



(19) 대한민국특허청(KR)  
 (12) 등록특허공보(B1)

(45) 공고일자 2012년03월27일  
 (11) 등록번호 10-1130487  
 (24) 등록일자 2012년03월19일

- (51) 국제특허분류(Int. Cl.)  
**G06T 3/40** (2006.01)
- (21) 출원번호 10-2005-0014608  
 (22) 출원일자 2005년02월22일  
 심사청구일자 2010년02월11일  
 (65) 공개번호 10-2006-0043083  
 (43) 공개일자 2006년05월15일  
 (30) 우선권주장  
 10/815,067 2004년03월31일 미국(US)  
 (56) 선행기술조사문헌  
 JP2002251624 A\*  
 WO2002097728 A2\*
- \*는 심사관에 의하여 인용된 문헌

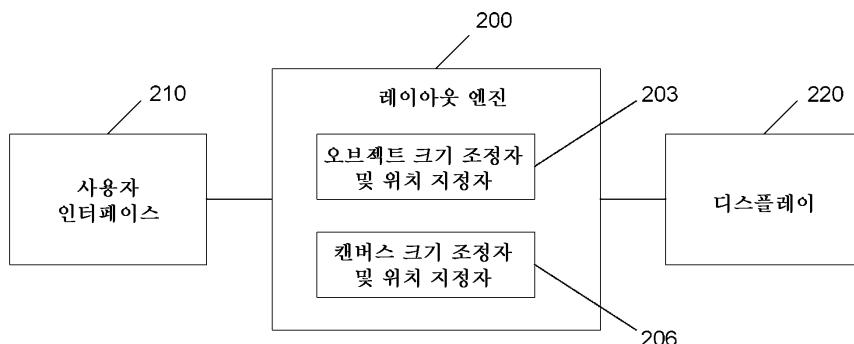
- (73) 특허권자  
**마이크로소프트 코포레이션**  
 미국 워싱턴주 (우편번호 : 98052) 레드몬드 원  
 마이크로소프트 웨이  
 (72) 발명자  
**티토프, 드미트리 지.**  
 미국 98052 워싱턴주 레드몬드 원 마이크로소프트  
 웨이 마이크로소프트 코포레이션 내  
**쿠페, 케니쓰 비.**  
 미국 98052 워싱턴주 레드몬드 원 마이크로소프트  
 웨이 마이크로소프트 코포레이션 내  
 (뒷면에 계속)  
 (74) 대리인  
**제일특허법인**

전체 청구항 수 : 총 19 항

심사관 : 박상철

(54) 발명의 명칭 **그리드 캔버스****(57) 요 약**

레이아웃은 복잡하지 않게 지능적으로 크기조정한다. 오브젝트 또는 엘리먼트들은 로우와 칼럼 그리드라인의 가상 그리드에 부착되고, 이것은 각각 정의된다. 오브젝트와 그리드 간의 관계는 양방향적이어서, 하나(그리드라인 또는 오브젝트)를 이동시키는 것은 다른 것(오브젝트 또는 그리드라인)에 영향을 줄 것이다. 다시 말하면, 오브젝트를 확대/축소하는 것은 그리드라인을 푸시할 것이고, 그리드라인을 이동시키는 것은 오브젝트를 확대/축소시킬 것이다. 더 많은 유연성을 제공하기 위해 가상 그리드는 오브젝트 전 또는 후에 생성될 수 있다. 자식 오브젝트는 자신의 가상 그리드를 가질 수 있는데, 이것은 요구되는 곳에 입자가 고운 제어를 허용한다.

**대 표 도** - 도3

(72) 발명자

영, 케니쓰 엘.

미국 98052 워싱턴주 레드몬드 원 마이크로소프트  
웨이 마이크로소프트 코포레이션 내

오베트흐킨, 올레그 브이.

미국 98052 워싱턴주 레드몬드 원 마이크로소프트  
웨이 마이크로소프트 코포레이션 내

페러데이, 피터

미국 98052 워싱턴주 레드몬드 원 마이크로소프트  
웨이 마이크로소프트 코포레이션 내

포슬러, 제시카 엘.

미국 98052 워싱턴주 레드몬드 원 마이크로소프트  
웨이 마이크로소프트 코포레이션 내

---

리헨보우어, 다니엘 알.

미국 98052 워싱턴주 레드몬드 원 마이크로소프트  
웨이 마이크로소프트 코포레이션 내

유안, 지안

미국 98052 워싱턴주 레드몬드 원 마이크로소프트  
웨이 마이크로소프트 코포레이션 내

크리소페, 제프리 티.

미국 98052 워싱턴주 레드몬드 원 마이크로소프트  
웨이 마이크로소프트 코포레이션 내

## 특허청구의 범위

### 청구항 1

그리드 캔버스가 렌더링된 디스플레이 장치로서, 상기 그리드 캔버스는,

캔버스;

로우들(rows)과 칼럼들(columns)이 교차하여 상기 캔버스 상에서 복수의 셀들을 형성하도록 상기 캔버스 상에서 로우들과 칼럼들을 정의하는 그리드 라인들 - 상기 그리드 라인들 각각은 각각의 크기조정 정보를 가지고 개별적으로 구성되고, 상기 그리드 라인들은 상기 캔버스 상의 컴포넌트들임 - ;

상기 캔버스 상의 상기 복수의 셀들에 걸쳐있는 사용자 인터페이스 엘리먼트 - 상기 사용자 인터페이스 엘리먼트는 상기 캔버스 상의 컴포넌트임 - ; 및

상기 사용자 인터페이스 엘리먼트가 걸쳐있는 상기 복수의 셀들을 형성하는 적어도 4개의 그리드 라인들의 바운딩 박스

를 포함하고,

상기 사용자 인터페이스 엘리먼트 및 상기 바운딩 박스 간의 관계는 4개의 여백들(margins)에 의해 정의되고, 각각의 여백은 상기 바운딩 박스의 적어도 4개의 그리드 라인들 중 하나 및 상기 사용자 인터페이스 엘리먼트 간의 거리를 각각 정의하며, 상기 사용자 인터페이스 엘리먼트는 상기 사용자 인터페이스 엘리먼트 및 상기 바운딩 박스 간의 상기 관계에 기초하여 상기 바운딩 박스 내에 부착되고,

여백은 고정된 값, 자동 값, 퍼센트 값 또는 가중된 값 중 임의의 하나이며,

만일 상기 캔버스 상의 임의의 컴포넌트에 변경이 이루어지면, 상기 사용자 인터페이스 엘리먼트 및 상기 바운딩 박스 간의 상기 관계는 상기 4개의 여백들 각각의 값에 따라 유지되고,

상기 사용자 인터페이스 엘리먼트 및 상기 바운딩 박스 간의 상기 관계와 상기 변경을 모두 반영하기 위해 상기 컴포넌트들이 재정의될 수 있도록, 상기 사용자 인터페이스 엘리먼트 및 상기 바운딩 박스 간의 상기 관계는 양방향적이며, 그에 맞추어 상기 사용자 인터페이스 엘리먼트는 상기 바운딩 박스 내에 계속 부착되어 있는, 그리드 캔버스가 렌더링된 디스플레이 장치.

### 청구항 2

제1항에 있어서,

그리드 라인은,

로우;

칼럼; 또는

적어도 하나의 로우 및 적어도 하나의 칼럼

중 적어도 하나에 의해 정의되는, 그리드 캔버스가 렌더링된 디스플레이 장치.

### 청구항 3

제2항에 있어서,

상기 로우 또는 상기 칼럼은 각각 가상 로우(virtual row) 또는 가상 칼럼(virtual column)인, 그리드 캔버스가 렌더링된 디스플레이 장치.

### 청구항 4

제1항에 있어서,

상기 복수의 셀들의 동일한 부분 또는 적어도 일부분에 걸쳐있는 제2 사용자 인터페이스 엘리먼트를 포함하는 제2 그리드 라인 바운딩 박스를 더 포함하는, 그리드 캔버스가 렌더링된 디스플레이 장치.

**청구항 5**

제1항에 있어서,

상기 바운딩 박스는 상기 사용자 인터페이스 엘리먼트를 포함하는 복수의 로우들 및 칼럼들을 포함하는, 그리드 캔버스가 렌더링된 디스플레이 장치.

**청구항 6**

제1항에 있어서,

그리드 라인은 상기 캔버스의 경계를 정의하는, 그리드 캔버스가 렌더링된 디스플레이 장치.

**청구항 7**

제1항에 있어서,

상기 캔버스 상의 상기 사용자 인터페이스 엘리먼트에 대한 상기 그리드 라인의 관계는 명시적 값(explicit value)으로서 정의되는, 그리드 캔버스가 렌더링된 디스플레이 장치.

**청구항 8**

제1항에 있어서,

상기 캔버스 상의 상기 사용자 인터페이스 엘리먼트에 대한 상기 그리드 라인의 관계는 자동 값(auto value)으로서 정의되는, 그리드 캔버스가 렌더링된 디스플레이 장치.

**청구항 9**

그리드 캔버스를 생성하는 방법으로서,

캔버스를 식별하는 단계;

상기 캔버스 상에 가상 그리드 라인을 정의하는 단계 - 상기 가상 그리드 라인은 상기 캔버스 상의 복수의 컴포넌트들 중 하나임 - ;

상기 캔버스 상의 복수의 셀들에 걸쳐있는 사용자 인터페이스 엘리먼트를 식별하는 단계 - 상기 사용자 인터페이스 엘리먼트는 상기 캔버스 상의 상기 복수의 컴포넌트들 중 하나이고, 상기 가상 그리드 라인이 정의되기 전 또는 상기 가상 그리드 라인이 정의된 후 중 적어도 하나의 경우에 상기 캔버스 상에 위치할 수 있음 - ;

상기 캔버스 상의 상기 사용자 인터페이스 엘리먼트에 대한 상기 가상 그리드 라인의 관계를 정의하는, 상기 가상 그리드 라인에 대한 속성 세트를 식별하는 단계;

상기 캔버스; 또는

상기 캔버스 상의 상기 복수의 컴포넌트들 중 적어도 하나

중 적어도 하나의 속성을 변경하는 단계;

상기 캔버스 상의 상기 사용자 인터페이스 엘리먼트의 레이아웃을 결정하는 단계 - 상기 사용자 인터페이스 엘리먼트의 상기 레이아웃은 상기 그리드 라인에 대한 상기 속성 세트에 의해 결정됨 - ;

상기 캔버스 상의 상기 사용자 인터페이스 엘리먼트에 대한 상기 가상 그리드 라인의 상기 관계를 유지하는 단계 - 상기 관계는 양방향적임 - ; 및

상기 사용자 인터페이스 엘리먼트가 크기조정되면 상기 그리드 라인이 이동하고, 상기 그리드 라인이 이동하면 상기 사용자 인터페이스 엘리먼트가 크기조정되는 단계

를 포함하는, 그리드 캔버스 생성 방법.

**청구항 10**

제9항에 있어서,

상기 캔버스 상의 상기 사용자 인터페이스 엘리먼트에 대한 상기 가상 그리드 라인의 관계를 식별하는 단계는

복수의 가상 그리드 라인들 및 복수의 사용자 인터페이스 엘리먼트들에 대하여 반복되는, 그리드 캔버스 생성 방법.

### 청구항 11

제9항에 있어서,

상기 캔버스에 동적으로 가상 그리드 라인을 추가하는 단계를 더 포함하는, 그리드 캔버스 생성 방법.

### 청구항 12

제9항에 있어서,

상기 캔버스 상에 복수의 가상 그리드 라인들을 포함하는 그리드를 오버레이하는(overlaying) 단계; 및

상기 캔버스 상의 상기 복수의 컴포넌트들 중 적어도 하나에 대한 상기 복수의 가상 그리드 라인들 중 적어도 하나의 관계를 식별하는 단계

를 더 포함하는, 그리드 캔버스 생성 방법.

### 청구항 13

제9항에 있어서,

그리드 상에 컴포넌트를 추가하는 단계를 더 포함하는, 그리드 캔버스 생성 방법.

### 청구항 14

제9항에 있어서,

상기 캔버스 상의 상기 복수의 컴포넌트들 중 적어도 하나에 대한 상기 가상 그리드 라인의 사전정의된 관계에 따라, 상기 캔버스 상에 상기 가상 그리드 라인을 위치시키는 단계를 더 포함하는, 그리드 캔버스 생성 방법.

### 청구항 15

제9항에 있어서,

상기 캔버스 상에 상기 가상 그리드 라인을 위치시키는 단계; 및

상기 캔버스 상의 상기 가상 그리드 라인의 상기 위치에 따라, 상기 캔버스 상의 상기 복수의 컴포넌트들 중 적어도 하나에 대한 상기 가상 그리드 라인의 관계를 식별하는 단계

를 더 포함하는, 그리드 캔버스 생성 방법.

### 청구항 16

제9항에 있어서,

상기 캔버스에 컴포넌트를 추가하는 단계;

상기 캔버스 상의 상기 엘리먼트에 대한 상기 가상 그리드 라인의 상기 관계를 유지하는 단계

를 더 포함하는, 그리드 캔버스 생성 방법.

### 청구항 17

제9항에 있어서,

상기 가상 그리드 라인은 복수의 가상 셀들을 정의하는 복수의 로우들 및 칼럼들에 의해 정의되고, 상기 복수의 컴포넌트들 중 적어도 하나는 상기 복수의 가상 셀들에 걸쳐있는, 그리드 캔버스 생성 방법.

### 청구항 18

제17항에 있어서,

상기 캔버스에 컴포넌트를 추가하는 단계 - 상기 추가된 컴포넌트는 상기 복수의 컴포넌트들 중 상기 적어도 하

나가 존재하는 상기 복수의 가상 셀들 중 동일한 셀들 가운데 적어도 하나에 존재함 - 를 더 포함하는, 그리드 캔버스 생성 방법.

### 청구항 19

제9항에 있어서,

상기 엘리먼트에 대한 가상 그리드 라인 바운딩 박스를 결정하는 단계를 더 포함하는, 그리드 캔버스 생성 방법.

### 청구항 20

삭제

### 청구항 21

삭제

### 청구항 22

삭제

### 청구항 23

삭제

### 청구항 24

삭제

### 청구항 25

삭제

### 청구항 26

삭제

### 청구항 27

삭제

### 청구항 28

삭제

## 명세서

### 발명의 상세한 설명

#### 발명의 목적

#### 발명이 속하는 기술 및 그 분야의 종래기술

[0017] 본 발명은 일반적으로, 컴퓨터 그래픽 분야에 관한 것이다. 특히, 본 발명은 디스플레이 상의 오브젝트의 레이아웃에 관한 것이다.

[0018] 사용자 인터페이스 엘리먼트들(윈도우 영역, 제어 등)의 레이아웃을 설계할 때, 종종 설계자 캔버스가 사용되고, 그 캔버스 상에 엘리먼트들(여기에서는 오브젝트라고도 지칭함)이 놓인다. 그러나, 어플리케이션이 구축되고 캔버스로 덮힌 영역이 크기조정(resize)됨에 따라, 지능적 방식으로 모든 엘리먼트들을 크기조정하는 것은 어려워지고 있다. 현재의 접근법은 설계자에게 불완전하거나 복잡하거나 불명료한 경향이 있다.

[0019] 레이아웃은, 설계자가 크기조정을 위한 코드를 기록하지 않고서도 크기조정되도록 설계하는 것이 바람직하다.

지능적으로 크기조정하는 어플리케이션 레이아웃을 생성하는 것은 불완전한 메카니즘, 과도하게 복잡한 메카니즘, 또는 설계자에 액세스불가능한 코드에 의해 현재 접근되는 문제이다.

[0020] 통상적인 메카니즘은 캔버스 스타일의 레이아웃이다. 이 유형의 레이아웃은 게시판에 게시물을 붙이는 것과 유사하게, 새로운 엘리먼트가 추가될 수 있는 영역을 제공한다. 게시판 상에서와 같이, 오브젝트는 한번 배치되면 프로그래머가 그 오브젝트를 명시적으로 이동시키지 않는 한 고정된다.

[0021] 다른 통상적인 메카니즘은 비례-크기조정 레이아웃이다. 이 레이아웃은, 예외적으로, 전체 게시판이 나중에 비례적으로 확대될 수 있는 단일 이미지로서 간주된다는 점을 제외하고 캔버스 스타일의 레이아웃과 동일하다. 이 유형의 레이아웃은 스크린 크기가 증가함에 따라 더 커지게 되어 있는 고정-기능 사용자 인터페이스에 대해서는 적합하지만(예를 들어, 단순한 게임), 스크린이 보다 더 높은 밀도의 정보를 표시하는 데 적용되어야 하는 경우에 대해서는 취약하다(예를 들어, 워드 프로세싱 디스플레이).

[0022] 또 다른 통상적인 레이아웃 메카니즘은 테이블-유형의 레이아웃이다. 이 레이아웃은 HTML의 레이아웃과 유사하다. 사용 가능한 공간은 별개의 셀들로 분할되는데, 이 각각의 셀은 컨텐츠에 대한 컨테이너이다. 종종, 보다 큰 엘리먼트들(예를 들어, 이미지)은 요구된 레이아웃을 생성하기 위하여 셀들 사이에서 분할되어야 한다.

[0023] 또 다른 유형의 레이아웃 메카니즘은 부착-기반 레이아웃이다. 이 레이아웃에서, 각각의 오브젝트는 자신을 전체 캔버스의 측면에 "부착"하도록 허용된다. 예를 들어, 좌측 및 우측에 부착하면, 어플리케이션이 크기조정되는 반면 여백이 고정되어 있기 때문에, 오브젝트의 폭은 변한다.

[0024] 자바의 GridBagConstraints과 같은 "고급" 레이아웃도 공지되어 있다. GridBagConstraints은 (스페닝 정보 및 여백을 포함하여) 위치 지정 정보를 각각의 자식에게 지정하고, 모든 자식 데이터를 추론하여 결과를 생성한다. GridBagConstraints 컨테이너 자체는 정보를 수반하지 않는다. 결과는 전술된 레이아웃들보다 더 기능적이지만, 도구화하기에는 매우 어렵다(즉, 간단명료한 사용자 인터페이스로 표시하기 어려움).

[0025] 각각의 오브젝트 좌표가 임의의 다른 좌표로부터의 표현으로서 효과적으로 정의될 수 있는 통상적인 레이아웃은 "스프링 및 받침대(Spring and struts)"이다. 예를 들어, "스프링 및 받침대" 환경에서는, 항상 10픽셀 떨어져 있는 2개의 엘리먼트를 형성할 수 있다. 초기 표현동안, 이 메카니즘은 특히, 실행시간이 최종적으로 테스크에 부적절한 재계산 요구와 매우 관련되었기 때문에, 빨리 보그 다운(bog down)된다.

### 발명이 이루고자 하는 기술적 과제

[0026] 상기 관점에서, 종래 기술의 한계와 결점을 극복하는 시스템 및 방법이 요구된다.

[0027] 본 발명은 복잡하지 않게 지능적으로 크기조정하는 레이아웃에 관한 것이다. 엘리먼트들(즉, 오브젝트들)은 로우(row) 및 칼럼(column) 그리드라인의 가상 그리드에 부착되는데, 이것은 개별적으로 정의된다. 오브젝트들과 그리드 간의 관계는 양방향적이어서 하나(그리드라인 또는 오브젝트)를 이동시키는 것은 다른 것(오브젝트 또는 그리드라인)에 영향을 줄 것이다. 다시 말하면, 오브젝트를 확대/축소하는 것은 그리드라인을 푸시할 것이고, 그리드라인을 이동시키는 것은 오브젝트를 확대/축소할 것이다. 가상 그리드는 더 많은 유연성을 제공하기 위해 오브젝트 이전 또는 이후에 생성될 수 있다. 자식 오브젝트는 자신의 가상 그리드를 가질 수 있는데, 이것은 요구되는 곳에 입자가 고운 제어를 허용한다.

[0028] 첨부 도면을 참조하는 다음의 예시적인 실시예의 상세한 설명으로부터 본 발명의 추가적인 특징 및 장점이 명백해질 것이다.

[0029] 상술한 요지 뿐만 아니라, 다음의 선호되는 실시예의 상세한 설명은 첨부 도면과 함께 잘 이해될 것이다. 본 발명을 설명하기 위하여, 본 발명의 예시적인 구성이 도면에 도시되지만, 본 발명은 개시된 특정 방법 및 수단으로 한정되지 않는다.

### 발명의 구성 및 작용

[0030] 본 발명은 오브젝트들이 로우 및 칼럼 그리드라인의 가상 그리드에 부착되고, 이것이 개별적으로 정의되는 레이아웃에 관한 것이다. 오브젝트들과 그리드 간의 관계는 양방향적이어서, 오브젝트를 확대/축소하는 것은 그리드라인을 푸시할 것이고, 그리드라인을 이동시키는 것은 오브젝트를 확대/축소할 것이다. 그러한 레이아웃 메카니즘은 여기에서 "그리드 캔버스" 레이아웃 메카니즘이라고 지칭한다.

[0031] 도 1에서는 예시적인 그리드 캔버스 레이아웃이 도시된다. 그리드 캔버스 레이아웃 메카니즘은 부모와 자식들

간의 협동이다. 부모(10)는 오브젝트들(30 및 40)이 임의의 장소에 마음대로 그려질 수 있고 위치된 곳에 남아 있는 효과적인 캔버스이다. 아무때나, 그리드 캔버스(10)는 요구되는 만큼의 많은 로우(R)와 칼럼(C)으로 분할될 수 있다. 그리드 캔버스의 각각의 자식은 박스에 대한 여백과 함께 그리드 바운딩 박스를 유지한다. 예를 들어, 오브젝트(30)에 대한 그리드 바운딩 박스는 로우(R1) 모두(예를 들어, R1C1:R1C2)이다. 여백 설정은 박스의 경계로부터의 적절한 오프셋을 사용하여, 그리드 바운딩 박스 내의 다수의 유형의 "부착"을 허용한다. 로우(R) 및 칼럼(C) 자체는 그리드 캔버스 부모(10)에 의해 소유되고, 통상적인 방식(예를 들어, 고정된 크기, 퍼센트 크기, 자동-크기, 가중된 크기)으로 크기지정(sizing)될 수 있다.

[0032] 본 발명은 그림을 위치만을 나타내는 가상 셀들로 나눌 수 있다. 따라서, 예를 들어, 오브젝트(30)는 각각이 스스로 이동될 필요가 있는 다수의 셀들로 쪼개질 필요가 없다. 대신에, 오브젝트(30)는 기저 그리드 캔버스(10) 상의 몇몇 가상 셀들에 걸쳐있다. 또한, 오브젝트는 중첩될 수 있다. 그리드라인은 가상이며 좌표계로서 작용하고, 가상의 셀들은 좌표계의 명시일 뿐이다. 본 발명은 그리드라인에 대해 위치되어야 하는 자식 셀들의 종속성을 깨뜨린다.

[0033] 본 발명은 자식들을 로우와 칼럼에서 계산하고 배열시킬 수 있는 사용자 인터페이스 패널에서 구현될 수 있다. 자식은 패널 또는 제어와 같은 임의의 오브젝트일 수 있다. 칼럼과 로우 그리드라인의 교차는 슬롯들의 그리드를 형성한다. 자식 또는 오브젝트는 몇몇의 인접한 그리드 슬롯들을 차지할 수 있다(예를 들어, 오브젝트(30)는 로우(R1), 칼럼(C1) 및 로우(R1), 칼럼(C2)에 의해 정의된 슬롯 또는 셀들을 차지함). 오브젝트의 레이아웃 행위는 그것이 걸쳐있는 칼럼과 로우 상에 설정함으로써 결정된다. 예를 들어, 그리드 슬롯은 대응하는 칼럼의 픽셀 폭 값, 및 대응하는 로우의 픽셀 높이 값을 지정함으로써, 고정된 크기를 지정받을 수 있다. 칼럼 폭은 그것을 통해 걸쳐있는 자식들의 요구된 폭에 대해 계산될 수 있다. 로우는 남아있는 사용가능한 공간에 대한 비율을 지정받을 수 있다. 2개 이상의 자식들은 레이아웃 슬롯들의 동일한 서브집합을 차지할 수 있기 때문에, 오브젝트들은 중첩할 수 있다(예를 들어, 도 5의 오브젝트들(20 및 30)을 참조).

[0034] 자식 오브젝트는 4개의 그리드 좌표(그리드 셀의 바운딩 박스), 경계 박스의 4개의 여백, 및 오브젝트의 높이와 폭과 같은 속성을 가질 수 있지만, 이것으로 제한되는 것은 아니다. 여백 및 높이와 폭은 "자동"으로 설계되어, 예를 들어, 그것이 적절한 비율과 위치를 유지하는 데 사용가능한 한 많은 공간을 차지하도록 할 수 있다. 특히, 폭과 높이에 있어서는, 바람직하게, 절대(예를 들어, 100픽셀, 2인치 등), 퍼센트(예를 들어 10퍼센트), 자동(즉, 자연 크기) 및 스타("\*")(예를 들어, \*, 25\* 등)와 같은 여러가지 유형의 명세가 있다. 스타는 남아있는 공간을 사용하기 위한 명세이다. 2개의 칼럼이 "\*"로서 표시되면, 그 2개의 오브젝트에 대한 공간은 그들의 스타 값에 비례하여 나뉘어진다. 바람직하게, 이것은 가중 기반 분배이다.

[0035] 도 2는 그리드 캔버스를 생성하는 예시적인 방법의 고레벨 흐름도이다. 단계(100)에서, 로우 및 칼럼 그리드라인의 가상 그리드를 지정함으로써 기저 그리드 또는 부모가 생성된다. 단계(110)에서, 그 오브젝트 또는 오브젝트들은 오브젝트가 가질 수 있는 여러가지 속성 및 특성을 사용하여 정의된다. 로우와 칼럼들도 이때 정의될 수 있다. 그 정의가 완료될 때까지 루프가 제공된다. 그후, 단계(120)에서, 그리드를 측정하고 오브젝트를 위치시킨다. 오브젝트가 그리드 내에 위치될 수 있기 전에 그리드가 선언되지만, 그리드는 오브젝트가 그 안에 위치되기 전 또는 후의 임의의 시간에 구성될 수 있음을 숙지한다. 예를 들어, 오브젝트가 먼저 페이지 상에 레이아웃될 수 있고, 그 후에 그리드가 구성될 수 있다.

[0036] 레이아웃 컨테이너는 바람직하게, 단일-로우, 단일-칼럼 그리드로서 개시한다. 그리드는 고정된 수의 로우와 칼럼을 갖지만, 셀은 가지고 있지 않다. 로우와 칼럼은 고정되거나 연장될 수 있다. 오브젝트가 그려지면, 바람직하게 2개의 유형의 특성, 즉, 그리드라인의 바운딩 박스를 지정하는 로우, 로우스팬, 콜(col), 콜스팬과 좌측, 우측, 상단, 하단, 폭, 높이를 부여받는다. 오브젝트는 그리드라인 바운딩 박스 내에 위치된다.

[0037] 따라서, 예시적인 그리드 캔버스는 풍부한 대화와 컴포넌트 레이아웃의 생성을 제공한다. 특정 집합에는 바람직하게, 칼럼과 로우에의 레이아웃; 중첩; 절대, 퍼센트, "투-컨텐츠(to-content)" 및 칼럼과 로우를 크기지정하는 가중된 속성에 대한 지원; 자식의 위치에 대한 전체 제어; 및 크기지정 행위가 있다.

[0038] 바람직하게, 오브젝트들은 그리드라인에 대해 로우와 칼럼으로 배열된다. 로우와 칼럼은 잡금될 수 있고/있거나 예를 들어, 최소 크기를 부여받을 수 있다. 오브젝트들은 그것에 걸쳐있는 로우와 칼럼의 설정에 따라 연장하도록 그리드 내에 위치될 수 있다. 다수의 오브젝트들은 각각 상이한 영역의 셀들을 사용하여 중첩할 수 있다. 그러한 예는 도 5에 되시되어 있고, 이하에서 설명할 것이다. 그리드가 데이터바인드 가능하고 그 로우와 칼럼의 수가 알려지지 않는것이 바람직하다.

- [0039] 본 발명에 따른 시스템은 최적 알고리즘을 구현하는 레이아웃 엔진, 오브젝트 크기와 위치를 결정하기 위한 프로세스, 및 그리드 자신의 칼럼들과 로우들의 크기 및 위치를 포함한다. 도 3에서 도시되는 바와 같이, 고레벨에서, 예시적인 레이아웃 엔진(200)은 컴퓨터 내에, 또는 마이크로프로세서나 컨트롤러와 함께 상주하고, 사용자 인터페이스(210) 및 디스플레이(220)와 통신한다. 레이아웃 엔진은, 오브젝트를 크기조정 및 위치지정하기 위한 루틴과(203), 캔버스를 크기조정 및 위치지정하기 위한 루틴(206)을 포함한다.
- [0040] 바람직하게, 레이아웃 결정은 2가지 모드로 {양 치수, 즉, 가로(폭)와 세로(높이) 양자에 있어서 독립적으로} 이루어질 수 있다. 제1 예시적인 모드는 사전-정의된 크기(예를 들면, 크기가 레이아웃 제작자에 의해 명백하게 설정될 때)에 대한 계산이다. 이 경우에, 크기는 오브젝트의 자연 크기를 고려하여 칼럼(로우)들 사이에서 분배된다. 제2 예시적인 모드는 자연 크기(예를 들면, 명백한 크기가 정의되지 않을 때)에 대한 계산이다. 이러한 경우에, 오브젝트의 자연 크기와 칼럼(로우)들에 대한 크기 명세를 고려하여 그리드의 자연 크기가 결정되고, 퍼센트 크기 명세와 가중된 속성 크기 명세의 경우에 칼럼(로우) 크기 속성을 유지한다.
- [0041] **실행시간(runtime)**
- [0042] 그리드 캔버스는 실행시간 시스템에서 사용될 수 있는 엘리먼트이다. 그리드 캔버스(10)는 도 4에 도시되는 바와 같이, 로우(R)와 칼럼(C)으로 나뉘어진다. 두개의 로우(R1 및 R2)와 2개의 칼럼(C1 및 C2)이 도4에 도시되어 있지만, 그리드 캔버스는 임의의 수의 로우와 칼럼들로 나뉘어질 수 있음을 숙지한다. 각각의 로우와 칼럼은 바람직하게 그에 대한 크기조정 정보(예를 들어, 고정, 퍼센트, 자동 또는 가중)를 갖는다.
- [0043] 자식 오브젝트들은 부모 그리드 캔버스 상의 어느곳에나 위치할 수 있다. 예시적인 자식이 4개의 오브젝트(20, 30, 40 및 50)로서 도 5에 도시되어 있다. 오브젝트(20 및 30)는 다수의 셀(로우와 칼럼이 셀을 형성하는 것으로 함)에 걸쳐 있다. 오브젝트(20)는 로우(R1 및 R2)와 칼럼(C1 및 C2)에 상주하는 한편, 오브젝트(30)는 로우(R1)와 칼럼(C1 및 C2)에 배치된다. 오브젝트(40)는 완전히 로우(R2)와 칼럼(C1)에 의해 형성된 셀 내에 있고, 오브젝트(50)는 완전히 로우(R2)와 칼럼(C2)에 의해 형성된 셀 내에 있다.
- [0044] 이 시점에서, 도 5의 레이아웃의 하단 로우(R2)는 절대 높이를 갖도록 요구되는 반면, 상단 로우(R1)는 모든 남아있는 공간을 소비하도록 요구된다. 이 레이아웃이 보다 높게 크기조정될 때, 그 요구된 결과는 도 6에 도시된다. 로우(R2)와, 그 안에 포함된 오브젝트 또는 오브젝트의 일부는 본래 고정된 높이로 남아있음을 유념한다. 한편, 로우(R1)와, 그 안에 포함된 오브젝트 또는 오브젝트의 일부는 수직으로 크기조정되어, 그 방향의 추가적인 공간에 걸치게된다. 로우(R1)에서 오브젝트와 셀의 서로에 대한 상대적 비율은 본래의 크기조정되지 않은 로우(R1)와 동일하게 남아 있다.
- [0045] 그러한 레이아웃은 로우/칼럼을 적절히 구성함으로써(예를 들어, 하단 로우 높이="150픽셀"; 상단 로우 높이="\*", 이때 "\*"는 고정된 로우/칼럼이 차지한 후의 사용 가능한 공간의 비율을 취하는 것을 의미함; 다른 로우가 "\*"값을 갖지 않는 경우, "\*"는 전체 남아있는 공간을 사용할 것임) 달성된다. C1, R1이 (0,0); C2, R1이 (1,0); C1, R2가(0,1); C2, R2가 (1,1)로 셀이 정의된 좌표계가 사용된다. 오브젝트의 가장자리와 셀의 대응하는 가장자리 간의 거리는 "여백"이고, 여백은 픽셀로 또는 자동으로 정의된다고 가정하면, 오브젝트 관계는 다음과 같이 오브젝트(20, 30, 40, 50)에 대해 정의될 수 있다.
- [0046] 오브젝트(20): 그리드 위치=(0,0)-(1,1){도시되는 바와 같이, 이것은 오브젝트가 셀(0,0)을 셀(1,1)에 거쳐있는 것을 의미함}; 여백=(10,10,10,10){이것은 각 방향의 가장자리로부터 그에 대응하는 그리드 셀 가장자리까지의 거리가 10픽셀임을 의미함, 도면은 크기조정하지 않을 것임을 유념한다}; 폭=자동; 높이=자동.
- [0047] 오브젝트(30): 그리드 위치=(0,0)-(1,0); 여백=(20,20,20,10); 폭=자동; 높이=자동.
- [0048] 오브젝트(40): 그리드 위치=(0,1)-(0,1); 여백=(20,0,자동,20); 폭=고정{예를 들어, 100}; 높이=자동.
- [0049] 오브젝트(50): 그리드 위치=(1,1)-(1,1); 여백=(10,10,10,자동); 폭=자동; 높이=고정{예를 들어, 40}.
- [0050] 바람직하게, 명백한 값은 명백한 측정을 나타내고, 자동 값은 남아있는 사용 가능한 공간을 소비한다. 여러 가지 구현 예에서, 여백은 모든 측면에 대해 지정될 수 있고, 추가적인 부착 플래그가 추가될 수 있음을 주지한다. 또한, 자식에 의해 요구된 값은 부모에 의해 성취될 수 없고, 그러한 경우에, 그리드 캔버스는 에러 메시지를 리턴함으로써, 예를 들어, 값이 유효하도록 값을 정정함으로써, 어쩔 수 없는 문제를 처리할 수 있다.
- [0051] 실행시간은 자식 오브젝트를 측정하고, 실행시간이 동작하는 임의의 제약에 따라 로우 폭과 칼럼 높이를 지정하고, 얼마나 많은 공간이 사용 가능한지에 기초하여 나중에 자식 오브젝트를 위치시키는 기능을 한다.

## [0052] 설계 시간

설계 시간 프로세스는 최소량의 사용자 명세를 사용하여 그리드 캔버스 레이아웃을 자동적으로 생성하는 동작이다. 레이아웃 제작자 또는 설계자는 레이아웃 모양 및 레이아웃 동적 행위를 정의하는 능력을 제공받는다.

도 7은 본 발명에 따라 그리드 캔버스 레이아웃을 생성하는 예시적인 방법을 도시한다. 단계(700)에서, 설계자는 오브젝트를 디스플레이 상의 특정 위치에 (예를 들어, 디스플레이 상의 사용자 인터페이스에) 위치시킨다. 단계(710)에서, 그것의 그리드라인 바운딩 박스가 결정된다. 이것은 상술된 그리드 위치 속성에 맵핑된다. 그 후, 단계(720)에서, 이 그리드라인 바운딩 박스의 범위를 요구된 위치와 비교함으로써, 적절한 여백 및 폭/높이가 예를 들어, 디폴트 스킴에 따라, 결정될 수 있다. 예시적인 디폴트 스킴은 (1)지정된 위치가 완전히 그리드 박스 범위의 좌측 반쪽 내에 있는 경우, 명백한 좌측 여백 및 폭이 설정되고, 우측 여백이 자동으로 설정되고; (2)지정된 위치가 완전히 그리드 박스 범위의 우측 반쪽 내에 있는 경우, 명백한 우측 여백과 폭이 설정되고, 좌측 여백이 자동으로 설정되고; (3)지정된 위치가 그리드라인 박스 범위의 중앙라인에 걸쳐있는 경우, 명백한 좌측 및 우측 여백이 설정되고, 폭이 자동으로 설정되고; (4)높이 및 상단/하단 여백이 아날로그 방식으로 처리되는 것; 중 하나가 된다.

도 8은 특히, 로우(R2)와 칼럼(C2)의 오브젝트(50)에 대하여, 상기 설정들을 설명하는 데 유용한 예시적인 그리드 캔버스를 도시한다. 고정된 값은 실선으로 도시되고, 자동 값은 점선으로 도시된다. 도시된 6개의 위치지정 지시자들 중 임의의 것(높이, 폭 및 여백)을 토클링(toggle)함으로써, 사용자는 위치지정 스킴을 바꿀 수 있고, 임의의 자동 값을 현재 효력있는 고정된 값으로 대체할 수 있다. 사용자는 또한, 요구되는 경우, 4개의 외부 앵커를 요구된 다른 그리드라인으로 이동시킴으로써, 그리드 라인 바운딩 박스를 조정할 수 있다. 오브젝트가 이동되거나 크기조정됨에 따라 그 새로운 위치가 재평가되고, 새로운 값이 계산됨에 따라 사용자가 표현한 임의의 명백한 요구를 모듈화한다. 로우 또는 칼럼이 추가 또는 삭제될 때, 근처의 오브젝트들이 재계산되고, 사용자가 표현한 임의의 명백한 요구를 모듈화한다.

예시적인 설계 방법은 오브젝트의 그리드라인 바운딩 박스를 결정하는 것을 포함한다. 상단 가장자리가 중간라인 상위에 있다면, 그 상단 여백은 고정된다. 하단 가장자리도 중간라인에 있다면, 그 높이는 고정되고 하단 여백은 동적인 것으로 설정된다. 그러나, 하단 가장자리가 중간라인 이하에 있다면, 그 하단 여백은 고정되고 그 높이는 동적인 것으로 설정된다. 지정은 새로운 그리드라인이 그려질 때 자동적으로 업데이트된다.

예로서, 화면 상에 그려질 수 있는 그리드라인이 있고, 오브젝트가 정상적으로 그려지고 수정될 수 있다(예를 들어, 오브젝트는 마음대로 그리드라인에 걸쳐있을 수 있다)고 가정하자. 그러한 환경에서, 도 9(여기에서는, 오브젝트의 예시적인 색으로서 "적색"이 제공됨)에 도시된 다수의 중첩하는 가로의 오브젝트(770)와 세로의 오브젝트(780)의 그림에 대해 다음의 예시적인 해결책이 제공된다.

```

<Grid Width="300" Height="300">
  <Row Size="*"/>
  <Row Size="*"/>
  <Row Size="*"/>
  <Column Size="*"/>
  <Column Size="*"/>
  <Column Size="*"/>

  <!-- Rows under the columns !-->

  <RoundedRect Fill="Red" Column="0" ColumnSpan="3" Row="1" RowSpan="1"
    LeftMargin="5" RightMargin="5" Width="Auto" TopMargin="5"
    BottomMargin="5" Height="Auto"
  />
  <RoundedRect Fill="Red" Column="0" ColumnSpan="3" Row="2" RowSpan="1"
    LeftMargin="5" RightMargin="5" Width="Auto" TopMargin="5"
    BottomMargin="5" Height="Auto"
  />

  <!-- Columns !-->

  <RoundedRect Fill="Red" Column="0" ColumnSpan="1" Row="0" RowSpan="3"
    LeftMargin="5" RightMargin="5" Width="Auto" TopMargin="5"
    BottomMargin="5" Height="Auto"
  />
  <RoundedRect Fill="Red" Column="0" ColumnSpan="1" Row="1" RowSpan="3"
    LeftMargin="5" RightMargin="5" Width="Auto" TopMargin="5"
    BottomMargin="5" Height="Auto"
  />
  <RoundedRect Fill="Red" Column="0" ColumnSpan="1" Row="2" RowSpan="3"
    LeftMargin="5" RightMargin="5" Width="Auto" TopMargin="5"
    BottomMargin="5" Height="Auto"
  />

  <!-- Top Row !-->

  <RoundedRect Fill="Red" Column="0" ColumnSpan="3" Row="0" RowSpan="1"
    LeftMargin="5" RightMargin="5" Width="Auto" TopMargin="5"
    BottomMargin="5" Height="Auto"
  />
</Grid>

```

[0058]

[0059]

칼럼과 로우 정보가 먼저 오고 자식들의 순서는 뽑기(draw) 순서이다. 또한, 각각의 오브젝트는 오브젝트가 내부에 맞는 그리드 사각형을 설명하는 속성을 첨부하였다. 여백은 지정된 그리드 사각형에 관하여 평가된다. 크기조정 동안 행위를 제어하는 추가적인 스내프토 정보가 추가될 수 있다. 그리드라인 값에 액세스한 시스템은 그리드 바운딩 박스, 및 바람직하게, 그리드 경계와 오프셋에 관한 사각형을 취한다.

[0060]

#### 예시적인 컴퓨팅 환경

[0061]

도 10은 본 발명이 구현될 수 있는 적절한 컴퓨팅 시스템 환경(800)의 예를 나타낸다. 컴퓨팅 시스템 환경(800)은 단지 적절한 컴퓨팅 환경의 일 예이며 본 발명의 사용 또는 기능의 범위에 제한을 가하도록 의도된 것은 아니다. 컴퓨팅 환경(800)은 예시적인 오피레이팅 환경(800)에 도시된 컴퓨터트들 중의 임의의 하나 또는 조합에 관하여 임의의 종속성(dependency) 또는 요구사항(requirement)을 갖는 것으로 해석되어서는 안된다.

[0062]

본 발명은 많은 다른 범용 또는 특수목적 컴퓨팅 시스템 환경들 또는 구성들과 함께 동작될 수 있다. 본 발명과 함께 사용하기에 적합할 수 있는 잘 알려진 컴퓨팅 시스템, 환경, 및/또는 구성의 예로는, 퍼스널 컴퓨터, 서버 컴퓨터, 핸드헬드(hand-held) 또는 랩탑 장치, 멀티프로세서 시스템, 마이크로프로세서-기반 시스템, 셋탑 박스(set top box), 프로그램가능한 가전제품(programmable consumer electronics), 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터, 상기의 시스템 또는 장치 중의 임의의 것을 포함하는 분산형 컴퓨팅 환경 등이 포함될 수 있지만, 이에 한정되지 않는다.

[0063]

본 발명은 컴퓨터에 의해 실행되는, 프로그램 모듈과 같은 컴퓨터 실행가능 명령과 일반적으로 관련하여 기술될 수 있다. 일반적으로, 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 데이터 유형을 구현하는 루틴, 프로그램, 오브젝트, 컴퓨터트, 데이터 구조 등을 포함한다. 본 발명은 또한 통신 네트워크 또는 다른 데이터 전송 매체를 통해 링크된 원격 프로세싱 장치에 의해 태스크를 수행하는 분산형 컴퓨팅 환경에서 실행될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈 및 그외 데이터는 메모리 저장 장치를 포함하는 국부 및 원격 컴퓨터 저장 매체 내에 위치할 수 있다.

[0064]

도 10을 참조하면, 본 발명을 구현하기 위한 예시적인 시스템은 컴퓨터(810)의 형태의 범용 컴퓨팅 장치를 포함한다. 컴퓨터(810)의 컴포넌트들로는, 프로세싱 유닛(820), 시스템 메모리(830), 및 시스템 메모리를 포함하는 다양한 시스템 컴포넌트를 프로세싱 유닛(820)에 연결시키는 시스템 버스(821)가 포함될 수 있지만, 이에 한정되는 것은 아니다. 시스템 버스(821)는 다양한 버스 아키텍처 중의 임의의 것을 사용하는 로컬 버스, 주변 버스, 및 메모리 버스 또는 메모리 컨트롤러를 포함하는 몇 가지 유형의 버스 구조 중의 임의의 것일 수 있다. 예로서, 이러한 아키텍처는 산업 표준 아키텍처(ISA) 버스, 마이크로 채널 아키텍처(MCA) 버스, 인핸드 ISA(Enhanced ISA; EISA) 버스, 비디오 일렉트로닉스 표준 어소시에이션(VESA) 로컬 버스, 및 (메자닌(Mezzanine) 버스로도 알려진) 주변 컴포넌트 상호접속(PCI) 버스를 포함하지만, 이에 한정되는 것은 아니다.

[0065]

컴퓨터(810)는 통상적으로 다양한 컴퓨터 판독가능 매체를 포함한다. 컴퓨터 판독가능 매체는 컴퓨터(810)에 의해 액세스될 수 있는 임의의 이용가능한 매체일 수 있으며, 휘발성 및 비휘발성 매체, 분리형(removable) 및 비분리형(non-removable) 매체를 둘다 포함한다. 예로서, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 포함할 수 있지만, 이에 한정되는 것은 아니다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령, 데이터 구조, 프로그램 모듈 또는 다른 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현되는 어플리케이션 및 비휘발성, 분리형 및 비분리형 매체를 둘다 포함한다. 컴퓨터 저장 매체는 RAM, ROM, EEPROM, 플래쉬 메모리 또는 기타 메모리 기술, CD-ROM, DVD(digital versatile disk) 또는 기타 광학 디스크 저장장치, 자기 카세트, 자기 테이프, 자기 디스크 저장장치 또는 기타 자기 저장장치, 또는 컴퓨터(810)에 의해 액세스될 수 있고 원하는 정보를 저장하는 데 사용될 수 있는 임의의 기타 매체를 포함할 수 있지만, 이에 한정되지 않는다. 통신 매체는 통상적으로 반송파 또는 기타 전송 메카니즘 등의 변조된 데이터 신호에 컴퓨터 판독가능 명령, 데이터 구조, 프로그램 모듈, 또는 다른 데이터를 구현하며, 임의의 정보 전달 매체를 포함한다. "변조된 데이터 신호"라는 용어는 신호 내에 정보를 인코딩하도록 설정되거나 변환된 특성을 하나 또는 그 이상을 갖는 신호를 의미한다. 예로서, 통신 매체는 유선 네트워크 또는 직접 유선 접속 등의 유선 매체와, 음향, RF, 적외선 및 기타 무선 매체 등의 무선 매체를 포함하지만, 이에 한정되지 않는다. 상술한 것들 중의의 임의의 조합이 컴퓨터 판독가능 매체의 범위 내에 포함되어야 한다.

[0066]

시스템 메모리(830)는 ROM(831) 및 RAM(832) 등의 휘발성 및/또는 비휘발성 메모리의 형태의 컴퓨터 저장 매체를 포함한다. 시동중과 같은 때에 컴퓨터(810) 내의 구성요소들간에 정보를 전송하는 것을 돋는 기본 루틴을 포함하는 기본 입출력 시스템(833; BIOS)은 일반적으로 ROM(831)에 저장된다. RAM(832)은 일반적으로 프로세싱 유닛(820)에 즉시 액세스될 수 있고 및/또는 프로세싱 유닛(820)에 의해 현재 작동되는 프로그램 모듈 및/또는 데이터를 포함한다. 예로서, (한정하고자 하는 것은 아님) 도 10은 오퍼레이팅 시스템(834), 어플리케이션 프로그램(835), 기타 프로그램 모듈(어플리케이션), 및 프로그램 데이터(어플리케이션)를 도시한다.

[0067]

컴퓨터(810)는 또한 다른 분리형/비분리형, 휘발성/비휘발성 컴퓨터 저장 매체를 포함할 수 있다. 단지 예로서, 도 10에는 비분리형 비휘발성 자기 매체로부터 판독하거나 그 자기 매체에 기록하는 하드 디스크 드라이브(840), 분리형 비휘발성 자기 디스크(852)로부터 판독하거나 그 자기 디스크에 기록하는 자기 디스크 드라이브(851), 및 CD-ROM 또는 기타 광학 매체 등의 분리형 비휘발성 광학 디스크(856)로부터 판독하거나 그 광학 디스크에 기록하는 광학 디스크 드라이브(855)가 도시되어 있다. 예시적인 오퍼레이팅 환경에서 사용될 수 있는 다른 분리형/비분리형, 휘발성/비휘발성 컴퓨터 저장 매체는 자기 테이프 카세트, 플래쉬 메모리 카드, DVD(Digital versatile disk), 디지털 비디오 테이프, 고체 RAM, 고체 ROM 등을 포함하지만 이에 한정되지 않는다. 하드 디스크 드라이브(841)는 일반적으로 인터페이스(840)와 같은 비분리형 메모리 인터페이스를 통해 시스템 버스(821)에 접속되고, 자기 디스크 드라이브(851) 및 광학 디스크 드라이브(855)는 일반적으로 인터페이스(850)와 같은 분리형 메모리 인터페이스에 의해 시스템 버스(821)에 접속된다.

[0068]

드라이브 및 그 관련 컴퓨터 저장 매체는 컴퓨터(810)를 위한 컴퓨터 판독가능 명령, 데이터 구조, 프로그램 모듈 및 기타 데이터의 저장을 제공한다. 도 10에서, 예를 들어, 하드 디스크 드라이브(841)는 오퍼레이팅 시스템(844), 어플리케이션 프로그램(845), 기타 프로그램 모듈(846), 및 프로그램 데이터(847)를 저장하는 것으로 도시된다. 이를 컴포넌트는 오퍼레이팅 시스템(834), 어플리케이션 프로그램(835), 기타 프로그램 모듈(836), 및 프로그램 데이터(837)와 동일할 수도 있고 다를 수도 있다. 오퍼레이팅 시스템(844), 어플리케이션 프로그램(845), 다른 프로그램 모듈(846), 및 프로그램 데이터(847)는 최소한 다른 복사본(different copies)임을 나타내기 위하여 다른 번호를 부여하였다. 사용자는 일반적으로 마우스, 트랙볼, 또는 터치 패드라 불리우는 포인팅 장치(861) 및 키보드(862)와 같은 입력 장치를 통해 컴퓨터(810)에 명령 및 정보를 입력할 수 있다. (도시되지 않은) 기타 입력 장치는 마이크로폰, 조이스틱, 게임 패드, 위성 안테나, 스캐너 등을 포함할 수 있다. 이를 입력 장치 및 그 외의 입력 장치는 시스템 버스에 연결된 사용자 입력 인터페이스(860)를 통해 종종 프로세

성 유닛(어플리케이션)에 접속되지만, 병렬 포트, 게임 포트 또는 유니버설 시리얼 포트(USB) 와 같은 기타 인터페이스 및 버스 구조에 의해 접속될 수 있다. 모니터(891) 또는 다른 유형의 디스플레이 장치는 또한 비디오 인터페이스(890) 등의 인터페이스를 통해 시스템 버스(821)에 접속된다. 모니터외에도, 컴퓨터는 또한 출력 주변 인터페이스(어플리케이션)를 통해 접속될 수 있는 스피커(897) 및 프린터(896) 등의 기타 주변 출력 장치를 포함할 수 있다.

[0069] 컴퓨터(810)는 원격 컴퓨터(880)와 같은 하나 이상의 원격 컴퓨터로의 논리적 접속을 이용한 네트워크 환경에서 동작할 수 있다. 원격 컴퓨터(880)는 퍼스널 컴퓨터, 서버, 라우터, 네트워크 PC, 피어(peer) 장치, 또는 기타 공통 네트워크 노드일 수 있으며, 비록 도 10 에는 메모리 저장 장치(881)만이 도시되어 있지만, 컴퓨터(810)에 관하여 상술한 구성요소 중 다수 또는 모든 구성요소를 일반적으로 포함할 수 있다. 도시된 논리적 접속은 근거리 통신망(LAN; 어플리케이션) 및 원거리 통신망(WAN; 873)을 포함하지만, 그 외의 네트워크를 포함할 수도 있다. 이러한 네트워크 환경은 사무실, 기업 광역 컴퓨터 네트워크(enterprise-wide computer network), 인트라넷, 및 인터넷에서 일반적인 것이다.

[0070] LAN 네트워크 환경에서 사용되는 경우, 컴퓨터(810)는 네트워크 인터페이스 또는 어댑터(870)를 통해 LAN(871)에 접속된다. WAN 네트워크 환경에서 사용되는 경우, 컴퓨터(810)는 일반적으로 인터넷 등의 WAN(873)을 통해 통신을 구축하기 위한 모뎀(872) 또는 기타 수단을 포함한다. 내장형 또는 외장형일 수 있는 모뎀(872)은 사용자 입력 인터페이스(860) 또는 기타 적절한 메카니즘을 통해 시스템 버스(821)에 접속될 수 있다. 네트워크 환경에서, 컴퓨터(810)에 관하여 도시된 프로그램 모듈 또는 그 일부분은 원격 메모리 저장 장치에 저장될 수 있다. 예로서 (한정하고자 하는 것은 아님), 도 10은 메모리 장치(881)에 상주하는 원격 어플리케이션 프로그램(885)을 도시한다. 도시된 네트워크 접속은 예시적인 것이며, 컴퓨터들간의 통신 링크를 구축하는 그 외의 수단이 사용될 수 있다.

[0071] 상술된 바와 같이, 본 발명의 예시적인 실시예가 여러 컴퓨팅 장치와 관련하여 설명되었지만, 그 기저 개념은 임의의 컴퓨팅 장치 또는 시스템에 적용될 수 있다.

[0072] 여기에서 설명된 여러가지 기술은 하드웨어 또는 소프트웨어와 관련하여, 또는 적절하게 양자의 조합과 관련하여 구현될 수도 있다. 따라서, 본 발명의 방법 및 장치, 또는 그의 특정 양태 또는 일부는 플로피 디스크, CD-ROM, 하드 드라이브 또는 임의의 다른 기계-판독가능 저장 매체와 같은 실제 매체에 포함된 프로그램 코드(즉, 명령)의 형태를 취할 수 있고, 여기서, 프로그램 코드가 컴퓨터와 같은 기계에 의해 로딩되고 실행될 때, 기계는 본 발명을 실행하기 위한 장치가 된다. 프로그램가능한 컴퓨터 상의 프로그램 코드 실행의 경우에, 컴퓨팅 장치는 일반적으로 프로세서, 프로세서에 의해 판독가능한 저장 매체(휘발성 및 비휘발성 메모리 및/또는 저장 엘리먼트를 포함함), 적어도 하나의 입력 장치, 및 적어도 하나의 출력 장치를 포함할 것이다. 프로그램(들)은 요구된다면, 어셈블리 또는 기계 언어로 구현될 수 있다. 임의의 경우에, 언어는 컴파일되거나 인터프리트된 언어일 수 있고 하드웨어 구현으로 조합될 수 있다.

[0073] 본 발명의 방법 및 장치는 또한, 광섬유를 통해 전자 배선 또는 케이블과 같은 일부 전송 매체 상에서 전송되는 프로그램 코드의 형태로 구현된 통신을 통해, 또는 임의의 다른 전송 형태를 통해 실행될 수 있고, 여기서, 프로그램 코드가 수신되어, EPROM, 게리트 어레이, 프로그램가능한 논리 장치(PLD), 클라이언트 컴퓨터 등과 같은 기계에 의해 로딩되고 실행될 때, 기계는 본 발명을 실행하기 위한 장치가 된다. 범용 프로세서 상에서 구현될 때, 프로그램 코드는 본 발명의 기능을 야기하도록 동작하는 유일한 장치를 제공하는 프로세서와 결합한다. 또한, 본 발명과 관련되어 사용된 임의의 저장 기술은 반드시 하드웨어와 소프트웨어의 조합일 수 있다.

[0074] 본 발명은 여러 도면의 바람직한 실시예와 관련하여 설명되었지만, 다른 유사한 실시예가 사용될 수도 있고, 본 발명의 범위에서 벗어나지 않고, 본 발명의 동일한 기능을 수행하기 위한 상술된 실시예에 변경 및 추가가 행해질 수 있다. 따라서, 본 발명은 임의의 단일 실시예로 한정되어서는 안되며, 오히려, 첨부 청구범위에 따른 범주 내에서 해석되어야 한다.

### 발명의 효과

[0075] 본 발명은 복잡하지 않게 지능적으로 크기조정하는 레이아웃을 제공한다.

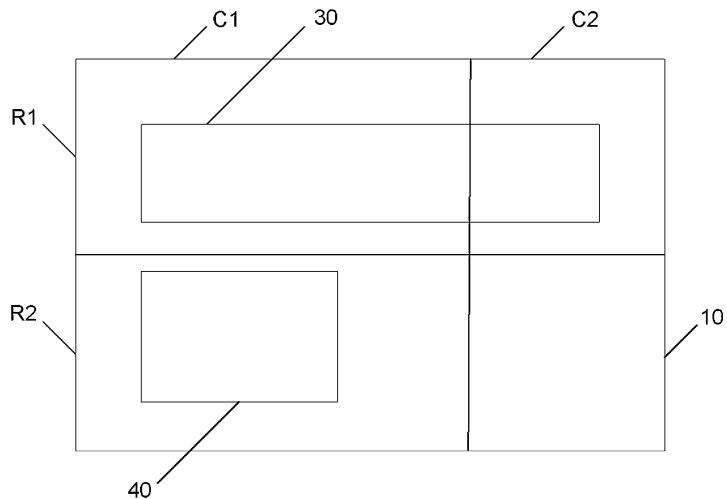
### 도면의 간단한 설명

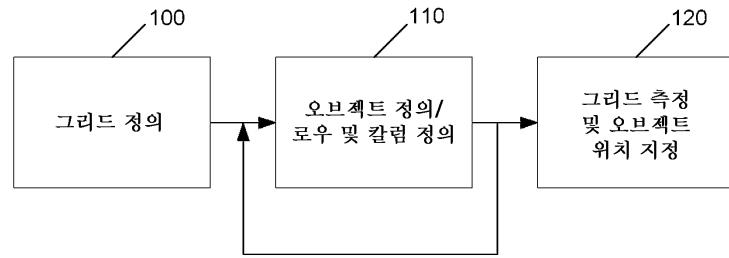
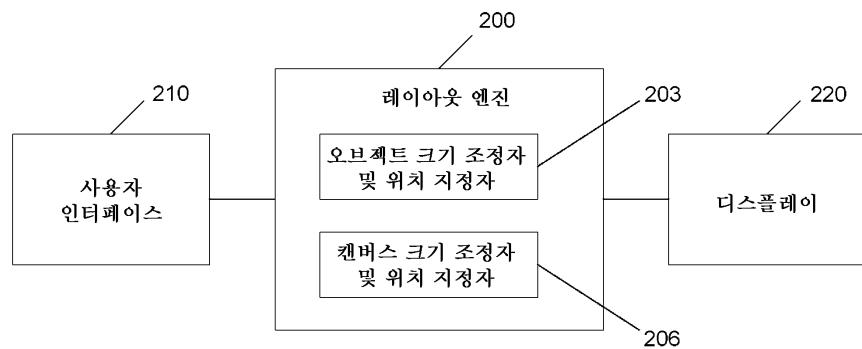
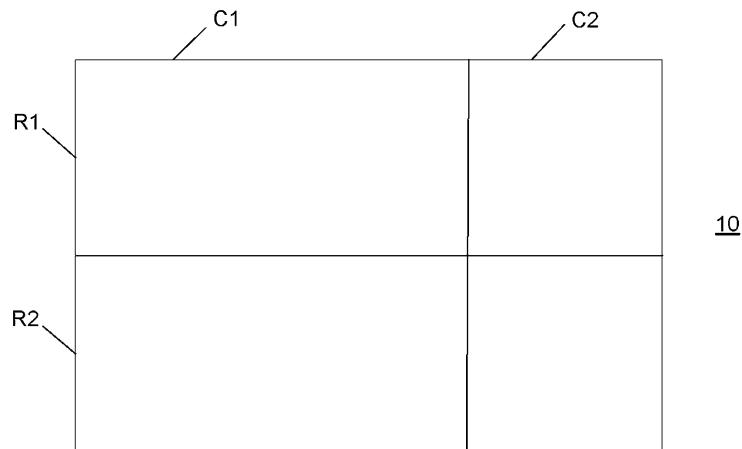
[0001] 도 1은 본 발명에 따른 오브젝트들을 갖는 예시적인 그리드 캔버스의 블록도.

- [0002] 도 2는 본 발명에 따른 그리드 캔버스를 생성하는 예시적인 방법의 고레벨 흐름도.
- [0003] 도 3은 본 발명에 따른 예시적인 시스템의 고레벨 블럭도.
- [0004] 도 4는 본 발명에 따른 또다른 예시적인 그리드 캔버스의 블럭도.
- [0005] 도 5는 본 발명에 따른 또다른 예시적인 그리드 캔버스의 블럭도.
- [0006] 도 6은 본 발명에 따른 또다른 예시적인 그리드 캔버스의 블럭도.
- [0007] 도 7은 본 발명에 따른 그리드 캔버스 레이아웃을 생성하는 예시적인 방법을 도시.
- [0008] 도 8은 본 발명에 따른 또다른 예시적인 그리드 캔버스의 블럭도.
- [0009] 도 9는 본 발명에 따른 다수의 중첩하는 오브젝트들의 예시적인 블럭도.
- [0010] 도 10은 본 발명의 양태가 구현될 수 있는 예시적인 컴퓨팅 환경을 도시하는 블럭도.
- [0011] <도면의 주요 부분에 대한 부호의 설명>
- [0012] 200 : 레이아웃 엔진
- [0013] 203 : 크기조정자 및 위치지정자
- [0014] 206 : 캔버스 크기조정자 및 위치지정자
- [0015] 210 : 사용자 인터페이스
- [0016] 220: 디스플레이

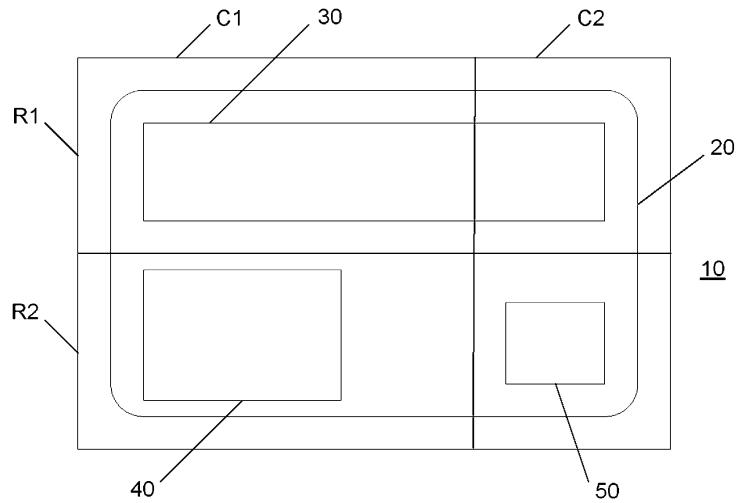
## 도면

### 도면1

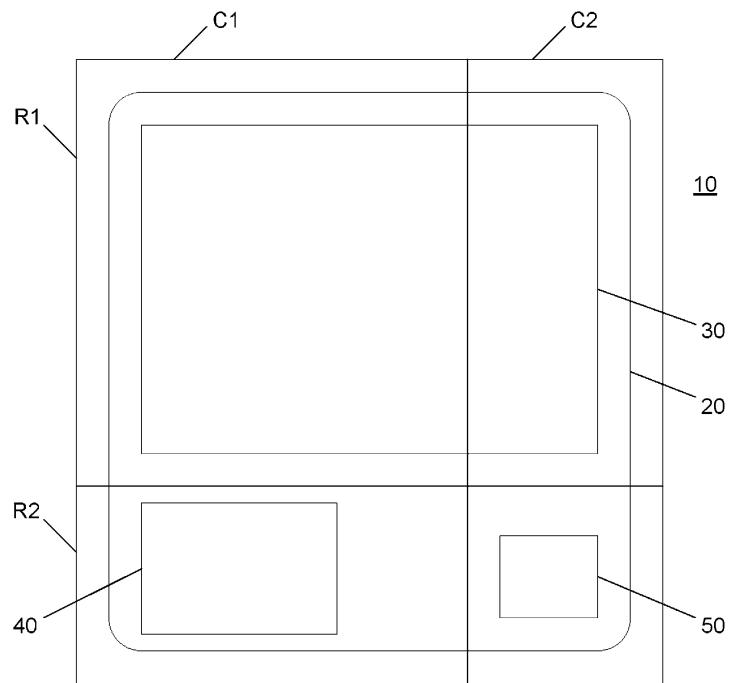


**도면2****도면3****도면4**

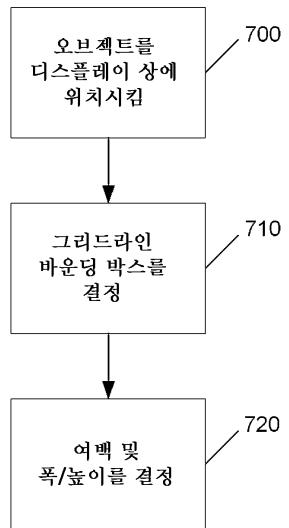
도면5



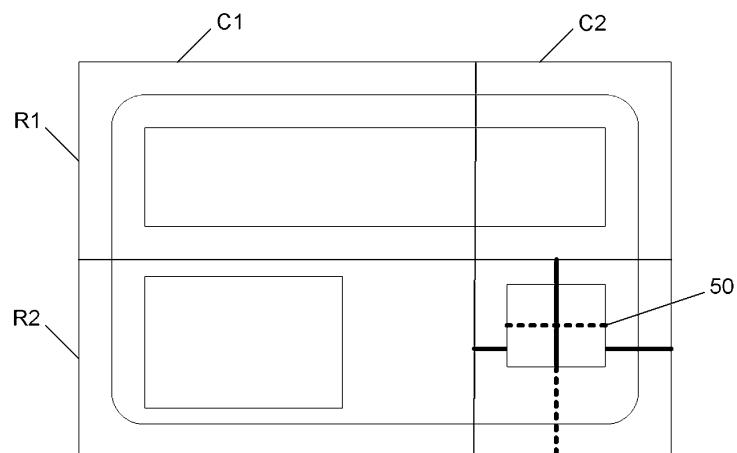
도면6



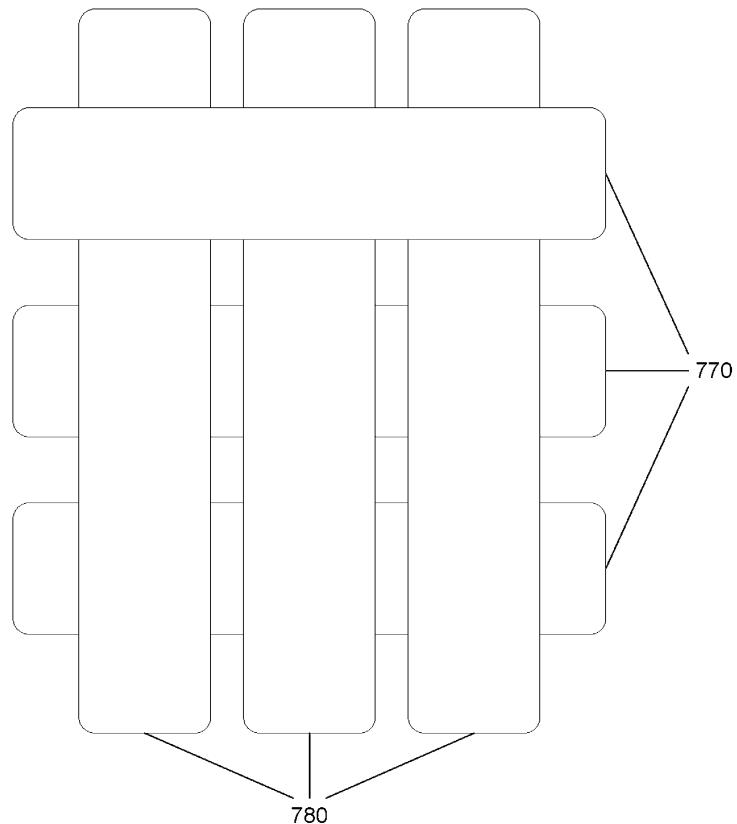
도면7



도면8



도면9



## 도면10

