는 하드웨어에 한정되는 의미는 아니다. 모듈은 어드레싱할 수 있는 저장 매체에 있도록 구성될 수도 있고 하나 또는 그 이상의 프로세서들을 실행시키도록 구성될 수도 있다. 따라서, 일 예로서 모듈은 소프트웨어 구성요소들, 객체지향 소프트웨어 구성요소들, 클래스 구성요소들 및 태스크 구성요소들과 같은 구성요소들과, 프로세스들, 함수들, 속성들, 프로시저들, 서브루틴들, 프로그램 코드의 세그먼트들, 드라이버들, 펌웨어, 마이크로코드, 회로, 데이터, 데이터베이스, 데이터 구조들, 테이블들, 어레이들, 및 변수들을 포함한다. 구성요소들과 모듈들에서 제공되는 기능은 더 작은 수의 구성요소들과 모듈들로 결합되거나 추가적인 구성요소들과 모듈들로 더 분리될 수 있다.

인터페이스 변환부(700)는 투풋 UI를 제공하는 어플리케이션으로부터 윈도우를 분석하여 텐풋 UI를 지원하도록 변환하는 모듈이다. 이는 소프트웨어를 통해 구현 가능하며, 이외에도 칩에 설계되는 SoC(System On Chip) 방식으로 구현 가능하다.

윈도우 분석부(710)는 어플리케이션을 구성하는 윈도우들에 대한 리소스 정보를 분석한다. 버튼, 판넬, 체크박스 등이 모두 윈도우이므로, 이들의 식별자 및 위치와 크기 정보등을 얻을 수 있다. 템플릿부(720)는 텐풋 UI 템플릿을 제공한다. 버튼의 이미지, 색상에서부터 전체 배경 이미지를 포함한다. 사용자는 미리 정해진 템플릿을 가지고 간단하게 변환하여 사용할 수 있다. 윈도우 변환부(730)는 상기 템플릿부(720)에 저장된 템플릿 정보를 이용하거나, 또는 사용자가 제작한 이미지 또는 색상으로 변환된 리소스를 생성한다. 저장부(740)는 윈도우 변환부(730)를 통해 변환된 인터페이스에 대한 리소스 정보를 저장하여, 추후 어플리케이션 실행시 텐풋 UI를 제공할 수 있도록 한다.

도 9는 본 발명의 일 실시예에 따른 인터페이스 처리부의 구성을 나타내는 도면이다.

인터페이스 처리부(800)는 인터페이스 변환부(700)에서 변환과정을 통해 생성된 인터페이스를 출력하고 사용자로부터 리모트 컨트롤을 통한 신호를 수신하여 처리하는 작업을 수행한다.

디스플레이 제어부(810)는 어플리케이션이 수행될 때 원래의 투풋 UI대신 텐풋 UI로 디스플레이하도록 제어한다. 디스플레이 제어부(810)는 텐풋 UI를 구성하는 리소스를 출력한다. 신호 수신부(820)는 리모트 컨트롤과 같은 원거리에서 입력을 수행하는 장치로부터 제어 신호 또는 입력 신호를 수신하여 이를 처리한다. 주로 상하좌우의 이동 방향에 대한 신호 또는 특정 메뉴를 선택하여 실행하는 신호를 수신한다. 메시지 생성부(830)는 수신한 신호를 해당 윈도우에 대한 이벤트를 보내기 위한 메시지를 생성한다. 메시지 송신부(840)는 상기 메시지를 어플리케이션으로 송신하여, 어플리케이션에 메시지에 따른 기능을 수행하도록 한다.

인터페이스 변환부(700)와 인터페이스 처리부(800)는 같은 컴퓨터 또는 같은 노트북에 존재할 수 있고, 또한 인터페이스 처리부(800)만 별도로 설치될 수 있다. 인터페이스 변환부(700)를 통해 생성된 텐풋 UI를 제공하는 데이터는 인터페이스 처리부(800)에 의해 텐풋 UI를 제공하므로, 인터페이스 변환부(700)없이도 인터페이스 처리부(800) 단독으로 텐풋 UI를

제공할 수 있다. 따라서, 특정 어플리케이션을 개발하는 회사에서 인터페이스 변환부(700)를 통해 텡풋 UI를 제공하는 데이터를 생성하여 인터페이스 처리부(800)와 텡풋 UI를 제공하는 데이터를 함께 배포할 수 있다. 이 경우, 사용자는 인터페이스 처리부(800)만을 가지고 특정 어플리케이션에 대한 텡풋 UI의 효과를 얻을 수 있다. 통상 변환시키는 과정은 개개인의 컴퓨터 또는 노트북 등의 장치에서 이루어질 수도 있고, 개발 업체에서 자사의 어플리케이션의 효율적 사용을 위해 인터페이스 변환부(700)만을 사용할 수 있다.

본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명이 그 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구의 범위에 의하여 나타내어지며, 특허청구의 범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

발명의 효과

본 발명을 구현함으로써 기존의 어플리케이션이 텡풋 UI를 제공하도록 변경할 수 있다.

기존 어플리케이션의 구동 메커니즘을 변화시키지 않도록 인터페이스를 제공하므로, 기존의 텡풋 UI와 같이 근거리에서 사용할 수 있는 어플리케이션을 손쉽게 원거리에서 사용할 수 있다. 또한, 인터페이스를 구성하는 데이터를 인터넷, 디스켓 등을 통해 배포할 수 있으므로 어플리케이션 개발사가 추가로 새로운 어플리케이션을 개발하는데 소요되는 비용과 시간을 절감할 수 있다.

또한 컴퓨터 시스템의 기존의 어플리케이션에 텡풋 UI를 추가하여 손쉽게 디지털 가전과의 융합을 가능하게 할 수 있다.

(57) 청구의 범위

청구항 1.

마우스 또는 키보드로 제어가능한 인터페이스를 포함하는 제 1 모드로 어플리케이션을 디스플레이 하는 단계;

상기 어플리케이션의 외관을 구성하는 데이터를 수집하는 단계;

리모트 컨트롤로 제어가능한 인터페이스를 포함하는 제 2 모드로 상기 어플리케이션을 디스플레이 하도록 상기 데이터를 변환하는 단계; 및

상기 변환된 데이터를 저장하는 단계를 포함하는, 어플리케이션의 인터페이스를 변환하는 방법.

청구항 2.

제 1항에 있어서,

상기 변환하는 단계는

상기 외관을 구성하는 데이터를 출력하는 단계;

상기 외관을 대체할 제 2 모드를 구성하는 템플릿 데이터를 출력하는 단계; 및

상기 템플릿 데이터 중 일부를 선택한 결과를 수신하는 단계를 포함하는, 어플리케이션의 인터페이스를 변환하는 방법.

청구항 3.

제 2항에 있어서,

상기 어플리케이션이 윈도우즈를 운영체제로 하는 경우 상기 외관을 구성하는 데이터는 리소스인, 어플리케이션의 인터페이스를 변환하는 방법.

청구항 4.

제 1항에 있어서,

상기 제 2 모드는 상기 어플리케이션을 1m 이상의 거리에서 제어가 가능하도록 상기 어플리케이션을 디스플레이하고 리모트 컨트롤로부터 제어 신호를 수신하는, 어플리케이션의 인터페이스를 변환하는 방법.

청구항 5.

제 1항에 있어서,

상기 외관을 구성하는 데이터는 상기 어플리케이션의 구성요소를 선택할 경우 제공하는 기능에 대한 정보를 포함하는, 어플리케이션의 인터페이스를 변환하는 방법.

청구항 6.

제 1항에 있어서,

상기 저장하는 단계 이후에,

상기 어플리케이션을 상기 변환된 데이터를 기초로 하여 상기 제 2 모드로 디스플레이하는 단계; 및

리모트 컨트롤로부터 상기 제 2 모드로 디스플레이된 인터페이스의 구성요소를 선택하는 입력 신호를 수신하는 단계를 더 포함하는, 어플리케이션의 인터페이스를 변환하는 방법.

청구항 7.

리모트 컨트롤로 제어가능한 인터페이스를 포함하는 모드로 어플리케이션을 디스플레이하는데 필요한 외관을 구성하는 데이터를 수신하는 단계;

상기 모드로 상기 어플리케이션을 디스플레이하는 단계; 및

리모트 컨트롤로부터 상기 모드로 디스플레이된 인터페이스의 구성요소를 선택하는 입력 신호를 수신하는 단계를 포함하는, 어플리케이션을 디스플레이하는 방법.

청구항 8.

제 7항에 있어서,

상기 모드는 상기 어플리케이션을 1m 이상의 거리에서 제어가 가능하도록 상기 어플리케이션을 디스플레이하고 리모트 컨트롤로부터 제어 신호를 수신하는, 어플리케이션을 디스플레이하는 방법.

청구항 9.

제 7항에 있어서,

상기 데이터는 상기 어플리케이션의 구성요소를 선택할 경우 제공하는 기능에 대한 정보를 포함하는, 어플리케이션을 디스플레이하는 방법.

청구항 10.

마우스 또는 키보드로 제어가능한 인터페이스를 포함하는 제 1 모드로 어플리케이션을 실행하며, 상기 어플리케이션이 상기 제 1 모드로 실행되는 경우, 상기 어플리케이션의 외관을 구성하는 데이터를 수집하는 어플리케이션 분석부;

리모트 컨트롤로 제어가능한 인터페이스를 포함하는 제 2 모드로 상기 어플리케이션을 디스플레이할 수 있도록 상기 데이터를 변환하는 어플리케이션 변환부를 포함하는, 컴퓨터 시스템.

청구항 11.

제 10항에 있어서,

상기 외관을 구성하는 데이터 및 상기 외관을 대체할 상기 제 2 모드를 구성하는 템플릿 데이터를 출력하는 템플릿부를 더 포함하며,

상기 어플리케이션 변환부는 상기 출력된 템플릿 데이터 중 일부를 선택한 값을 수신하여 상기 외관을 구성하는 데이터를 변환하는, 컴퓨터 시스템.

청구항 12.

제 10항에 있어서,

상기 제 2 모드는 상기 어플리케이션을 1m 이상의 거리에서 제어하도록 상기 어플리케이션을 디스플레이하고 제어 신호를 수신하는, 컴퓨터 시스템.

청구항 13.

제 10항에 있어서,

상기 외관을 구성하는 데이터는 상기 외관의 구성요소와 연관된 기능을 제공하는데 필요한 정보를 포함하는, 컴퓨터 시스템.

청구항 14.

제 10항에 있어서,

상기 어플리케이션이 윈도우즈를 운영체제로 하는 경우 상기 외관을 구성하는 데이터는 리소스인, 컴퓨터 시스템.

청구항 15.

리모트 컨트롤이 송신한 입력 신호를 수신하는 신호 수신부;

리모트 컨트롤로 제어가능한 인터페이스를 포함하는 모드로 어플리케이션이 실행될 수 있도록 상기 어플리케이션의 외관을 구성하는 데이터를 수신하는 데이터 수신부; 및

상기 어플리케이션을 상기 모드로 디스플레이하는 디스플레이 제어부를 포함하며,

상기 디스플레이 제어부는 상기 입력 신호에 따라 상기 모드로 실행중인 어플리케이션의 소정의 기능을 실행하는, 컴퓨터 시스템.

청구항 16.

제 15항에 있어서,

상기 모드는 상기 어플리케이션을 1m 이상의 거리에서 제어가 가능하도록 상기 어플리케이션을 디스플레이하고 리모트 컨트롤로부터 제어 신호를 수신하는, 컴퓨터 시스템.

청구항 17.

제 15항에 있어서,

상기 어플리케이션의 외관을 구성하는 데이터는 상기 외관의 구성요소를 실행시킬 경우 제공하는 기능에 대한 정보를 포함하는, 컴퓨터 시스템.

청구항 18.

제 15항에 있어서,

상기 신호 수신부가 수신한 신호를 상기 어플리케이션에 전달될 소정의 메시지로 변환하는 메시지 생성부; 및

상기 생성된 메시지를 어플리케이션에 송신하는 메시지 송신부를 더 포함하는, 컴퓨터 시스템.

청구항 19.

리모트 컨트롤이 송신한 입력 신호를 어플리케이션을 구성하는 소정의 구성 요소에 대한 이벤트 신호로 변환하는 이벤트 처리부;

상기 어플리케이션의 상기 구성요소가 상기 이벤트 신호를 받는 경우 제공하는 기능을 실행하는 제어부를 포함하며,

상기 어플리케이션은 리모트 컨트롤로 제어가능한 인터페이스를 포함하는 모드로 실행되며, 상기 모드는 상기 어플리케이션을 1m 이상의 거리에서 제어가 가능하도록 상기 어플리케이션을 디스플레이하고 리모트 컨트롤로부터 제어 신호를 수신하는, 소프트웨어.

청구항 20.

제 19항에 있어서,

상기 어플리케이션의 외관을 구성하는 데이터는 상기 외관의 구성요소를 실행시킬 경우 제공하는 기능에 대한 정보를 포함하는, 소프트웨어.

청구항 21.

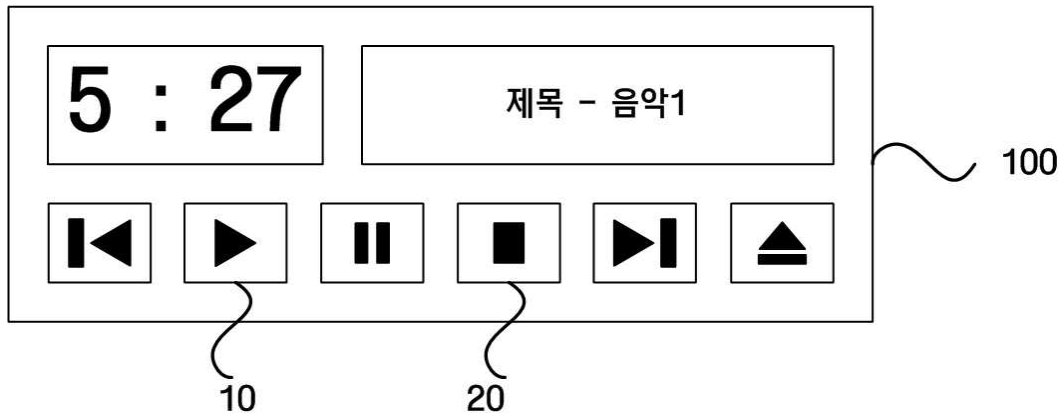
리모트 컨트롤로 제어가능한 인터페이스를 포함하는 모드로 어플리케이션을 실행하도록 상기 어플리케이션의 외관을 구성하는 데이터; 및

상기 어플리케이션이 실행될 때, 상기 데이터로 상기 어플리케이션의 외관을 변환하여 제공하는 인터페이스 처리부를 포함하며,

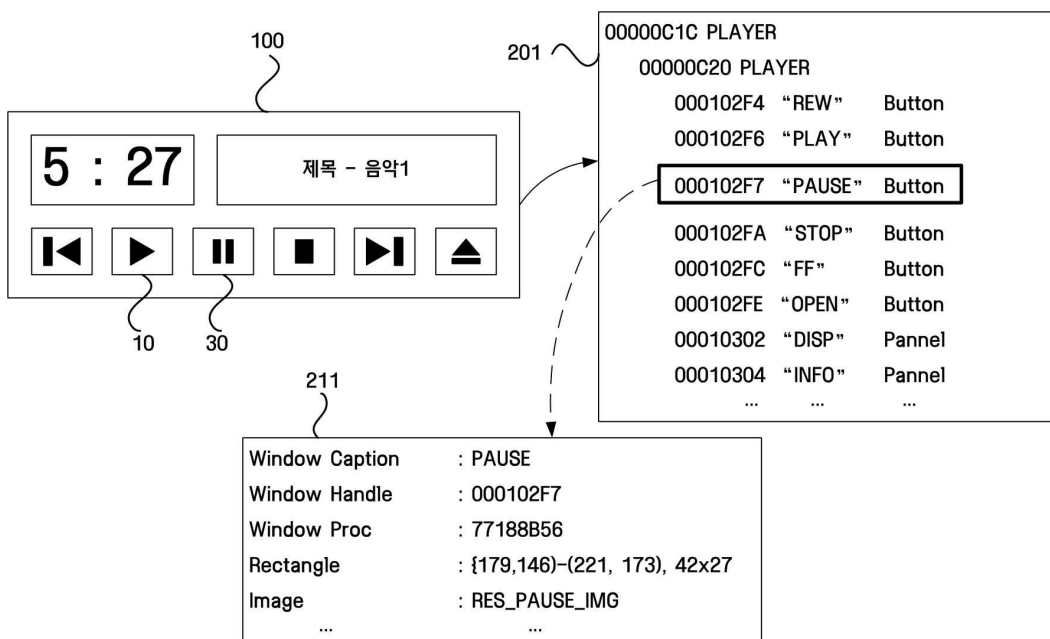
상기 데이터는 상기 어플리케이션의 구성요소를 실행시킬 경우 제공하는 기능에 대한 정보를 포함하는, 저장매체.

도면

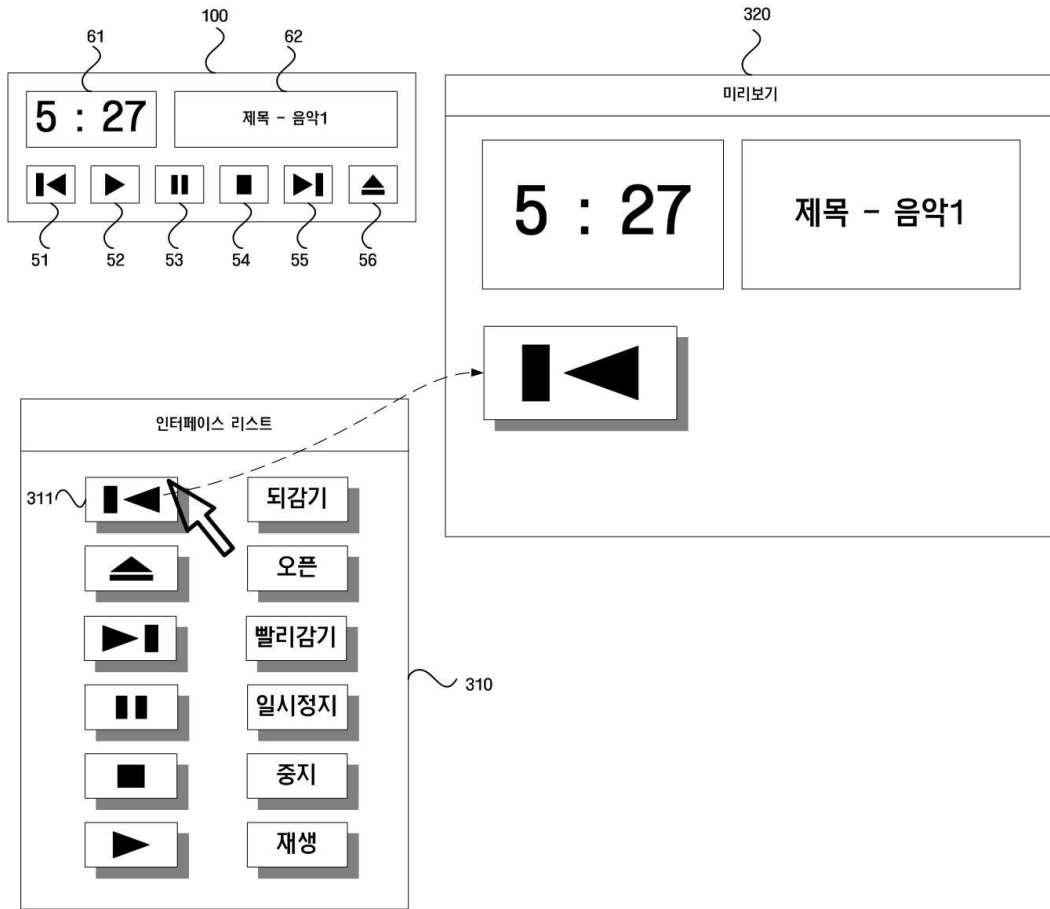
도면1



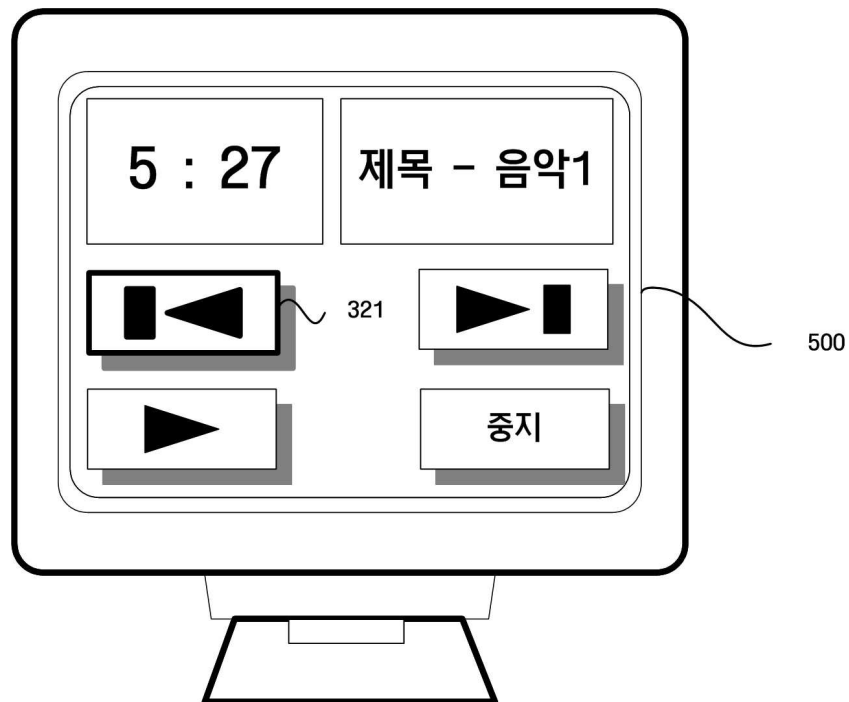
도면2



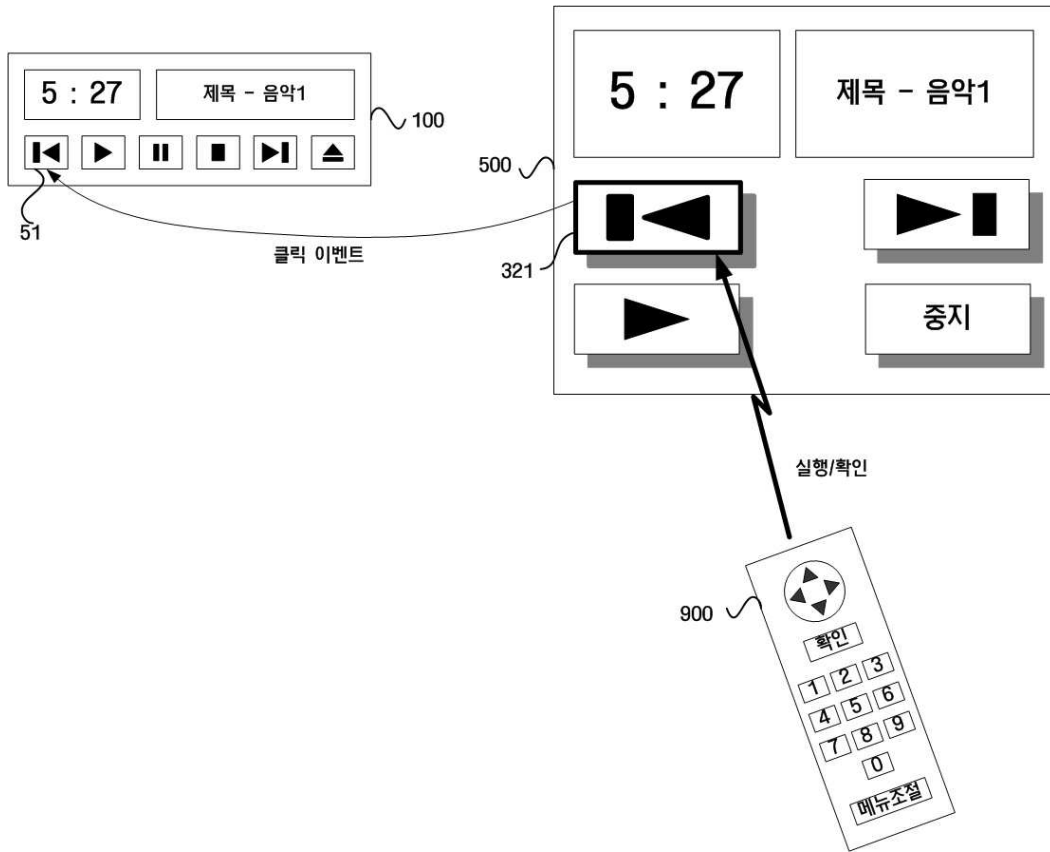
도면3



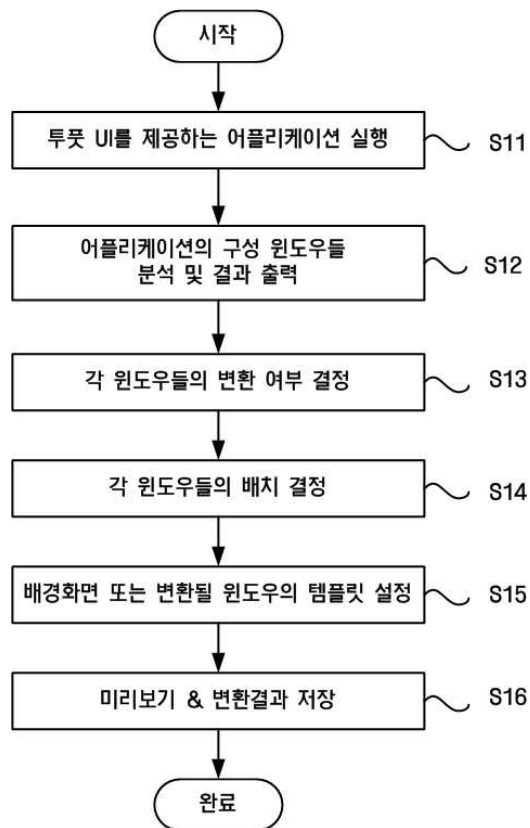
도면4



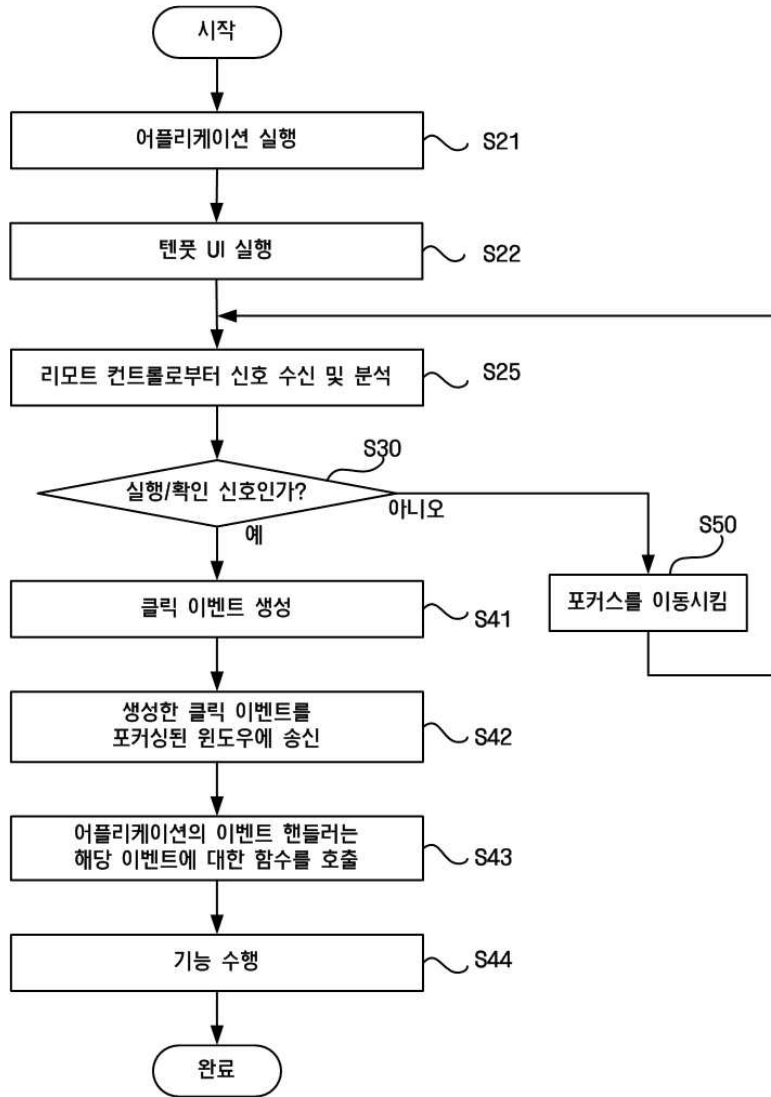
도면5



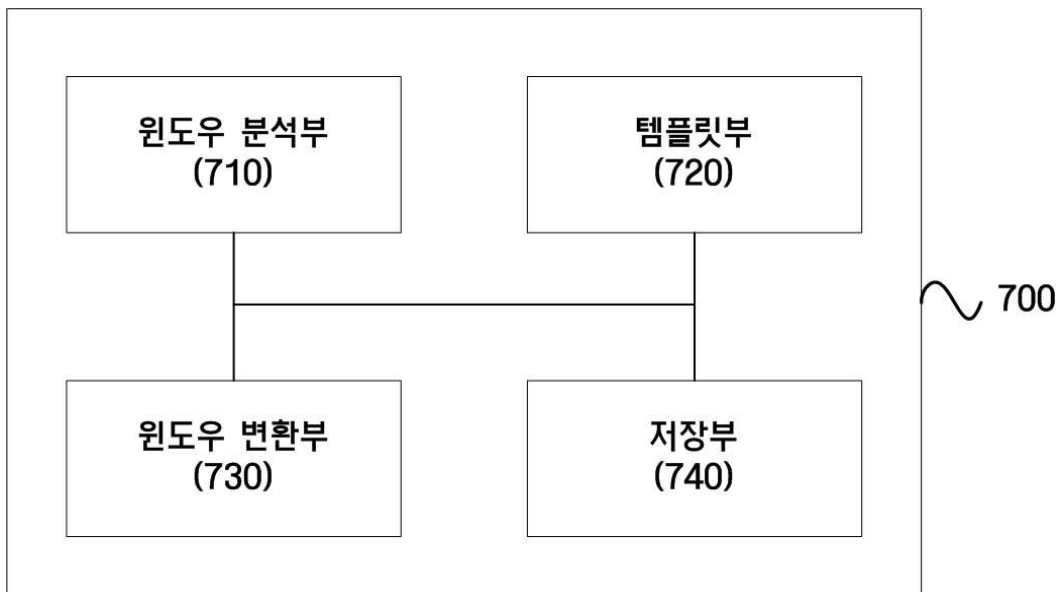
도면6



도면7



도면8



도면9

