



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년05월20일

(11) 등록번호 10-1522150

(24) 등록일자 2015년05월14일

(51) 국제특허분류(Int. Cl.)

G06F 3/01 (2006.01) G06F 3/048 (2006.01)

(21) 출원번호 10-2014-7014024(분할)

(22) 출원일자(국제) 2008년05월23일

심사청구일자 2014년05월26일

(85) 번역문제출일자 2014년05월26일

(65) 공개번호 10-2014-0081892

(43) 공개일자 2014년07월01일

(62) 원출원 특허 10-2009-7024427

원출원일자(국제) 2008년05월23일

심사청구일자 2013년04월29일

(86) 국제출원번호 PCT/US2008/064740

(87) 국제공개번호 WO 2008/148021

국제공개일자 2008년12월04일

(30) 우선권주장

11/904,189 2007년09월26일 미국(US)

60/931,710 2007년05월25일 미국(US)

(56) 선행기술조사문헌

KR1020070040821 A*

US07093034 B2*

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

마이크로소프트 코포레이션

미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이

(72) 발명자

마지브, 맥심

미국 98052-6399 워싱턴주 레드몬드 원 마이크로
소프트 웨이 마이크로소프트 코포레이션 국제 특
허부 내

앤더슨, 스콧 다니엘

미국 98052-6399 워싱턴주 레드몬드 원 마이크로
소프트 웨이 마이크로소프트 코포레이션 국제 특
허부 내

(뒷면에 계속)

(74) 대리인

제일특허법인

전체 청구항 수 : 총 20 항

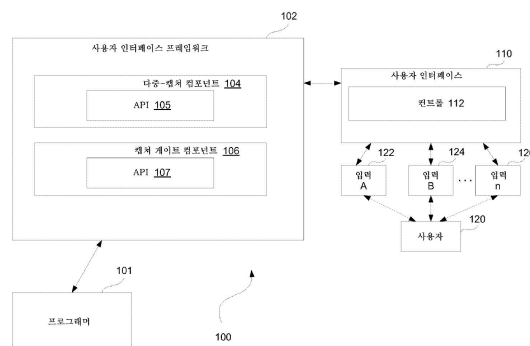
심사관 : 문영재

(54) 발명의 명칭 다중 입력 컨트롤의 선택적인 활성화

(57) 요약

복수의 소스로부터의 입력이 복수의 컨트롤에 캡처될 수 있도록 (예를 들어, 프로그래머에 의해) 구성될 수 있는 컨트롤 시스템이 개시된다. 또는 대안적으로, 본 시스템은 복수의 소스로부터의 입력이 단일 컨트롤로 캡처되도록 구성될 수 있다. 본 시스템은 또한 개별 컨트롤에 대한 충돌하는 이벤트를 필터링하기 위한 일관되고 확장 가능한 기법을 제공한다. 일실시예에서, 본 시스템은 주어진 컨트롤에 대하여 추가적인 입력이 캡처될 수 있는지 여부에 대하여 선택적인 판정을 할 수 있도록 (예를 들어, 프로그래머의 요구된 선호도에 기초하여) 구성될 수 있다.

대표도



(72) 발명자

웨이스, 존 구이도 아트킨스

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제 특허부
내

레비, 로버트

미국 98052-6399 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 국제 특허부
내

명세서

청구범위

청구항 1

입력들을 관리하는 컴퓨터-구현 방법으로서,

제1 입력 메커니즘으로부터 입력 - 상기 입력은 사용자 인터페이스 컨트롤로 향하는 것임 - 을 수신하는 단계와,

제1 사용자 인터페이스 메커니즘이 상기 사용자 인터페이스 컨트롤로 캡처되도록 하는 단계와,

제2 입력 메커니즘으로부터 입력 - 상기 입력은 상기 사용자 인터페이스 컨트롤로 향하는 것임 - 을 수신하는 단계와,

상기 사용자 인터페이스 컨트롤에 할당된 상태를 식별하는 단계와,

상기 식별된 상태에 기초하여, 상기 제2 입력 메커니즘으로부터의 입력을 필터링(filtering out)하거나, 제2 사용자 인터페이스 메커니즘이 상기 사용자 인터페이스 컨트롤로 캡처되도록 하는 단계 - 상기 제1 및 제2 사용자 인터페이스 메커니즘은 상기 사용자 인터페이스 컨트롤에 의해 동시에 캡처됨 -

를 포함하는 컴퓨터-구현 방법.

청구항 2

제1항에 있어서,

제1 입력 메커니즘으로부터 입력을 수신하는 단계는 터치 스크린 인터페이스를 통해 입력을 수신하는 단계를 포함하고, 제2 입력 메커니즘으로부터 입력을 수신하는 단계는 동일한 상기 터치 스크린 인터페이스를 통해 입력을 수신하는 단계를 포함하는

컴퓨터-구현 방법.

청구항 3

제1항에 있어서,

상기 제1 및 제2 입력 메커니즘이 캡처되도록 하는 단계는 각각의 메커니즘으로부터의 후속 이벤트를 상기 사용자 인터페이스 컨트롤로 향하게 하는 단계를 포함하는

컴퓨터-구현 방법.

청구항 4

제1항에 있어서,

상기 제1 입력 메커니즘이 캡처되도록 하는 단계는 상기 제1 메커니즘으로부터의 후속 이벤트를 상기 제1 입력 메커니즘이 해제될 때까지 상기 사용자 인터페이스 컨트롤로 향하게 하는 단계를 포함하는

컴퓨터-구현 방법.

청구항 5

제1항에 있어서,

상기 제1 및 제2 입력 메커니즘에 의해 수신되는 상기 입력들은 상기 제1 및 제2 입력 메커니즘들이 구현되는 동일한 장치와의 별개의 상호작용들(separate interactions)인

컴퓨터-구현 방법.

청구항 6

입력 시스템으로서,

컨트롤을 포함하는 사용자 인터페이스와,

제1 입력 메커니즘 및 제2 입력 메커니즘과,

상기 제1 및 제2 입력 메커니즘 중 하나 또는 둘 다를 상기 컨트롤로 캡처하는 것을 가능하게 하도록 구성되는 사용자 인터페이스 프레임워크와,

어떤 상황에서 상기 제1 및 제2 입력 메커니즘 중 하나 또는 둘 다가 상기 컨트롤로 캡처될 수 있는지를 선택하도록 구성되는 애플리케이션 프로그램 인터페이스

를 포함하되,

상기 사용자 인터페이스는 상기 애플리케이션 프로그램 인터페이스의 선택에 따라 상기 제2 입력 메커니즘으로부터의 입력을 필터링할 수 있는

입력 시스템.

청구항 7

제6항에 있어서,

상기 애플리케이션 프로그램 인터페이스는 상기 제1 및 제2 입력 메커니즘 둘 다 동시에 상기 컨트롤로 캡처되는 것을 가능하게 하는 멀티-캡처 인터페이스를 포함하는

입력 시스템.

청구항 8

제6항에 있어서,

상기 애플리케이션 프로그램 인터페이스는 상기 사용자 인터페이스 프레임워크의 확장인

입력 시스템.

청구항 9

제6항에 있어서,

상기 애플리케이션 프로그램 인터페이스는 개발자로 하여금 상기 컨트롤이 한 번에 하나 이상의 입력 메커니즘을 캡처하지 못하도록 제한할 수 있게 하는 캡처 게이트 인터페이스를 포함하는

입력 시스템.

청구항 10

제6항에 있어서,

상기 애플리케이션 프로그램 인터페이스는 개발자로 하여금 상기 컨트롤을 포함하는 일 군의 컨트롤(a group of

controls)이 한 번에 하나 이상의 입력 메커니즘을 캡처링하지 못하도록 제한하게 하는 캡처 게이트 인터페이스를 포함하는

입력 시스템.

청구항 11

제6항에 있어서,

상기 애플리케이션 프로그램 인터페이스는 획득(acquire) 및 해제(release) 루틴과 연관된 캡처 게이트 클래스를 포함하는

입력 시스템.

청구항 12

제6항에 있어서,

상기 제1 및 제2 입력 메커니즘은 단일의 터치 스크린 입력 장치에 대한 별개의 입력인

입력 시스템.

청구항 13

제6항에 있어서,

상기 제1 및 제2 입력 메커니즘은 별개의 입력 장치들로부터의 별개의 입력인

입력 시스템.

청구항 14

컴퓨팅 장치로서,

복수의 별개의 입력 메커니즘들 각각으로부터 별개의 입력을 수신하는 입력 관리 컴포넌트 - 상기 별개의 입력은 사용자 인터페이스 컨트롤로 향함 - 와,

상기 입력 관리 컴포넌트에 의해 이용되어 상기 복수의 별개의 입력 메커니즘들이 상기 사용자 인터페이스 컨트롤로 동시에 캡처되도록 하는 컴퓨터 프로세서 - 상기 컴퓨터 프로세서는 상기 사용자 인터페이스 컨트롤에 할당된 상태를 식별하고, 상기 식별된 상태에 기초하여 상기 복수의 별개의 입력 메커니즘들 중의 하나 이상으로부터의 입력을 필터링할 수 있음 -

를 포함하는 컴퓨팅 장치.

청구항 15

제14항에 있어서,

상기 별개의 입력 메커니즘들은 동일한 장치 상에서 구현되어 상기 별개의 입력 모두가 상기 장치로부터 수신될 수 있도록 하는

컴퓨팅 장치.

청구항 16

제14항에 있어서,

상기 별개의 입력 메커니즘들은 단일의 터치 스크린 인터페이스의 일부로서 구현된 입력 컴포넌트인 컴퓨팅 장치.

청구항 17

제14항에 있어서,

상기 컴퓨터 프로세서는 상기 입력 관리 컴포넌트에 의해 더 이용되어 상기 별개의 입력 메커니즘들로 하여금 상기 별개의 입력 메커니즘들로부터의 후속 이벤트들을 상기 사용자 인터페이스 컨트롤로 향하도록 하는 컴퓨팅 장치.

청구항 18

제14항에 있어서,

상기 컴퓨터 프로세서는 상기 입력 관리 컴포넌트에 의해 더 이용되어 상기 별개의 입력 메커니즘들로 하여금 동시에 캡처되는 결과로서 상기 별개의 입력 메커니즘들에 대한 후속 입력들을 상기 사용자 인터페이스 컨트롤로 향하도록 하는 컴퓨팅 장치.

청구항 19

제14항에 있어서,

상기 컴퓨터 프로세서는 상기 입력 관리 컴포넌트에 의해 더 이용되어 동시에 캡처되는 결과로서 상기 별개의 입력 메커니즘들에 대한 후속 입력들의 지향을 제한하는 컴퓨팅 장치.

청구항 20

제14항에 있어서,

상기 별개의 입력 메커니즘들에 의해 수신된 상기 별개의 입력들은 동일한 장치와의 별개의 상호작용인 컴퓨팅 장치.

발명의 설명

배경 기술

[0001]

컨트롤은 GUI(graphical user interface)를 구축하기 위해 컴퓨터 프로그래머에 의해 사용되는 인터페이스 요소이다. 상이한 컨트롤은 컴퓨터 프로그램의 사용자와의 상이한 유형의 상호작용을 지원한다. 라디오 버튼, 체크 박스, 드롭-다운 리스트 및 톨바와 같은 일부 컨트롤은 선택-유형 기능(selection-type functionality)을 지원한다. 탭 및 스크롤바와 같은 다른 컨트롤은 네비게이션-유형 기능(navigation-type functionality)을 지원한다. 텍스트 박스 및 콤보 박스와 같은 다른 컨트롤은 텍스트 입력 기능을 지원한다. 톨 팁, 프로그레스바 및 라벨과 같은 다른 컨트롤은 출력 기능을 지원한다. 또한 다른 컨트롤들은 윈도우-지향 기능을 지원한다. 많은 경우에, 컴퓨터 프로그래머는 선택할 수 있는 컨트롤들의 집합을 포함하는 컨트롤 툴킷(control toolkit)에 액세스할 수 있다.

[0002]

최근에 하나 또는 복수의 유저에 의한 복수의 입력의 동시 (또는 실질적으로 동시) 활성화를 지원하는 장치가

소개되고 있다. 그러나, 현재의 컨트롤 시스템은 일반적으로 이러한 유형의 복수 입력 기법을 효율적으로 관리하도록 구성되지 않았다. 많은 현재의 시스템은, 다중-입력(예를 들어, 다중-터치) 시나리오 지원에 관한 기능의 측면에서 특히 효율적이지 않은 단일 입력 컨트롤로 한정된다. 이러한 환경에서 단일 입력 컨트롤이 제대로 작동하지 않고/않거나 고장난 것처럼 보이는 것이 통상적이다.

[0003] 예를 들어, "업" 버튼, "다운" 버튼 및 드래깅 또는 스크롤을 위한 "썸(THUMB)" 슬라이드와 같은 다른 컨트롤들을 포함하는 복합 컨트롤인 종래의 스크롤바를 포함하는 인터페이스를 상상할 수 있다. 많은 다중-입력 시스템에서, 사용자가 업 및 다운 버튼을 동시에 누르려고 시도하면, 버튼은 동시에 업 명령과 다운 명령을 실행한다. 애플리케이션이 충돌하는 이벤트를 효율적으로 처리하도록 구성될 가능성은 적을 것이다.

[0004] 컨트롤 충돌을 방지하기 위한 신뢰할만한 해결책은 전체 애플리케이션에 대한 사용자 입력을 오직 단일 입력으로 제한하는 것이다. 그러나, 이러한 환경에서 다중-입력 기능은 가치가 없다. 따라서, 하나 또는 복수의 사용자로부터의 복수의 입력을 효율적으로 관리하도록 구성된 컨트롤 기법이 요구된다.

[0005] 상기의 개시는 단지 일반적인 배경 정보를 제공하는 것이고 청구된 주요 발명의 범위를 결정하기 위한 용도로 사용하기 위한 것으로 해석되지 않는다.

발명의 내용

[0006] 복수의 소스로부터의 입력을 복수의 컨트롤로 동시에 캡처할 수 있도록 (예를 들어, 프로그래머에 의해) 구성될 수 있는 컨트롤 시스템이 개시된다. 또한 또는 대안적으로, 시스템은 복수의 소스로부터의 입력을 단일 컨트롤로 동시에 캡처할 수 있도록 구성될 수 있다. 시스템은 또한 개개의 컨트롤에 대하여 충돌하는 이벤트를 필터링(filtering out)하기 위한 일관되고 확장가능한 기법을 제공한다. 일실시예에서, 시스템은 주어진 컨트롤에 대하여 추가적인 입력이 캡처될 수 있는지 여부에 대하여 선택적인 판정을 할 수 있도록 (예를 들어, 프로그래머의 요구된 선호도에 기초하여) 구성될 수 있다.

[0007] 본 설명은 아래의 실시예에 더 상세히 개시되는 발명의 발체를 단순화된 형식으로 소개하기 위하여 제공된다. 본 설명은 청구된 주요 발명의 주요한 태양 또는 핵심적인 태양을 식별하기 위한 것으로 해석되지 않으며, 청구된 주요 발명의 범위를 결정하기 위한 목적으로 사용하기 위한 것으로 해석되지 않는다. 청구된 주요 발명은 배경 기술에 개시된 임의의 또는 모든 문제점을 해결하기 위한 구현으로 한정되지 않는다.

도면의 간단한 설명

[0008] 도 1은 컴퓨터 프로그래밍 환경의 개략적인 도면.
 도 2는 예시적인 사용자 인터페이스의 도면.
 도 3은 예시적인 사용자 인터페이스와 관련된 사용자 인터페이스 요소의 트리 표현.
 도 4는 지원되지 않는 입력을 필터링하는 CaptureGate 기능으로 마킹된 사용자 인터페이스 요소의 트리 표현.
 도 5는 다중-캡처 시나리오에 따라 입력을 관리하기 위한 방법을 나타내는 흐름도.
 도 6은 캡처 게이트 시나리오에 따라 입력을 관리하기 위한 방법을 나타내는 흐름도.
 도 7은 컴퓨팅 시스템 환경의 개략적인 도면.

발명을 실시하기 위한 구체적인 내용

[0009] 도 1은 컴퓨터 프로그래밍 환경(100)의 개략적인 도면이다. 환경(100)에서 프로그래머는 사용자 인터페이스(110)를 생성하기 위해 사용자 인터페이스 프레임워크(102)와 상호작용한다. 인터페이스(110)는 복수의 컨트롤(112)을 포함한다.

[0010] 도 1에서, 사용자는 입력 메커니즘(122, 124 및 126)의 하나 이상의 방식에 의해 사용자 인터페이스(110)와 상호작용하는 것으로 도시된다. 임의의 또는 모든 입력 메커니즘(122, 124 및 126)의 조작에 의해, 사용자(120)는 프레임워크(102)와의 상호작용 방식에 의해 프로그래머(101)에 의해 생성되고 구성되는 컨트롤(112)과 상호작용할 수 있다. 일실시예에서, 복수의 입력이 입력(122, 124 및 126)의 하나 이상으로부터 동시에 (또는 실질

적으로 동시에) 전달될 수 있고, 오직 단일 사용자로부터 비롯될(originate) 필요는 없다. 입력 메커니즘(122, 124 및 126)은 이에 한정되지는 않지만, 예를 들어 마우스 입력 메커니즘, 터치스크린 입력 메커니즘, 음성 입력 메커니즘, 레이저 포인터 입력 메커니즘 또는 사용자가 입력을 전달할 수 있도록 하는 임의의 다른 메커니즘이 될 수 있다는 것을 유의해야 한다. 이론적으로, 당연히 도시된 세 개의 입력 메커니즘보다 더 많은 입력 메커니즘이 존재할 수 있다. 더 나아가, 모든 세 개의 입력(122, 124 및 126)은 단일 입력 메커니즘으로부터 비롯된 개개의 입력일 수 있다(예를 들어, 단일 터치스크린 메커니즘에 대한 복수의 동시 입력).

[0011] 본 명세서에서 사용되는 용어 "동시"는 엄밀한 동시로서 해석되지 않는다는 것을 유의해야 한다. 충돌은 입력 사이의 임의의 시간 기간, 예를 들어 수 초 내, 밀리초 내 또는 다른 적절한 및/또는 설정된 시간 기간 내에서 예상된다.

[0012] 프레임워크(102)는 예시적으로 다중-캡처 컴포넌트(104) 및 캡처 게이트 컴포넌트(106)를 포함한다. 대응하는 API(application program interface; 105 및 107)를 포함하는 컴포넌트(104 및 106)는 다중-입력 컨트롤에 대한 특화된 지원을 수용하기 위한 프레임워크(102) 기능의 확장을 지원한다. 다중-캡처 컴포넌트(104)는 예시적으로 컨트롤(112)이 복수의 입력 메커니즘(예를 들어, 메커니즘(122, 124 및/또는 126)의 하나 이상)으로부터 입력을 캡처할 수 있도록 하고 장치로부터의 모든 이벤트를 컨트롤로 전향(redirect)할 수 있도록 한다. 대안적으로 또는 추가적으로, 다중-캡처 컴포넌트(104)는 복수의 컨트롤(112)이 복수의 입력 메커니즘(예를 들어, 복수의 입력 메커니즘으로부터의 동시 입력)으로부터 입력을 캡처할 수 있도록 하고 장치로부터 이벤트를 적절한 컨트롤로 분배하도록 한다. 캡처 게이트 컴포넌트(106)는 프로그래머(101)가 충돌하는 이벤트를 필터링하기 위한 체계화된 방법을 제공한다. 일실시예에서, 컴포넌트(106)는 예시적으로 특정한 컨트롤에 대한 충돌 상황을 방지하기 위하여 프로그래머(101)가 사용자 입력을 제한하는 컨트롤 전용 로직을 정의할 수 있도록 한다.

[0013] 사용자 프레임워크(102) 내의 컴포넌트(104 및 106) 기능의 더 상세한 개시로 돌아가기 전에, 우선 적어도 단일-입력 컨트롤 구현을 위한 현재 기술의 간략한 개관을 살펴볼 필요가 있다. 통상적인 시나리오에서, 사용자 인터페이스 프레임워크는 입력 장치의 최초 캡처의 방법을 통해 특정한 컨트롤로 입력을 전향하고 이를 대응하는 사용자 인터페이스 요소(예를 들어, 대응하는 컨트롤)로 결합하는 방법을 제공한다. 따라서, 입력 장치가 캡처될 때, 입력 경로(input pipeline)는 캡처 시간 기간 동안의 상기 장치의 실제 위치와 관계없이 (예를 들어, 마우스가 캡처될 때 커서의 위치와 관계없이) 입력을 상기 장치로부터 캡처 컨트롤로 전향시킨다.

[0014] 통상적인 단일 입력 시나리오의 예시는 마우스 입력 장치와의 상호작용을 지원하는 통상적인 페인팅 애플리케이션(painting application)에서 발견될 수 있다. 사용자가 캔버스 상에서 "그리기"를 하려고 할 때, 그 또는 그녀는 커서를 캔버스 상에서 이동하고, 예를 들어 마우스 버튼을 누르고 있음으로써 "마우스 다운" 이벤트를 개시한다. 마우스 다운 이벤트 시에, 애플리케이션은 마우스 장치를 캔버스 컨트롤로 캡처한다. 이러한 캡처는, 예를 들어 마우스 버튼을 누르고 있는 것을 해제함으로써 "마우스 업" 형식의 사용자-개시 해제 시에 종료된다. 캡처 시간 기간 동안, 모든 마우스 이벤트(커서 이동 등)는 커서 위치가 캔버스 영역의 바깥(예를 들어, 툴바 사용자 인터페이스 요소 상)에 있을 때 캔버스 컨트롤로 전달된다.

[0015] 컴포넌트(104)는 예시적으로 모든 포인터-형 장치("터치"를 포함함)를 독립적인 물리적 또는 가상의 장치로 취급하고 컨트롤 라이브러리(또는 애플리케이션)가 임의의 이러한 장치를 독립적으로 캡처하도록 함으로써 종래의 접근으로 확장한다. 따라서, 일실시예에서 컨트롤은 발생하는 모든 접촉을 캡처하여 모든 대응하는 후속적인 이벤트를 컨트롤로 라우팅하기 위하여 다중-터치-인식(multi-touch-aware)으로 구성된다.

[0016] 일실시예에서, 프레임 워크(102)는 필수적이지는 않지만 단일 컨트롤로 입력 장치의 캡처를 한정하기 위한 제한을 가하도록 구성된다. 이러한 제한은 잠재적인 모호성을 방지할 수 있다. 그러나, 분명히 단일 컨트롤이 복수의 입력 장치를 캡처하도록 구성될 수 있다는 것이 예상된다. 일실시예에서, 프레임워크(102)는 캡처가 계속되거나 또는 해제되었다는 승인으로서 "GotCapture" 및 "LostCapture" 이벤트를 발생시키도록 구성된다. 일실시예에서, 장치가 분리될 때(예를 들어, 손가락이 터치-장치를 떠날 때, 마우스 버튼이 해제될 때 등), 프레임워크(102)는 캡처를 자동으로 해제하도록 구성된다.

[0017] 캡처 게이트 컴포넌트(106)는 예시적으로 충돌 이벤트의 필터링을 지원한다. 이러한 기능의 개시를 위한 컨텍스트를 제공하기 위하여, 예시가 제공된다. 도 2는 예시적인 사용자 인터페이스(200)의 도시이다. 인터페이스(200)는 예시적으로 사용자가 팔레트 영역(pallet area; 204)에서 그림을 생성할 수 있도록 하는 페인팅 애플리케이션의 부분이다. 버튼(202)의 그룹은 사용자가 상이한 기능들 사이에서 스위칭할 수 있도록 한다(예를 들어, 페인트브러시, 지우개, 확대경 등)(예시적인 네 개의 상이한 버튼(202)이 도 2에 도시됨). 스크롤바(206)는 사용자가 팔레트 영역을 원하는 대로 상하로 이동시킬 수 있도록 한다. 스플리터(splitter; 208)는 인

터페이스의 좌측과 우측을 분리한다. 본 기술분야의 당업자는 인터페이스(200)와 실질적으로 동일한 인터페이스를 가지는 애플리케이션에 친숙할 가능성이 높다.

- [0018] 도 3은 인터페이스(200)의 예시적인 요소(예를 들어, 컨트롤 등)의 트리 표현이다. 예시적으로 도 3에 도시된 요소의 일부는 다중 장치(또는 다중 터치)에 의해 동시에 동작할 수 있으나 일부는 그렇지 않을 수도 있다. 예를 들어, 예시적으로 인터페이스(200)의 요소의 기능은 다음의 원칙에 기초하는 상호작용을 지원하도록 구성되는 것이 가장 바람직하다고 가정한다.
- [0019] 1. 각 버튼(202)은 오직 단일 입력에 의해 활성화될 수 있다(예를 들어, 단일 터치에 의해 눌러짐). 이는 버튼 동작을 상대적으로 명확하게 한다.
- [0020] 2. 버튼(202)은 본질적으로 "도구 선택기"이고, 페인트 애플리케이션은 한번에 오직 하나의 활성화 도구를 지원하는 제한을 가지는 것으로 가정하면, 오직 단일 입력을 위한 이러한 버튼을 포함하는 그룹박스(210)에 대한 입력을 제한하는 것이 의미가 있다.
- [0021] 3. 스플리터(208)는 예시적으로 사용자에게 의해 좌측 또는 우측 판넬을 크기조정(resize)하도록 드래깅될 수 있다. 마찬가지로 이는 오직 단일 입력을 통한 조작으로 제한하는 것이 의미가 있다.
- [0022] 4. 스크롤바(206)는 프레스 업 및 프레스 다운 및/또는 썸 슬라이더의 드래깅과 같은 충돌 동작을 방지하기 위하여 오직 단일 입력만을 허용해야 한다.
- [0023] 5. 스크롤바뿐만 아니라 썸 슬라이드 내의 업 및 다운 버튼은 각각의 단일 입력으로 작업하기 위하여 설정될 수 있다. 또한, 이러한 요소가 복합 컨트롤(스크롤바)의 일부이면, 넓은 스크롤바(broader ScrollBar)는 예시적으로 동시에 프레스 업 및 다운 및 썸 슬라이드의 드래깅을 하는 것을 방지하기 위한 다른 제한을 추가한다.
- [0024] 6. 인터페이스(200)와 관련된 나머지 컨트롤은 예시적으로 다중 입력을 허용한다. 예를 들어, 사용자는 한 손으로 툴을 선택하는 동안, 다른 손으로 캔버스 영역(204)에 복수의 손가락으로 계속 그릴 수 있다.
- [0025] 일실시예에서, 캡처 게이트 컴포넌트(106)는 프로그래머(101)가 다양한 컨트롤이 복수 입력을 관리하는 방식을 특정할 수 있도록 한다. 컴포넌트(106)는 예시적으로 추상 클래스 "CaptureGate"에 적어도 두 개의 동작, 주로 1) Try to Acquire 및 2) Release를 제공한다. 프레임워크(102)와의 상호작용을 통해 프로그래머(101)는 주어진 컨트롤에 대한 적절한 구현이 가능하도록 CaptureGate 기능을 레버리지(leverage)할 수 있다.
- [0026] 일실시예에서, 캡처 게이트 컴포넌트(106)는 특정한 사용자 인터페이스 요소와 관련하여 CaptureGate가 설정될 수 있도록 하는 지원을 제공하여 프레임워크(102)를 확장시킨다. 더 나아가, 일실시예에서 캡처 게이트 컴포넌트(106)는 특정한 요소에 대하여 (부모 체인(parent chain)에 따라) 최상위 CaptureGate를 탐색하도록 구성된다.
- [0027] 본 기술분야의 당업자는 캡처 게이트 기능이 주어진 사용자 인터페이스 프레임워크에 통합되는 방식에 대한 세부사항이 주어진 프레임워크와 관련된 구현 세부사항에 기초하여 변경될 수 있다는 것을 이해할 수 있다. 다시 말하면, 캡처 게이트 구현의 세부사항은 프레임워크 종속적이 될 가능성이 높다. 예시적으로, 워싱턴, 레드몬드의 마이크로소프트 사에 의해 제공되는 제품과 관련된 그래픽 서브시스템인 WPF(Windows Presentation Foundation)는 시각적 트리의 임의의 사용자 인터페이스 요소 상에서 설정되고 검색될 수 있는 부착된 속성을 정의하기 위한 기능에 대한 지원을 제공한다. 이러한 경우에, CaptureGate는 부착된 속성의 값으로서 구현될 수 있다. 이는 특정한 프레임워크 내에서의 구현의 단지 하나의 예시이다.
- [0028] 통상적으로, 사용자 인터페이스 프레임워크는 Capture() 및 Release() 메소드를 통해 캡처를 조작하기 위한 기능에 대한 지원을 나타낸다. 일실시예에서, 캡처 게이트 컴포넌트(106)는 최상위 CaptureGate를 탐색하여 이를 획득하는 것을 시도하기 위한 기능으로 이러한 방법을 확장한다. (예를 들어, Capture()에 대해) 획득이 실패하면 예시적으로 오류가 반환된다. 추가적으로, 컴포넌트(106)는 (예를 들어, Release()에 대하여) 최상위 CaptureGate를 탐색하여 이를 해제하도록 구성된다.
- [0029] 더 나아가, 일실시예에서 캡처 게이트 컴포넌트(106)는 프레임워크 레벨에서 예시적으로 특화된 기능을 가지는 특화된 CaptureGate에 대한 구현을 제공한다. 예를 들어, SingleCaptureGate는 예시적으로 획득을 오직 단일 장치로 제한한다. SingleCaptureGate는 본질적으로 추상 CaptureGate 클래스의 특성화이다. 캡처 게이트 컴포넌트(106)는 예시적으로 컨트롤 작성자에게 임의의 동작을 제공하기 위하여 동일한 인터페이스(Acquire 및 Release)로 다른 CaptureGate를 구현하는 것을 선택하기 위한 기능에 대한 권한을 부여한다. 예를 들어, 버튼이 N개의 동시 접촉에 의해 눌러져야 하는 것이 요구되면, 컨트롤 작성자는 동시에 N개의 입력을 허용하고 나머

지를 필터링하도록 허용하는 CaptureGate를 제공할 수 있다. 본 기술분야의 당업자는 본 명세서에 제공된 예시적인 특성화가 본 발명의 범위 내의 CaptureGate 최적화의 많은 예시의 단지 일부라는 것을 받아들일 수 있다.

[0030] 따라서, 일실시예에서 컨트롤 클래스의 작성자(예를 들어, 프로그래머(101))가 컨트롤 UI 요소 상의 CaptureGate의 설정에 의해 클래스를 "단일 입력 전용"으로서 지정할 수 있도록 하기 위한 지원이 제공된다. 상기의 예시에서, 모든 RadioButton, Up 및 Down 버튼은 Button 생성자에서 그 위에 SingleCaptureGate 설정을 할 수 있다. 버튼(또는 임의의 컨트롤)이 입력을 캡처하려고 시도할 때, 캡처 방법은 자동으로 (CaptureGate의 소스로서 사용되는 RadioButton1, GroupBox에 대한) 최상위 CaptureGate를 탐색하여 이를 획득하기 위해 시도한다. 일실시예에서, Capture가 오류 표시를 반환하면, 컨트롤은 그 장치로부터의 입력과 이로부터의 후속적인 이벤트를 무시한다. 도 2 및 3의 예시에서 계속하여, 도 4는 지원되지 않는 입력을 필터링하기 위해 CaptureGate로 마킹된(어두운 원으로 표시됨) 특정한 사용자 인터페이스 요소를 보여준다.

[0031] 일실시예에서, CaptureGate를 나타내는 API(107)는 다음과 같다.

```
Public abstract class CaptureGate
{
    protected virtual bool TryAcquire(InputDevice, UIElement);
    protected virtual void Release(InputDevice);
    public static void SetCaptureGate(UIElement, CaptureGate);
}
```

[0032] 일실시예에서, 본 명세서에 개시된 다른 방법은 사용자 인터페이스 프레임워크에 내재적(internal)이고, 예를 들어 대부분의 사용자 인터페이스 프레임워크가 제공하는 Capture() 및 Release()를 통해 간접적으로 호출된다. 언급된 모든 것들에 대해, 본 기술분야의 당업자는 또한 본 발명의 범위 내의 다른 구현 기법이 있다는 것을 받아들일 수 있다.

[0034] 도 5는 본 명세서에 개시된 다중-캡처 기능에 부합하는 다중-캡처 시나리오에 따라 입력을 관리하기 위한 방법(500)의 일실시예를 나타내는 블록 흐름도이다. 블록(502)에 따르면, 입력은 제1 입력 메커니즘으로부터 수신되고, 입력은 사용자 인터페이스 컨트롤로 향한다. 블록(504)에 따르면, 제1 입력 메커니즘은 사용자 인터페이스 컨트롤로 캡처된다. 블록(506)에 따르면, 입력은 제2 입력 메커니즘으로부터 수신되고, 입력은 또한 사용자 인터페이스 컨트롤로 향한다. 블록(508)에 따르면, 제2 입력 메커니즘은 제1 입력 메커니즘과 동시에 사용자 인터페이스 컨트롤로 캡처된다.

[0035] 도 6은 본 명세서에 개시된 캡처 게이트 기능에 따라 입력을 관리하기 위한 방법(600)의 일실시예를 나타내는 블록 흐름도이다. 블록(602)에 따르면, 입력은 입력 메커니즘으로부터 수신된다. 입력은 사용자 인터페이스 컨트롤로 향한다. 블록(604)에 따르면, CaptureGate 상태가 사용자 인터페이스 컨트롤에 할당되었는지 여부에 대한 판정이 행해진다. 마지막으로, 블록(606)에 따르면 예시적으로 상기의 판정에 기초하여, 입력 메커니즘이 사용자 인터페이스 컨트롤로 캡처되거나 또는 캡처되지 않는다. 물론, 이러한 최종 판정의 일부는 하나 이상의 입력 메커니즘이 이미 사용자 인터페이스로 캡처되었는지 여부에 따라 다를 수 있고, 또한 적용가능한 CaptureGate 상태와 관련된 파라미터에 따라 다를 수 있다.

[0036] 도 7은 본 실시예가 구현될 수 있는 적절한 컴퓨팅 시스템 환경(700)의 예시를 도시한다. 컴퓨팅 시스템 환경(700)은 오직 적절한 컴퓨팅 환경의 일예시이고 사용 또는 기능의 범위에 관한 어떠한 한정을 제시하는 것으로 해석되지 않는다. 또한 컴퓨팅 환경(700)은 예시적인 운영 환경(700)에 도시된 컴포넌트의 임의의 하나 또는 조합과 관련된 종속성 또는 요건을 가지는 것으로 해석되지 않는다.

[0037] 본 실시예는 수많은 다른 범용 또는 전용 컴퓨팅 시스템 환경 또는 구성에서 동작할 수 있다. 본 명세서에 개시된 실시예에서 사용하기에 적절한 잘 알려진 컴퓨팅 시스템, 환경 및/또는 구성의 예시는 이에 한정되지는 않지만, 개인용 컴퓨터, 서버 컴퓨터, 휴대용 또는 랩탑 장치, 셋탑 박스, 멀티프로세서 시스템, 마이크로프로세서-기반 시스템, 프로그램가능한 소비자 전자기기, 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터, 전화 시스템, 임의의 상기 시스템 또는 장치를 포함하는 분산 컴퓨팅 환경 등을 포함한다.

[0038] 본 실시예는 컴퓨터에 의해 실시되는 프로그램 모듈과 같은 컴퓨터-실행가능 명령어의 일반적인 컨텍스트에서 개시될 수 있다. 일반적으로, 프로그램 모듈은 특정한 작업을 수행하거나 또는 특정한 추상 데이터 유형을 구현하는 루틴, 프로그램, 객체, 컴포넌트, 데이터 구조 등을 포함한다. 본 기술분야의 당업자는 임의의 형식의

컴퓨터 판독가능 매체 상에 기록될 수 있는 프로세서 실행가능 명령어로서 본 명세서에 제공되는 개시 및 도면을 구현할 수 있다.

[0039] 본 실시예는 또한 통신 네트워크를 통해 연결되는 원격 프로세싱 장치에 의해 작업이 수행되는 분산 컴퓨팅 환경에서 실시될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈은 메모리 저장 장치를 포함하는 로컬 및 원격 컴퓨터 저장 매체 모두에 위치할 수 있다.

[0040] 도 7을 참조하면, 본 실시예를 구현하기 위한 예시적인 시스템은 컴퓨터(710) 형식의 범용 컴퓨팅 장치를 포함한다. 컴퓨터(710)의 컴포넌트는 이에 한정되지는 않지만, 프로세싱 유닛(720), 시스템 메모리(730) 및 시스템 메모리를 포함하는 다양한 시스템 컴포넌트를 프로세싱 유닛(720)과 연결하는 시스템 버스(721)를 포함할 수 있다. 시스템 버스(721)는 메모리 버스 또는 메모리 컨트롤러, 주변 버스 및 임의의 다양한 버스 아키텍처를 사용하는 로컬 버스를 포함하는 임의의 다양한 유형의 버스 구조가 될 수 있다. 한정이 아닌 예시로서, 이러한 아키텍처는 ISA(industry standard architecture) 버스, MCA(micro channel architecture) 버스, EISA(enhanced ISA) 버스, VESA(video electronics standards association) 로컬 버스 및 메자닌(mezzanine) 버스로서도 알려진 PCI(peripheral component interconnect) 버스를 포함한다.

[0041] 컴퓨터(710)는 통상적으로 다양한 컴퓨터 판독가능 매체를 포함한다. 컴퓨터 판독가능 매체는 컴퓨터(710)에 의해 액세스될 수 있는 임의의 사용가능한 매체가 될 수 있고 휘발성 및 비휘발성 매체, 착탈식 및 비-착탈식 매체를 모두 포함한다. 한정이 아닌 예시로서, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 포함할 수 있다. 컴퓨터 저장 매체는, 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 다른 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 착탈식 및 비-착탈식 매체를 모두 포함한다. 컴퓨터 저장 매체는 이에 한정되지는 않지만, RAM, ROM, EEPROM, 플래시 메모리 또는 다른 메모리 기술, CD-ROM, DVD(digital versatile disk) 또는 다른 광디스크 저장소, 마그네틱 카세트, 마그네틱 테이프, 마그네틱 디스크 저장소 또는 다른 마그네틱 저장 장치 또는 원하는 정보를 저장하는데 사용될 수 있고 컴퓨터(710)에 의해 액세스될 수 있는 임의의 다른 매체를 포함한다. 통신 매체는 통상적으로 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 반송파 또는 다른 전송 메커니즘과 같은 변조된 데이터 신호의 다른 데이터를 구현하고 임의의 정보 전달 매체를 포함한다. 용어 "변조된 데이터 신호"는 자신의 특정 집합의 하나 이상을 가지고 신호의 정보를 인코딩하기 위한 방식으로 변경된 신호를 의미한다. 한정이 아닌 예시로서, 통신 매체는 유선 네트워크 또는 직접-유선 연결과 같은 유선 매체 및 어쿠스틱, RF, 적외선 및 다른 무선 매체와 같은 무선 매체를 포함한다. 상기의 임의의 조합이 또한 컴퓨터 판독가능 매체의 범위 내에 포함될 것이다.

[0042] 시스템 메모리(730)는 ROM(read only memory; 731) 및 RAM(random access memory; 732)과 같은 휘발성 및/또는 비휘발성 메모리 형식의 컴퓨터 저장 매체를 포함한다. 예를 들어, 시동 시에 컴퓨터(710) 내의 구성요소 사이의 정보의 전달을 도와주는 기본 루틴을 포함하는 BIOS(basic input/output system; 733)는 통상적으로 ROM(731)에 저장된다. RAM(732)은 통상적으로 즉시 액세스 가능하고/하거나 프로세싱 유닛(720)에 의해 현재 동작 중인 데이터 및/또는 프로그램 모듈을 포함한다. 한정이 아닌 예시로서, 도 7은 운영 체제(734), 애플리케이션 프로그램(735), 다른 프로그램 모듈(736) 및 프로그램 데이터(737)를 도시한다. 프로그램(735)은 본 명세서에 실시예가 상세히 개시되는 사용자 인터페이스 개발 환경(100)의 하나 이상의 컴포넌트를 포함할 수 있는 것으로 도시된다. 이는 이러한 컴포넌트가 구현되는 환경의 단지 하나의 예시이다. (예를 들어, 프로그램(745), 운영 체제(734) 또는 프로그램(785)의 부분으로서) 다른 구현이 또한 본 발명의 범위 내에서 고려될 수 있다.

[0043] 컴퓨터(710)는 또한 다른 착탈식/비착탈식, 휘발성/비휘발성 컴퓨터 저장 매체를 포함할 수 있다. 오직 예시적으로, 도 7은 비-착탈식, 비휘발성 마그네틱 매체로부터 판독하거나 또는 이에 기록하는 하드디스크 드라이브(741), 착탈식, 비휘발성 마그네틱 디스크(752)로부터 판독하거나 또는 이에 기록하는 마그네틱 디스크 드라이브(751) 및 CD ROM 또는 다른 광매체와 같은 착탈식, 비휘발성 광디스크(756)로부터 판독하거나 또는 이에 기록하는 광디스크 드라이브(755)를 도시한다. 예시적인 운영 환경에서 사용될 수 있는 다른 착탈식/비-착탈식, 휘발성/비휘발성 컴퓨터 저장 매체는 이에 한정되지는 않지만, 마그네틱 테이프 카세트, 플래시 메모리 카드, DVD(digital versatile disk), 디지털 비디오 테이프, 솔리드 스테이트 RAM, 솔리드 스테이트 ROM 등을 포함한다. 하드디스크 드라이브(741)는 통상적으로 인터페이스(740)와 같은 비-착탈식 메모리 인터페이스를 통해 시스템 버스(721)와 연결되고, 마그네틱 디스크 드라이브(751) 및 광디스크 드라이브(755)는 통상적으로 인터페이스(750)와 같은 착탈식 메모리 인터페이스에 의해 시스템 버스(721)와 연결된다.

[0044] 상기에 개시되고 도 7에 도시된 드라이브 및 이들의 관련된 컴퓨터 저장 매체는 컴퓨터(710)를 위한 컴퓨터 판

독가능 명령어, 데이터 구조, 프로그램 모듈 및 다른 데이터의 저장소를 제공한다. 도 7에서, 예시적으로 하드 디스크 드라이브(741)가 운영 체제(744), 애플리케이션 프로그램(745), 다른 프로그램 모듈(746) 및 프로그램 데이터(747)를 저장하는 것으로 도시된다. 이러한 컴포넌트는 운영 체제(734), 애플리케이션 프로그램(735), 다른 프로그램 모듈(736) 및 프로그램 데이터(737)와 동일하거나 또는 이와 상이할 수 있다는 것을 유의해야 한다. 운영 체제(744), 애플리케이션 프로그램(745), 다른 프로그램 모듈(746) 및 프로그램 데이터(747)는, 최소한 이들이 상이한 카피임을 나타내기 위하여 본 명세서에서 상이한 번호가 주어졌다.

[0045]

사용자는 키보드(762), 마이크로폰(763) 및 마우스, 트랙볼 또는 터치 패드와 같은 포인팅 장치(761)와 같은 입력 장치를 통해 컴퓨터(710)로 명령 및 정보를 입력할 수 있다. 다른 입력 장치(도시되지 않음)는 조이스틱, 게임 패드, 위성 접시, 스캐너 등을 포함할 수 있다. 이러한 모든 입력 장치는 대부분 시스템 버스와 연결된 사용자 입력 인터페이스(760)를 통해 프로세싱 유닛(720)과 연결되지만, 병렬 포트, 게임 포트 또는 USB(universal serial bus)와 같은 다른 인터페이스 및 버스 구조에 의해 연결될 수도 있다. 모니터(791) 또는 다른 유형의 디스플레이 장치는 또한 비디오 인터페이스(790)와 같은 인터페이스를 통해 시스템 버스(721)와 연결된다. 모니터뿐만 아니라, 컴퓨터는 또한 출력 주변 인터페이스(790)를 통해 연결될 수 있는 스피커(797) 및 프린터(796)와 같은 다른 주변 출력 장치를 포함할 수 있다.

[0046]

컴퓨터(710)는 원격 컴퓨터(780)와 같은 하나 이상의 원격 컴퓨터와의 논리적 연결을 사용해서 분산 환경에서 동작할 수 있다. 원격 컴퓨터(780)는 개인용 컴퓨터, 휴대용 장치, 서버, 라우터, 네트워크 PC, 피어 장치(peer device) 또는 다른 공통 네트워크 노드가 될 수 있고, 통상적으로 컴퓨터(710)와 관련하여 상기에 개시된 구성요소의 대다수 또는 전부를 포함한다. 도 7에 도시된 논리적 연결은 LAN(local area network; 771) 및 WAN(wide area network; 773)을 포함하지만, 또한 다른 네트워크를 포함할 수 있다. 이러한 네트워킹 환경은 사무실, 기업형 컴퓨터 네트워크, 인트라넷 및 인터넷에서 통상적이다.

[0047]

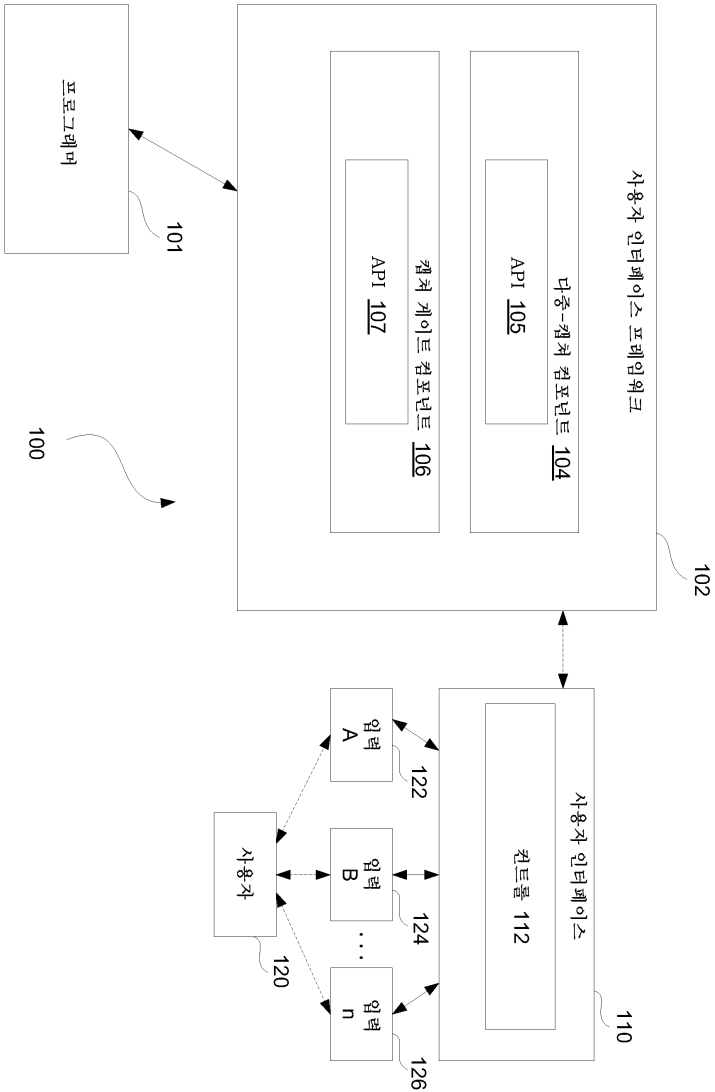
LAN 네트워킹 환경에서 사용될 때, 컴퓨터(710)는 네트워크 인터페이스 또는 어댑터(770)를 통해 LAN(771)과 연결된다. WAN 네트워킹 환경에서 사용될 때, 컴퓨터(710)는 통상적으로 모뎀(772) 또는 인터넷과 같은 WAN(773)상의 통신을 설정하기 위한 다른 수단을 포함한다. 내장형 또는 외장형이 될 수 있는 모뎀(772)은 사용자 입력 인터페이스(760)를 통해 시스템 버스(721)와 연결될 수 있다. 네트워크 환경에서, 컴퓨터(710)와 관련하여 도시된 프로그램 모듈 또는 이의 부분은 원격 메모리 저장 장치에 저장될 수 있다. 한정이 아닌 예시로서, 도 7은 원격 컴퓨터(780) 상에 상주하는 원격 애플리케이션 프로그램(785)을 도시한다. 도시된 네트워크 연결은 예시적이고 컴퓨터 사이에 통신을 설정하는 다른 수단이 사용될 수 있다는 것을 이해할 수 있다.

[0048]

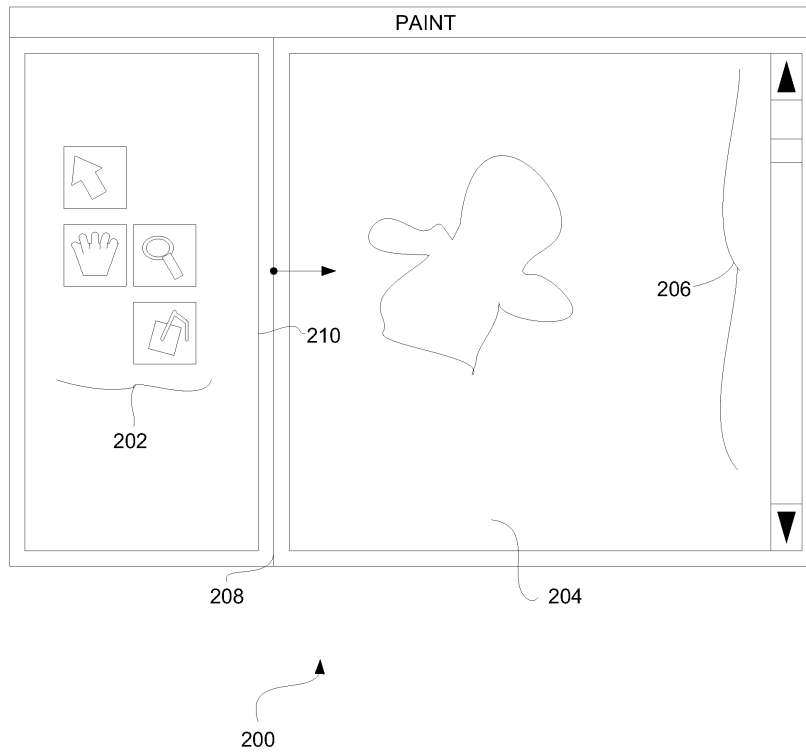
주요 발명이 특정한 구조적인 태양 및/또는 방법론적인 동작에 관한 언어로 개시되었지만, 첨부된 청구항에 정의된 주요 발명이 상기에 개시된 특정한 태양 또는 동작에 한정될 필요는 없다는 것을 이해할 수 있다. 더 나아가, 상기에 개시된 특정한 태양 및 동작은 청구항을 구현하는 예시적인 형식으로서 개시된다.

도면

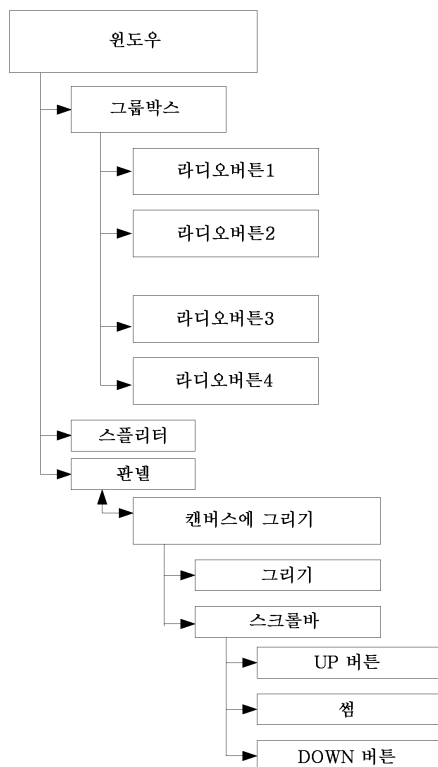
도면1



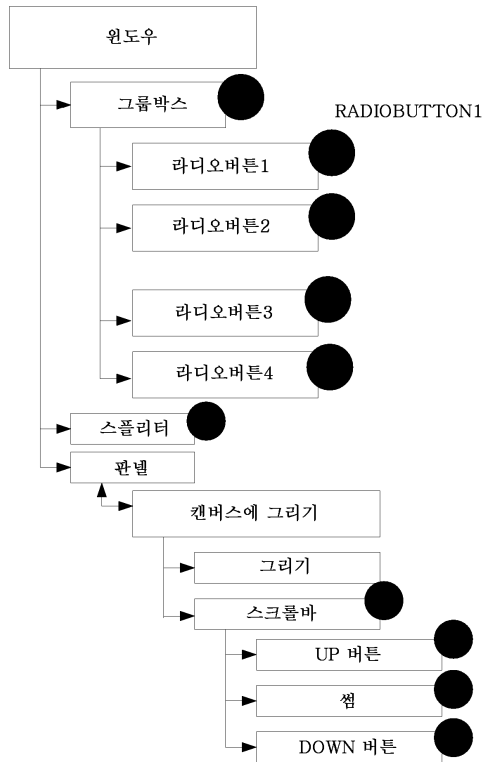
도면2



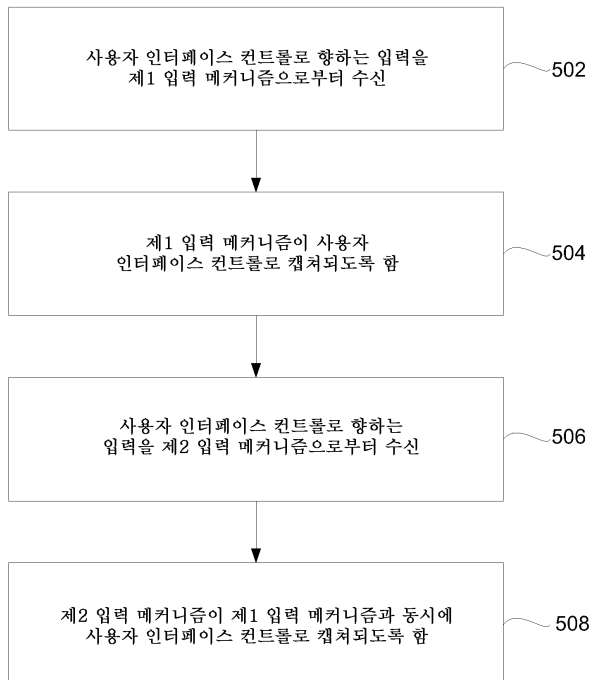
도면3



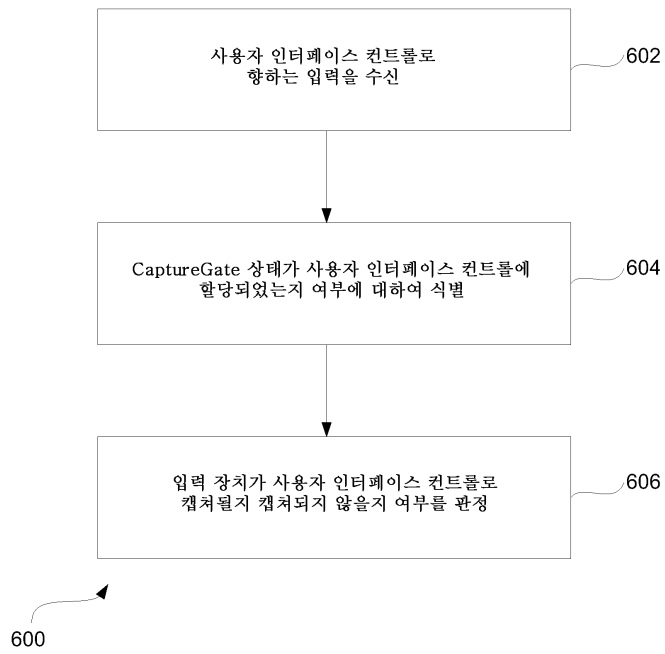
도면4



도면5



도면6



도면7

