



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2010년10월29일
(11) 등록번호 10-0990531
(24) 등록일자 2010년10월21일

(51) Int. Cl.

G06F 3/038 (2006.01) G06F 3/03 (2006.01)

(21) 출원번호 10-2009-0055855

(22) 출원일자 2009년06월23일

심사청구일자 2009년06월23일

(56) 선행기술조사문헌

KR1020070099093 A

KR100509505 B1

KR1020070048544 A

US20060107222 A1

(73) 특허권자

동서대학교산학협력단

부산광역시 사상구 주례2동 산69-1 동서대학교

(72) 발명자

윤태수

부산광역시 사상구 주례2동 반도보라메머드타운아파트112-803

이동훈

부산광역시 해운대구 우1동 경남마리나아파트 101-306호 15-1

(뒷면에 계속)

(74) 대리인

최한수

전체 청구항 수 : 총 2 항

심사관 : 천대식

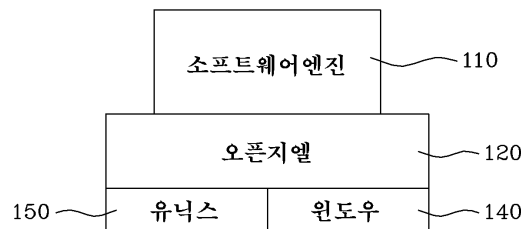
(54) 다중입력장치 호환성을 제공하는 클래스 구현 방법

(57) 요약

조이스틱, USB 캡 등을 포함하는 다중입력장치와 소프트웨어 엔진 간의 호환을 지원하는 클래스 구현 방법에 대해 개시한다.

상기 클래스 구현 방법은, 입력장치가 캡처 윈도우를 생성하고, 캡처 윈도우와 드라이버를 연결하고, 캡처 윈도우의 캡처속도를 설정하고, 디스플레이 모드를 설정하고, 입력장치로부터 획득한 영상을 버퍼에 저장하고, 버퍼의 주소를 클래스에 저장하고, 주소를 입력받아, 입력장치로부터 획득한 영상을 디스플레이하는 단계를 포함하는 것을 특징으로 한다.

대표도 - 도1



(72) 발명자

김광태

부산광역시 부산진구 전포1동 312-6번지 2통2반

박현우

부산광역시 해운대구 반여1동 장산3차롯데낙천대아
파트 301-2102

김기현

부산광역시 남구 용호2동 오륙도 sk뷰아파트
113-1601

특허청구의 범위

청구항 1

콘텐츠를 제작함에 있어서 입력장치와 소프트웨어 엔진과의 호환을 위한 클래스 구현 방법으로서,
 상기 입력장치가 캡처 윈도우를 생성하는 단계;
 상기 캡처 윈도우와 드라이버를 연결하는 단계;
 상기 캡처 윈도우의 캡처속도를 설정하는 단계;
 디스플레이 모드를 설정하는 단계;
 상기 입력장치로부터 획득한 영상을 버퍼에 저장하는 단계;
 상기 버퍼의 주소를 클래스에 저장하는 단계; 및
 상기 주소를 입력받아, 상기 입력장치로부터 획득한 영상을 디스플레이하는 단계를 포함하는 것을 특징으로 하는 클래스 구현 방법.

청구항 2

콘텐츠를 제작함에 있어서 입력장치와 소프트웨어 엔진과의 호환을 위한 클래스 구현 방법으로서,
 임의의 모션 타입을 레퍼런스하는 모션객체를 생성하는 단계;
 상기 입력장치로부터 데이터를 추출하기 위한 클래스를 생성하는 단계;
 상기 데이터를 추출하여 상기 클래스에 저장하는 단계; 및
 상기 저장된 데이터를 상기 모션 객체에 적용하기 위한 함수를 계산하는 단계를 포함하는 것을 특징으로 하는 클래스 구현 방법.

명세서

발명의 상세한 설명

기술 분야

[0001] 본 발명은 다중입력장치 호환성을 제공하는 클래스 구현 방법에 관한 것으로, 구체적으로는 조이스틱, USB 캠 등의 다중입력장치와 소프트웨어 엔진 간의 호환을 지원하는 클래스 구현 방법에 관한 것이다.

배경 기술

[0002] 컴퓨터 그래픽 기술과 하드웨어 장치의 발달로 컴퓨터를 이용해 가상의 환경을 실제와 같이 느끼고자 하는 가상 현실에 대한 연구가 다방면으로 진행되어 왔다. 과거에는 가상현실이 주로 군사 시뮬레이션 위주로 사용되었지만 현재는 시뮬레이션, 원격 통신, 교육, 산업, 국방, 쇼핑, 문화 등 거의 모든 분야에서 가상현실기법을 사용하고 있다. 그러나 컴퓨터 안에서 가상환경을 만들고 실제와 같은 다양한 호환이 가능하도록 하는 것은 어려운 작업과정을 거쳐야만 하므로 기술적인 개발에 많은 어려움이 있는 것이 현실이다.

[0003] 가상현실 시뮬레이션을 디스플레이 장치에 디스플레이하는 과정에 있어서 다양한 입력장치가 사용될 수 있다. 그러나, 종래 기술에 따르면 실감형 가상현실을 구현하기 위한 외부장치와의 접속이 용이하지 않으며, 이를 충족시키기 위해서는 고가의 부가적인 톨을 구비해야 한다는 문제점이 있고, 소프트웨어 엔진의 다양한 입력 장치에 대한 호환이 잘 이루어지지 않아, 별도의 프로그램을 설치하거나 다운로드하여야 하는 불편함이 있다.

[0004] 또한, 대부분의 소프트웨어 엔진은 일부 입력장치에 대해서만 제한적인 지원이 이루어지기 때문에 개발자가 원

하는 입력장치를 사용하지 못하는 불편함이 있다.

[0005]

발명의 내용

해결 하고자하는 과제

[0006]

본 발명은 전술한 종래의 문제점을 해결하기 위하여 안출된 것으로서, 실감형 가상현실을 구현하기 위해 별도의 프로그램을 설치하거나 고가의 톨을 부가적으로 구입할 필요없이, 소프트웨어 엔진의 다양한 입력 장치와 소프트웨어 엔진 간의 호환을 지원하는 클래스 구현 방법을 제공하고자 한다.

[0007]

또한, 일부 입력장치에 대해서만 제한적인 지원이 이루어지는 소프트웨어의 자유로운 사용에 있어서도 사용자 편의를 향상시키고자 한다.

과제 해결수단

[0008]

전술한 기술적 과제의 해결을 위하여 본 발명의 일 실시예에 따른 클래스 구현 방법은, 콘텐츠를 제작함에 있어서 입력장치와 소프트웨어 엔진과의 호환을 위한 클래스 구현 방법으로서, 상기 입력장치가 캡처 윈도우를 생성하는 단계; 상기 캡처 윈도우와 드라이버를 연결하는 단계; 상기 캡처 윈도우의 캡처속도를 설정하는 단계; 디스플레이 모드를 설정하는 단계; 상기 입력장치로부터 획득한 영상을 버퍼에 저장하는 단계; 상기 버퍼의 주소를 클래스에 저장하는 단계; 및 상기 주소를 입력받아, 상기 입력장치로부터 획득한 영상을 디스플레이하는 단계를 포함하는 것을 특징으로 한다.

[0009]

또한, 본 발명의 다른 실시예에 따른 클래스 구현 방법은, 콘텐츠를 제작함에 있어서 입력장치와 소프트웨어 엔진과의 호환을 위한 클래스 구현 방법으로서, 임의의 모션 타입을 레퍼런스하는 모션객체를 생성하는 단계; 상기 입력장치로부터 데이터를 추출하기 위한 클래스를 생성하는 단계; 상기 데이터를 추출하여 상기 클래스에 저장하는 단계; 및 상기 저장된 데이터를 상기 모션 객체에 적용하기 위한 함수를 계산하는 단계를 포함하는 것을 특징으로 한다.

효 과

[0010]

본 발명에 의하면 별도의 프로그램을 설치하거나 다운로드할 필요없이, 다양한 입력 장치에 대한 소프트웨어 엔진의 호환성을 확장시켜 가상현실과 관련된 시뮬레이션 등을 수행하고자 하는 사용자의 이용상의 편의를 도모할 수 있다.

[0011]

또한, 실감형 가상현실을 구현하기 위해 고가의 톨을 부가적으로 구입할 필요없이, 소프트웨어 엔진의 다양한 입력 장치에 대한 호환을 가능하게 하여 다중입력장치와 소프트웨어 엔진 간의 호환을 지원하는 클래스 구현 방법을 제공하고자 한다.

[0012]

또한, 일부 입력장치에 대해서만 제한적인 지원이 이루어지는 소프트웨어의 자유로운 사용에 있어서도 사용자 편의를 향상시키고자 한다.

발명의 실시를 위한 구체적인 내용

[0013]

이하 도면을 참조하여 본 발명에 관하여 살펴보기로 하며, 본 발명을 설명함에 있어서 관련된 공지기술 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명은 생략할 것이다.

[0014]

그리고 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있으므로 그 정의는 본 발명을 설명하는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다.

- [0015] 이하의 도 1 은 본 발명에 따른 소프트웨어 엔진의 시스템 구성도이고, 도 2 는 모션 클래스의 베이스 및 자(子) 클래스 다이어그램이고, 도 3 은 입력 클래스의 베이스 및 자(子) 클래스 다이어그램이고, 도 4 는 본 발명에 따라 구현한 클래스를 추가한 시스템 구성도이다.
- [0016] 컴퓨터 안에서 가상환경을 만들고 실제와 같은 다양한 인터랙션(interaction) 이 가능하도록 하는 것은 복잡하고 어려운 작업과정을 요하고, 실시간적인 요소들이 충분히 만족되어야 한다.
- [0017] 본 발명에서는 상기와 같은 요구를 충족시키기 위해 확장성이 뛰어난 실행환경을 제공할 수 있는 소프트웨어 엔진을 사용한다. 상기 소프트웨어 엔진은 간단한 환경이나 객체를 모델링할 수 있는 크리에이터 툴 (creator tool)을 제공한다. 상기 크리에이터 툴은 3D-Max 에서 모델링된 데이터를 로드해서 사용할 수 있는 기능을 제공하고, 실시간성을 위해 정보량을 최적화하는 기능 또한 제공할 수 있다.
- [0018] 다음으로, 상기 소프트웨어 엔진은 날씨나 기후등과 같은 환경적인 요인들을 간단한 세팅을 통해서 구현할 수 있도록 하는 프라임 툴 (prime tool) 을 제공한다. 이러한 프라임 툴의 사용으로 인해 가상환경을 쉽게 구성하고 다양한 네비게이션 방법들을 통해 보다 인터랙티브한 가상현실을 구현할 수 있다.
- [0019] 또한, 상기 소프트웨어 엔진은 API (application programming interface) 를 통해 확장성 있는 프로그램 개발을 지원할 수 있다. 상기 API 는 운영체제와 응용프로그램 사이의 통신에 사용되는 메시지 형식으로, 운영체제나 C, C++, Pascal 등과 같은 언어로 응용 프로그램을 만들 때, 윈도우를 만들고 파일을 여는 것과 같은 처리를 할 수 있도록 1000여 개 이상의 함수로 구성되어 있다. 한편, 상기 API는 프로그래머를 위한 운영체제나 프로그램의 인터페이스이므로 사용자와 직접 대하게 되는 그래픽 사용자 인터페이스나 명령형 인터페이스와는 구별된다.
- [0020] 한편, 상기 API는 응용 프로그램이 운영체제나 데이터베이스 관리 시스템과 같은 시스템 프로그램과 통신할 때 사용되는 언어나 메시지 형식을 가지며, 프로그램 내에서 실행을 위해 특정 서브루틴에 연결을 제공하는 함수를 호출하는 것으로 구현될 수 있다.
- [0021] 본 발명에서는 상기 소프트웨어 엔진을 사용하여 실감형 가상현실을 효과적, 경제적으로 구현하기 위해 외부입력장치와 상기 소프트웨어 엔진을 호환시키는 클래스를 구현한다.
- [0022] 예를 들어, 외부입력장치와 소프트웨어 엔진을 호환시키는 클래스로는 상기 소프트웨어 엔진에서 조이스틱을 제어하기 위해 DirectX 와 연동하여 데이터 값을 전달하는 조이스틱 클래스와 VFW(Video for windows) 를 사용하여 USB 캠의 영상을 버퍼에 저장한 후 소프트웨어 엔진의 디스플레이 버퍼에 그 값을 넘겨서 캠의 영상을 소프트웨어 엔진에서 처리할 수 있는 USB 캠 클래스가 있다.
- [0023] 여기서, VFW 는 동영상 압축 관리자로서, 마이크로소프트 윈도우가 디지털 비디오를 재생하게끔 하는 마이크로소프트가 개발한 멀티미디어 프레임워크이다.
- [0024] 도 1 은 본 발명에 따른 소프트웨어 엔진의 시스템 구성을 나타내는 블록도이다. 상기 소프트웨어 엔진 (110) 은 UNIX(130)와 Windows(140) 를 운영체제로 하여 오픈지엘 (OpenGL; Open Graphics Library, 120) 을 기반으로 하여 구성된다.
- [0025] OpenGL 은 그래픽 API 로, 프로그래머에게 단일한 API를 제공함으로써 서로 다른 3차원 가속기간의 복잡성을 감추고, 모든 구현이 완전한 OpenGL 기능 집합을 지원하도록 요구함으로써 하드웨어 플랫폼마다 다른 기능들을 감

추는 기능을 제공한다. 여기서, 필요하다면 소프트웨어 시뮬레이션을 통하여 하드웨어 플랫폼마다 다른 기능들을 감추어 모든 구현이 완전한 OpenGL 기능 집합을 지원하도록 할 수 있다.

- [0026] 상기 소프트웨어 엔진은 확장성이 뛰어난 실행환경을 제공하기 때문에 시스템을 최대한 활용하여 최대의 생산성을 구현할 수 있다.
- [0027] 본 발명의 일 실시형태에 따르면, 소프트웨어 엔진과 호환되는 다중 입력장치의 하나로서 조이스틱을 예시한다.
- [0028] 즉, 소프트웨어 엔진은 조이스틱을 통하여 객체와 인터랙션할 수 있으며, 이를 위해 조이스틱 클래스를 구현한다. 조이스틱 클래스 구현에는 모션 클래스와 입력 클래스를 사용한다.
- [0029] 모션 클래스는 프라임 톨에서 만들어진 모션을 소프트웨어 엔진 API 에서 조작하기 위해 모션 객체를 레퍼런스 하는 클래스이다. 이는 객체들의 모션을 정의하며, 입력장치 (예를 들어, 조이스틱) 각각의 값을 받아서 모션 객체에 적용시킨다.
- [0030] 도 2 는 예시적인 모션 클래스(200)의 베이스 및 자(子) 클래스 다이어그램이다. 모션 클래스의 자(子) 클래스로는 예를 들어, 모션드라이브(MotionDrive) 클래스(210), 모션플라이(MotionFly) 클래스(220), 모션스핀(MotionSpin) 클래스(230), 모션워크(MotionWalk) 클래스(240), 모션 와프(MotionWarp) 클래스(250), 모션게임(MotionGame) 클래스, 모션UFO 클래스 등이 있을 수 있다. 다만, 이들은 예시적인 것일 뿐이므로, 모션 클래스의 자 클래스가 이들로 한정되는 것은 아니다.
- [0031] 이 중 모션드라이브 클래스를 설명하면, 이는 자동차의 이동효과를 주기 위한 모션 타입 클래스이다. 모션드라이브 클래스의 멤버 변수에는 예를 들어 복수 개의 열거형 데이터가 있을 수 있는데, 이들 열거형 데이터는 속도와 관련된 변수, 핸들 움직임을 표현하는 변수등을 포함할 수 있다. 여기서, 속도와 관련된 변수는 예를 들어 속도의 증가, 감소, 정지의 경우를 지정할 수 있으며, 핸들 움직임을 표현하는 변수는 예를 들어 핸들 움직임 벡터의 X 축에 해당하는 헤딩값을 다루는 값을 정의할 수 있다.
- [0032] 다음으로, 도 3 은 예시적인 입력 클래스(300)의 베이스 및 자(子) 클래스 다이어그램이다. 상기 소프트웨어 엔진의 입력 클래스는 입력장치를 나타내는 기본적인 클래스로서, 입력장치로부터 입력받은 데이터를 저장하는 벡터를 포함한다.
- [0033] 입력 클래스의 자(子) 클래스로는 예를 들어, 입력컴포지트(InputComposite;310), 입력컨트롤러(InputController;320), 입력키보드(InputKeyboard;330), 입력마우스(InputMouse;340), 입력트래커(InputTracker;350), 입력조이스틱(InputJoystick; 360), 입력핀치글러브(InputPinchGloves; 380) 등이 있을 수 있다. 다만, 이들도 상기 모션 클래스와 마찬가지로 예시적인 것일 뿐이므로, 입력 클래스의 자 클래스가 이들로 한정되는 것은 아니다.
- [0034] 다음으로, 입력조이스틱 클래스의 구현과정에 대해 설명한다. 입력조이스틱 클래스는 입력클래스를 상속받아 구현할 수 있다.
- [0035] 즉, 입력조이스틱 클래스는 조이스틱을 입력장치로 사용하기 위해 입력 클래스로부터 유도된 임의의 클래스이다. 즉, 입력조이스틱 클래스는 조이스틱을 입력장치로 사용하기 위해 입력 클래스로부터 유도된 임의의 클래스이다. DirectX 를 사용하여 조이스틱 값을 입력조이스틱 클래스로 가져올 수 있다. DirectX 를 초기화하고, 조이스틱의 조작 데이터를 가져오기 위한 클래스로 입력조이스틱데이터 클래스를 만들고 입력조이스틱의 멤버 변수로 가진다.

- [0036] 예시적인 조이스틱 클래스의 구현 방법은 다음과 같다.
- [0037] 임의의 모션 타입을 레퍼런스하기 위해 API 에서 모션객체를 생성하고, 입력장치로부터 데이터를 추출하기 위한 입력조이스틱 클래스를 생성하고, DirectX 를 이용하여 조이스틱의 데이터를 저장한다.
- [0038] 그리고, 입력 조이스틱 클래스로 들어온 데이터를 모션 클래스에 삽입하기 위해 입력컴포지트에 넣은 뒤 모션 클래스의 멤버함수를 이용하여 모션벡터에 조이스틱의 데이터를 적용한다.
- [0039] 다음으로 VFW (Video for window) 를 사용하여 소프트웨어 엔진에서 지원하지 않는 외부입력장치인 USB 캐에 관한 클래스를 설명한다.
- [0040] 먼저 VFW 를 사용하여 먼저 캡을 셋업하여 영상을 획득한다. 영상을 획득하여 출력하는 방법은 다음과 같다. VFW 를 이용하여 캡처 윈도우를 생성하고, 캡처 윈도우와 드라이버를 연결한다. 그리고, 캡처속도, 즉, 초당 캡처되는 프레임 수를 설정하고, 화면에 디스플레이 모드를 설정한다.
- [0041] 이러한 과정을 거쳐 캡으로부터 획득한 영상을 버퍼에 저장하는데, 획득한 영상을 소프트웨어 엔진으로 가져와 화면에 디스플레이하기 위해서 버퍼의 주소를 저장한다. 즉, 추후에 캡을 통하여 획득한 이미지를 저장한 버퍼의 주소를 넘겨받음으로써, 캡의 영상을 소프트웨어 엔진에서 디스플레이 할 수 있다.
- [0042] 도 4 는 구현된 클래스를 추가한 본 발명의 전체 시스템 (400) 구성도이다. 이는 상기에서 설명한 본 발명의 전체 시스템을 요약하여 개시한다.
- [0043] 이상 본 발명의 설명을 위하여 도시된 실시예는 본 발명이 구체화되는 하나의 실시예에 불과하며, 도면에 도시된 바와 같이 본 발명의 요지가 실현되기 위하여 다양한 형태의 조합이 가능함을 알 수 있다.
- [0044] 따라서 본 발명은 상기한 실시예에 한정되지 않고, 이하의 특허청구범위에서 청구하는 바와 같이 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 분야에서 통상의 지식을 가진 자라면 누구든지 다양한 변경실시가 가능한 범위까지 본 발명의 기술적 정신이 있다고 할 것이다.

[0045]

도면의 간단한 설명

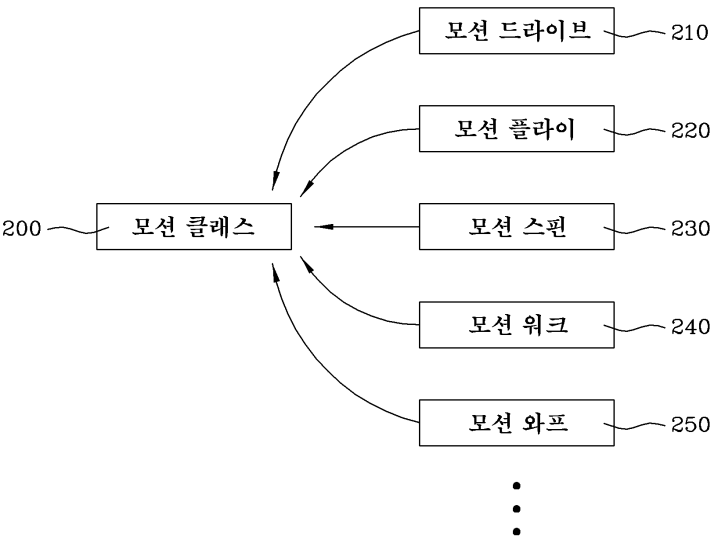
- [0046] 도 1 은 소프트웨어 엔진의 시스템 구성도.
- [0047] 도 2 는 모션 클래스의 베이스 및 자(子) 클래스 다이어그램.
- [0048] 도 3 은 입력 클래스의 베이스 및 자(子) 클래스 다이어그램.
- [0049] 도 4 는 본 발명에 따라 구현한 클래스를 추가한 시스템 구성도.
- [0050] <도면의 주요 참조부호에 대한 설명>
- [0051] 100 : 소프트웨어 엔진
- [0052] 200 : 모션 클래스
- [0053] 300 : 입력 클래스
- [0054]

도면

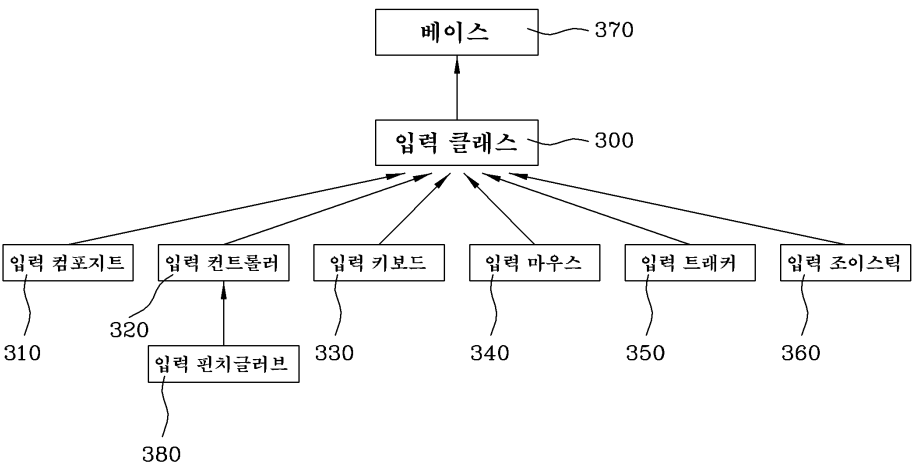
도면1



도면2



도면3



도면4

