

	(19) 대한민국특허청(KR) (12) 공개특허공보(A)	(11) 공개번호 10-2007-0099093 (43) 공개일자 2007년10월09일
(51) Int. Cl.	G06F 9/06 (2006.01) G06F 9/44 (2006.01) (21) 출원번호 10-2006-0030028 (22) 출원일자 2006년04월03일 심사청구일자 2006년04월03일	(71) 출원인 동서대학교산학협력단 부산광역시 사상구 주례2동 산69-1 동서대학교 (72) 발명자 김광태 부산 부산진구 전포1동 322-75번지 효정빌라 101호 박현우 부산 연제구 연산8동 384-5 정원그린빌 502호 (뒷면에 계속)

전체 청구항 수 : 총 2 항

(54) 인터랙티브콘텐츠 제작을 위한 다중입력장치 호환 래퍼클래스 개발방법

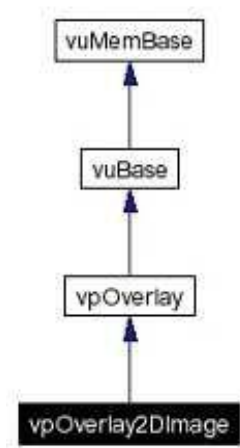
(57) 요약

본 발명은 가상현실콘텐츠와 같은 인터랙티브콘텐츠제작을 위한 소프트웨어엔진상에서 다양한 입력장치(조이스틱,USB카메라 등)호환을 위한 래퍼클래스 및 그 개발방법에 관한 것이다.

인터랙티브콘텐츠제작엔진은 일부 입력장치에 대해서만 제한적으로 지원하는 한계 때문에 개발자가 원하는 입력장치를 사용하지 못하는 경우가 있으며, 가격 또한 고가이기 때문에 특수한 입력장치를 사용하기 위해, 다른 엔진이나 별도의 옵션을 구매하기에는 많은 경제적인 부담이 든다.

본 발명에서는 이러한 문제를 해결하기 위하여, 개발자가 사용하고자 하는 입력장치와 인터랙티브콘텐츠엔진의 호환을 위한 래퍼클래스 및 그 개발방법을 제안함으로써 사용자가 손쉽게 저렴한 비용으로 콘텐츠제작을 도모하는 것이 본 발명의 목적이다.

대표도 - 도8



(72) 발명자

이동훈

부산 부산진구 개금3동 신개금엘지2단지아파트 20
6동 1705호

윤태수

부산 사상구 주례2동 반도보라메머드타운아파트
112동 803호

김형수

부산 부산진구 부암3동 458-76

특허청구의 범위

청구항 1

인터랙티브콘텐츠제작을 위한 소프트웨어엔진상에서 다양한 입력장치(조이스틱,USB카메라 등)호환을 위한 랩퍼 클래스 및 그 개발방법

청구항 2

제 1항에 있어서, DirectX를 이용하여 조이스틱의 값을 얻고, vpInputJoystick으로 값을 넘겨주기 이전에 페달의 값을 수정하는 알고리즘을 삽입하여 조이스틱 페달의 속도조절하는 방법

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

- <9> 컴퓨터 그래픽스 기술과 하드웨어 장치의 발달로 컴퓨터를 이용하여 가상의 환경을 실제와 같이 느끼고자 하는 가상현실에 대한 연구가 다방면으로 진행되어 왔다. 오늘날의 가상현실은 거의 모든 분야에서 사용되고 있다고 말해도 과언이 아닐 것이다. 과거에는 주로 군사 시뮬레이션 위주로 사용되었지만 현재는 시뮬레이션, 원격 통신, 교육, 산업, 국방, 쇼핑, 문화 등 거의 모든 분야에서 가상현실 기법을 응용하고 있다. 그러나 컴퓨터 안에서 가상환경을 만들고 실제와 같은 다양한 인터랙션이 가능하도록 하는 것은 매우 복잡하고 어려운 작업과정들을 거쳐야만 하고, 실시간 적인 요소들이 충분히 만족되어야 하기 때문에 기술적인 개발에 많은 어려움이 따른다.
- <10> 이러한 문제점들을 개선하기 위해 VR엔진들이 개발되어 상용화 되고 있다. VR엔진의 종류는 Tachyon, RTRE, Quake, Unreal, Littech, Eon, Virtool 등 다양한 기능을 하는 엔진들이 있다. 이러한 엔진들 중에는 군사용 시뮬레이터나 항만 시뮬레이터를 쉽고, 편리하게 개발 할 수 있는 Vega Prime이 있다.
- <11> Vega Prime은 3가지 요소로 구성되어 있다. 첫째로 간단한 환경이나 object를 모델링할 수 있는 Multigen creator 툴을 제공한다. Multigen creator는 Maya나 3D-Max에서 모델링된 데이터를 로드해서 사용할 수 있는 기능을 제공하고 있고, 실시간성을 위해서 정보량을 최적화하는 기능을 제공하고 있다. 둘째로 날씨나 기후등과 같은 환경적인 요인들을 간단한 세팅을 통해서 구현할 수 있도록 하는 LynX Prime 툴을 제공한다. 가상환경을 쉽게 구성하고 다양한 네비게이션 방법들을 통해 보다 인터랙티브한 가상현실을 구현할 수 있다. 마지막으로 Vega API를 통해 확장성있는 프로그램 개발을 지원한다. 그러나 Vega Prime 엔진의 사용은 환경제작이나 간단한 인터랙션 제작에는 용이하지만, 실감형 가상현실을 구현하기 위한 외부장치와의 연결이 용이하지 않다는 단점이 있다. 이를 충족시키기 위해서는 또 다른 고가의 툴을 구입해야만 한다.

발명이 이루고자 하는 기술적 과제

- <12> 상기와 같은 종래의 문제점을 해결하기 위한 본 발명의 목적은 실감형 가상현실을 효과적, 경제적으로 구현하기 위해 외부입력장치와 Vega Prime엔진을 호환시키는 랩퍼 클래스 및 개발방법을 제공함에 있다.

발명의 구성 및 작용

- <13> 상기한 목적을 달성하기 위한 본 발명의 구성은 크게 VR엔진에서 조이스틱 제어를 위한 조이스틱 랩퍼클래스 개발방법과 USB캠을 통하여 영상을 획득하기 위한 USB캠 랩퍼클래스 개발방법으로 나뉘어진다.
- <14> vpMotion 클래스는 LynX Prime에서 만들어진 모션을 Vega API에서 조작하기 위해 모션 객체를 레퍼런스하는 기본적인 클래스로써, object들의 모션을 정의하며, 입력장치 각각의 값을 받아서 모션 객체에 적용시킨다. 도 2는 vpMotion 클래스의 base 및 child 클래스 다이어그램이다. vpMotion 클래스에는 자동차의 이동효과를 주기 위한 모션 타입으로 vpMotionDrive 클래스가 있다. vpMotionDrive 클래스의 멤버변수에는 두 개의 열거형 데이터가 있는데, 첫 번째 멤버 변수는 enum SourceBoolean으로 속도의 증가, 감소, 정지의 세가지 경우에 대한 정의가 되어있다. 두 번째 멤버 변수는 enum SourceFloat으로 핸들의 x축에 해당하는 헤딩 값을 다루는 값이 정의

되어있다.

- <15> vpInput 클래스는 입력장치를 나타내는 기본적인 클래스로써, 입력 장치로부터 넘어오는 데이터를 저장하는 vector를 가진다. 도 3은 vpInput 클래스의 base 및 child클래스 다이어그램이다.
- <16> 도 7은 조이스틱 래퍼클래스 개발방법에 관한 구성도이다. 도 7의 vpInputJoystick 클래스는 개발한 조이스틱 클래스 이름으로써 vpInput 클래스를 상속받아 구현하였다.
- <17> vpInputJoystick 클래스는 조이스틱을 입력장치로 사용하기 위해 vpInput 클래스로부터 유도된 임의의 클래스이다. 그러나 Vega Prime의 클래스만으로는 조이스틱을 인식할 수가 없기 때문에 DirectX를 사용하여 조이스틱 값을 vpInputJoystick으로 가져와야 한다. DirectX를 초기화하고, 조이스틱의 조작 데이터를 가져오기 위한 클래스로 vpInputJoystickDXImpl을 만들고 vpInputJoystick의 멤버 변수로 가진다. 도 4는 vpInputJoystick 클래스의 동작과정을 나타낸다.
- <18> 도 3의 vpInputComposite은 vpInput으로부터 유도된 클래스이며, 입력장치로부터 가져온 데이터를 vpMotion과 연결을 위해vpMotion::setInput-(vpInput *Input)함수의 인자로 넘겨진다. vpInputJoystick의 값을 vpInputComposite에 넘겨줄 때는 도 5와 같이 vpMotion의 열거형 변수에 정의된 값의 순서대로 삽입해야 한다.
- <19> 페달이 없는 일반적인 조이스틱이라면 vpInput-Composite을 통해 vpMotion의 열거형 변수에 정의된 값의 순서로 삽입하면 그 값이 이상 없이 전달 될 것이나, 본 발명에서는 페달이 존재하는 조이스틱에 한하여, 페달의 유격에 따른 float형 Y축 좌표 값을 가져온다. vpMotion을 상속받은 vpMotionDrive는 자동차의 속도를 버튼의 눌러짐과 눌러지지 않음을 판단하는 Boolean Vector에서 처음 3개의 값을 가져와 속도를 제어하는 식으로 되어있다. 그러므로 기존의 설계되어 있는 vpMotionDrive을 그대로 사용하여서는 페달에 의한 속도조절이 불가능한 상태인 것이다.
- <20> 이러한 문제를 해결하기 위해서 본 발명에서는 DirectX를 이용하여 조이스틱의 값을 얻고, vpInputJoystick으로 값을 넘겨주기 이전에 페달의 값을 수정하는 알고리즘을 삽입, 속도조절에 관련된 Bool 값을 임의로 변경하여 vpInputComposite에 넘겨준다. 도 6은 문제해결 방법의 과정이다.
- <21> 도 7은 조이스틱 클래스 전체 도면이다. 상기 도를 참조하면 조이스틱 클래스의 구현 방법에 대해 설명하면, 첫째, LynX에서 만들어진 모션타입을 레퍼런스 하기위해 API에서 vpMotion객체를 생성한다. 둘째, 입력장치로부터 데이터를 가져오기 위한 vpInputJoystick을 생성하고 DirectX를 이용하여 조이스틱의 데이터를 저장한다. 셋째, vpInputJoystick으로 들어온 데이터를 vpMotion에 삽입하기 위해 vpInputComposite에 넣은 뒤 vpMotion의 멤버함수를 이용하여 vpMotion에 조이스틱의 데이터를 적용한다.
- <22> USB 캠을 VR엔진에 인터페이스시키는 방법을 설명하면 다음과 같다. 캠 클래스는 VFW를 사용하여 먼저 캠을 셋업하여 영상을 획득하여야 한다. 영상의 획득 및 출력하는 방법은 첫째, 캡처 윈도우 생성: VFW의 capCreateCaptureWindow()를 이용하여 생성한다. 둘째, 드라이버 연결 : VFW의 capDriverConnect()를 이용하여 캡처 윈도우와 드라이버를 연결한다. 셋째, 캡처속도 설정 : VFW의 capPreviewRate()를 이용하여 초당 캡처되는 프레임 수를 설정한다. 넷째, 프리뷰 : VFW의capPreview()를 이용하여 화면에 디스플레이 모드를 설정한다.
- <23> 이러한 과정을 거쳐 캠으로부터 획득한 영상을 버퍼에 저장하는데, 획득한 영상을 Vega Prime으로 가져와 화면에 디스플레이 하기 위해서 vpOverlay2DImage클래스의 addImageFile(const char* filename)에 버퍼의 주소를 넘겨준다. vpOverlay2DImage 클래스(도 8)는 vpOverlay 클래스의 자식클래스로써, 화면에 디스플레이 되고 있는 프레임의 영역에 이미지를 로드하는 기능을 한다 addImageFile()는 화면에 디스플레이 할 이미지파일의 주소를 인자로 받는데, 여기서는 캠을 통하여 획득한 이미지를 저장한 버퍼의 주소를 넘겨받음으로써, 캠의 영상을 Vega Prime에서 디스플레이 할 수 있다.
- <24> 도9는 래퍼클래스를 추가한 베가프라임의 엔진 구성도를 나타낸다.

발명의 효과

- <25> 본발명에서 개발한 래퍼 클래스인 조이스틱과 USB 캠 클래스는 개발에 사용된 조이스틱 및 USB 캠이 아니더라도 Vega Prime과 호환을 이루는데 문제가 없도록 입력장치의 제약사항들을 준수하여 표준에 맞게 개발되었다.
- <26> 개발한 래퍼 클래스를 이용하여 드라이빙, 영상인식 시뮬레이터를 개발할 때 Vega Prime등의 VR엔진과 호환을 이루는데 문제가 없으며, 성능 면에서도 효과적이다.

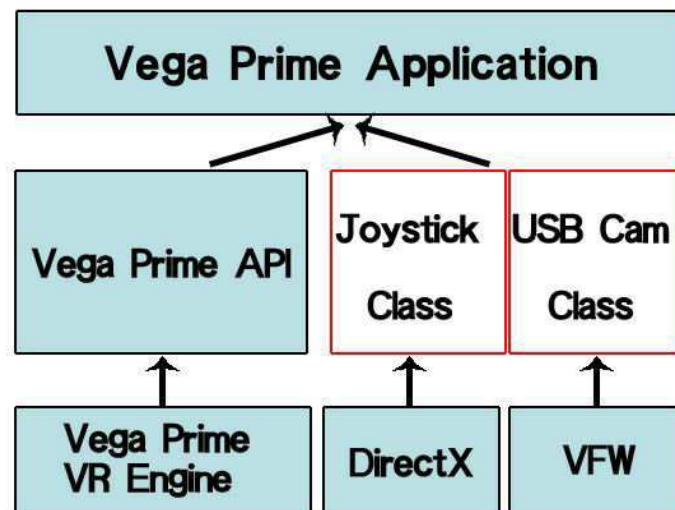
<27> 또한, 본 발명을 이용하여 조이스틱 클래스에 차량의 동역학 부분을 추가하여 더욱 실감적인 드라이빙 시뮬레이터 개발이 가능하다.

도면의 간단한 설명

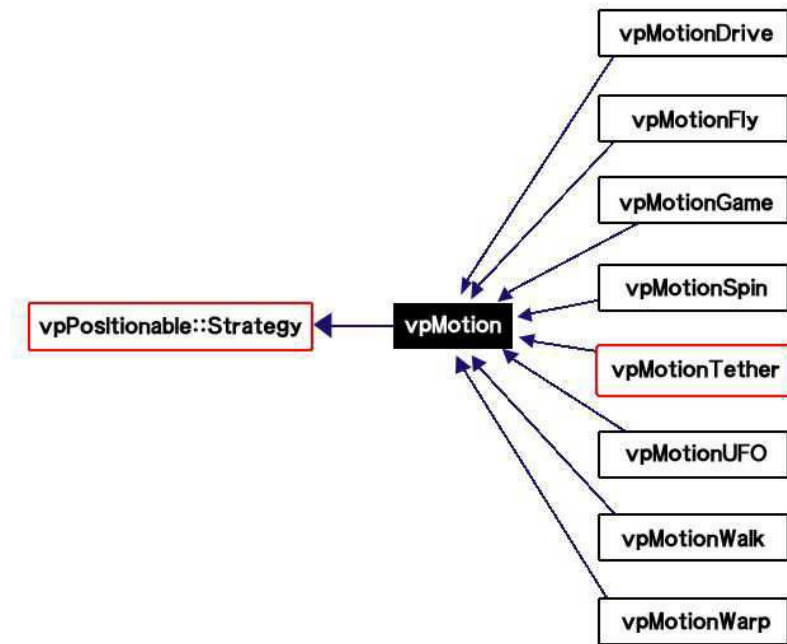
- <1> 도1 래퍼 클래스를 추가한 시스템 구성도
- <2> 도2 vpMotion클래스의 상속관계
- <3> 도3 vpInput클래스의 상속관계
- <4> 도4 vpInputJoystick 클래스의 동작과정
- <5> 도5 vpInputComposite에 값을 삽입하는 예
- <6> 도6 조이스틱 페달속도 조절방법
- <7> 도7 조이스틱 클래스 전체 다이어그램
- <8> 도8 vpOverlay2DImage 클래스의 상속관계

도면

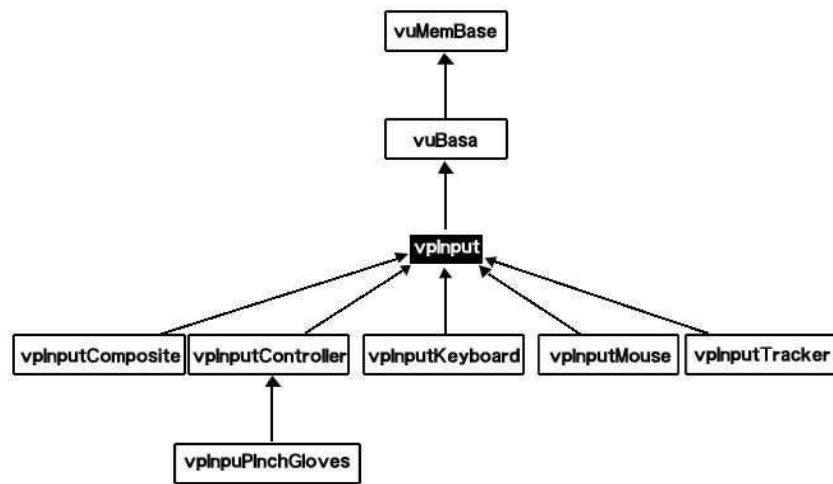
도면1



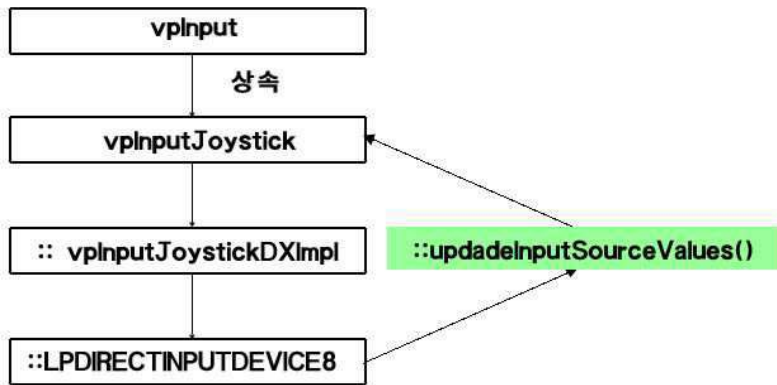
도면2



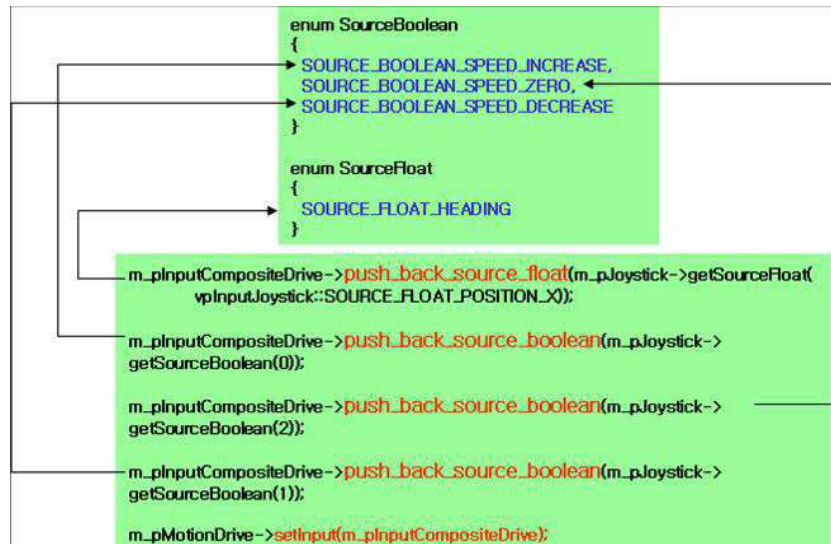
도면3



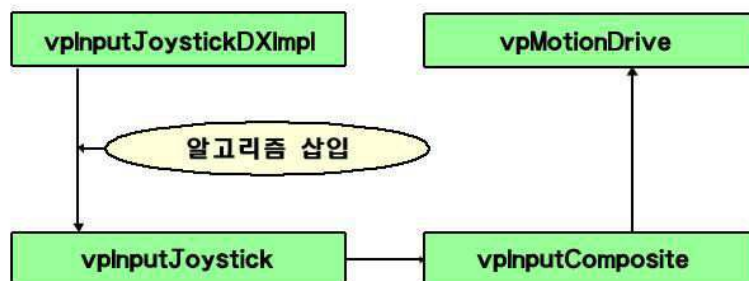
도면4



도면5



도면6



도면7



도면8

