



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2018년11월19일

(11) 등록번호 10-1919898

(24) 등록일자 2018년11월13일

(51) 국제특허분류(Int. Cl.)

A63F 13/822 (2014.01) A63F 13/55 (2014.01)

A63F 13/65 (2014.01) A63F 13/828 (2014.01)

(52) CPC특허분류

A63F 13/822 (2015.01)

A63F 13/55 (2015.01)

(21) 출원번호 10-2016-0098421

(22) 출원일자 2016년08월02일

심사청구일자 2016년08월02일

(65) 공개번호 10-2017-0016801

(43) 공개일자 2017년02월14일

(30) 우선권주장

15/224,427 2016년07월29일 미국(US)

62/201,004 2015년08월04일 미국(US)

(56) 선행기술조사문현

KR1020100069730 A

JP2015073745 A

KR100507587 B1

JP2015008984 A

(73) 특허권자

핫헤드 게임즈 인크.

캐나다 (우편번호 브이6지2티1) 브리티시 콜럼비아 밴쿠버 더블유 웬더 스트리트 1555

(72) 발명자

닐슨 스캇

캐나다 비씨 브이6지 2티1 밴쿠버 웨스트 웬터 스트리트 1555 핫헤드 게임즈 인코퍼레이티드

아이피 제이미

캐나다 비씨 브이6지 2티1 밴쿠버 웨스트 웬터 스트리트 1555 핫헤드 게임즈 인코퍼레이티드

잉글하트 마이크

캐나다 비씨 브이6지 2티1 밴쿠버 웨스트 웬터 스트리트 1555 핫헤드 게임즈 인코퍼레이티드

(74) 대리인

특허법인와이에스장

전체 청구항 수 : 총 21 항

심사관 : 이동하

(54) 발명의 명칭 가상 카드 플레이 기반 상호작용 게임 경험

**(57) 요 약**

가상 카드 플레이 기반의 상호작용 게임 경험이 여기 서술된다. 몇몇 방법은 한 쌍의 가상 카드 각각에 대한 하나 이상의 스킬을 선택하는 단계, 차이 값을 정립하기 위해 가상 카드 쌍 각각에 대한 하나 이상의 스킬을 이용하여 가상 카드 쌍의 비교 분석을 포함하는, 가상 카드 상을 이용하는 가상 카드 배틀을 수행하는 단계, 및 차이 값을 이용하여 선택된 난이도 레벨을 가지는 가상 활동을 생성하는 단계를 포함한다. 가상 활동은 가상 카드 배틀의 상호작용 버전이다.

(52) CPC특허분류

*A63F 13/65* (2015.01)

*A63F 13/828* (2015.01)

*A63F 2300/807* (2013.01)

---

## 명세서

### 청구범위

#### 청구항 1

가상 게임 시스템의 컴퓨터 구현 방법으로서, 상기 가상 게임 시스템에 의해 구현되는 상기 방법은:

가상 카드 배틀을 위한 적어도 하나의 한 쌍의 가상 카드의 선택을 수신하는 단계;

상기 가상 카드 쌍의 각각에 대한 하나 이상의 스킬을 선택하는 단계;

상기 하나 이상의 스킬을 이용한 비교 분석을 수행함으로써 가상 카드 배틀을 실행하여 차이 값을 계산하는 단계;

가상 활동에 대한 복수의 속성을 선택하기 위해 상기 차이 값을 이용하는 단계로서, 상기 가상 활동은 상기 가상 카드 배틀의 상호작용 버전인 상기 가상 활동에 대한 복수의 속성을 선택하기 위해 상기 차이 값을 이용하는 단계; 및

상기 복수의 속성을 가지는 상기 가상 활동을 상기 가상 카드 배틀의 플레이어에게 제공하는 단계를 포함하는 것을 특징으로 하는 방법.

#### 청구항 2

제1 항에 있어서, 상기 가상 카드 쌍 중 첫 번째 것은 상기 플레이어에게 속하고, 상기 가상 카드 쌍 중 두 번째 것은 제2 플레이어에게 속하는 것을 특징으로 하는 방법.

#### 청구항 3

제1 항에 있어서, 상기 하나 이상의 스킬은 가상 게임 시스템에 의해 랜덤하게 선택되는 것을 특징으로 하는 방법.

#### 청구항 4

제1 항에 있어서, 상기 가상 카드 쌍 부근에 상기 차이 값을 디스플레이 하는 단계를 더 포함하는 것을 특징으로 하는 방법.

#### 청구항 5

제1 항에 있어서, 상기 가상 활동에 대하여 선택된 상기 복수의 속성은 증가되거나 감소되는 난이도 레벨을 포함하는 것을 특징으로 하는 방법.

#### 청구항 6

제5 항에 있어서, 상기 난이도 레벨은 상기 차이 값의 크기에 따라 증가되거나 감소되는 것을 특징으로 하는 방법.

#### 청구항 7

제5 항에 있어서, 상기 복수의 속성의 개수가 상기 차이 값의 크기에 따라 증가되거나 감소되는 것을 특징으로 하는 방법.

#### 청구항 8

제1 항에 있어서, 상기 차이 값을 차이 값 범위와 비교함으로써 상기 가상 카드 배틀에 대한 난이도 레벨을 판정하는 단계를 더 포함하고, 상기 차이 값 범위는 상기 복수의 속성을 정의하는 다양한 난이도 레벨과 연관되어 있는 것을 특징으로 하는 방법.

#### 청구항 9

가상 게임 시스템의 컴퓨터 구현 방법으로서, 상기 가상 게임 시스템에 의해 구현되는 상기 방법은:

제1 파티의 가상 카드의 선택을 수신하는 단계;

제2 파티의 가상 카드의 선택을 수신하는 단계;

가상 카드 배틀을 위한 한 쌍의 가상 카드를 선택하는 단계로서, 상기 가상 카드 쌍은 상기 제1 파티의 가상 카드 중 하나 및 상기 제2 파티의 가상 카드 중 하나를 포함하는 것인, 상기 가상 카드 배틀을 위한 한 쌍의 가상 카드를 선택하는 단계;

상기 가상 카드 쌍 각각에 대한 하나 이상의 스킬을 선택하는 단계;

차이 값을 정립하기 위해 상기 가상 카드 쌍 각각에 대한 상기 하나 이상의 스킬을 이용하여 상기 가상 카드 쌍의 비교 분석을 수행하는 단계;

상기 차이 값을 이용하여 가상 활동에 대한 복수의 속성을 선택하는 단계를 포함하고,

상기 가상 활동은 상기 가상 카드 배틀의 상호작용 버전인 것을 특징으로 하는 방법.

#### 청구항 10

제9 항에 있어서, 상기 비교 분석은 상기 가상 카드 쌍 각각에 대한 상기 하나 이상의 스킬을 난이도 레벨 범위에 대해 비교하여 상기 차이 값을 계산하는 단계를 더 포함하는 것을 특징으로 하는 방법.

#### 청구항 11

제9 항에 있어서, 상기 가상 카드 배틀의 플레이어에 대한 상기 복수의 속성을 가진 상기 가상 활동을 실행하는 단계를 더 포함하는 것을 특징으로 하는 방법.

#### 청구항 12

가상 게임 시스템의 컴퓨터 구현 방법으로서, 상기 가상 게임 시스템에 의해 구현되는 상기 방법은:

한 쌍의 가상 카드 각각에 대한 하나 이상의 스킬을 선택하는 단계로서, 상기 하나 이상의 스킬은 각각 스킬 값을 가지는 것인, 상기 한 쌍의 가상 카드 각각에 대한 하나 이상의 스킬을 선택하는 단계;

가상 카드 배틀을 위한 가상 카드 배틀 타입을 랜덤하게 선택하는 단계;

차이 값을 정립하기 위해 상기 가상 카드 쌍 각각에 대한 상기 하나 이상의 스킬 값을 이용한 상기 가상 카드 쌍의 비교 분석을 포함하는, 상기 가상 카드 쌍을 이용한 가상 카드 배틀을 수행하는 단계; 및

상기 차이 값을 이용하여, 선택된 난이도 레벨을 가지는 가상 활동을 생성하는 단계를 포함하고,

상기 가상 활동은 상기 가상 카드 배틀의 상호작용 버전인 것을 특징으로 하는 방법.

#### 청구항 13

제12 항에 있어서, 상기 가상 카드 쌍 각각은 운동선수를 나타내는 것을 특징으로 하는 방법.

#### 청구항 14

제12 항에 있어서, 상기 하나 이상의 스킬은 운동선수와 연관된 스포츠와 관련된 것을 특징으로 하는 방법.

#### 청구항 15

제13 항에 있어서, 상기 가상 활동은 스포츠와 관련된 3차원 스포츠 활동인 것을 특징으로 하는 방법.

#### 청구항 16

제14 항에 있어서, 상기 난이도 레벨은 상대 플레이어의 수, 환경적 조건, 타겟 영역 또는 크기, 조준 안정성, 타겟 거리, 및 이들의 조합 중 임의의 것으로부터 선택된 복수의 속성을 포함하는 것을 특징으로 하는 방법.

#### 청구항 17

제15 항에 있어서, 상기 난이도 레벨은 상기 차이 값을 크기에 따라 증가 또는 감소되는 것을 특징으로 하는 방법.

**청구항 18**

제16 항에 있어서, 상기 가상 카드 쌍은 랜덤하게 선택되는 것을 특징으로 하는 방법.

**청구항 19**

제17 항에 있어서, 상기 가상 활동에 대하여 선택된 속성은 상기 가상 카드 배틀의 주제(subject matter)를 기초로 선택되는 것을 특징으로 하는 방법.

**청구항 20**

제18 항에 있어서, 상기 속성은 상기 차이 값이 속하는 난이도 레벨 범위를 기초로 더 선택되는 것을 특징으로 하는 방법.

**청구항 21**

가상 게임 시스템의 컴퓨터 구현 방법으로서, 상기 가상 게임 시스템에 의해 구현되는 상기 방법은:

제1 파티의 가상 카드의 선택을 수신하는 단계;

제2 파티의 가상 카드의 선택을 수신하는 단계;

가상 카드 배틀을 위한 한 쌍의 가상 카드를 선택하는 단계로서, 상기 가상 카드 쌍은 상기 제1 파티의 가상 카드 중 하나 및 상기 제2 파티의 가상 카드 중 하나를 포함하는 것인, 상기 가상 카드 배틀을 위한 한 쌍의 가상 카드를 선택하는 단계;

상기 가상 카드 쌍 각각에 대한 하나 이상의 스킬을 선택하는 단계;

차이 값을 정립하기 위해 상기 가상 카드 쌍 각각에 대한 상기 하나 이상의 스킬을 이용하여 상기 가상 카드 쌍의 비교 분석을 수행하는 단계;

차이 값 범위를 포함하는 테이블의 검색(lookup)을 수행하는 단계로서, 각각의 범위는 상기 가상 카드 배틀을 기초로 하는 가상 활동에 대한 난이도 레벨과 연관되어 있고, 상기 검색은 계산된 차이 값이 어떤 차이 값 범위에 속하는지 판정하는 것인, 상기 차이 값 범위를 포함하는 테이블의 검색을 수행하는 단계;

플레이어가 상기 가상 활동에 참여하는 방법을 정의하는 복수의 선택된 가상 활동 파라미터 각각에 대하여 불연속 확률 밀도 계산을 수행하는 단계; 및

상기 복수의 선택된 가상 활동 파라미터 각각에 대한 값을 다이내믹하게 선택하기 위해 난수 스크립팅을 수행하는 단계; 및

상기 플레이어에 대하여 상기 가상 활동을 실행하는 단계를 포함하는 것을 특징으로 하는 방법.

**발명의 설명****기술 분야**

[0001]

본 발명은 일반적으로 가상 게임에 관한 것이고, 제한하는 것은 아니지만 더욱 상세하게는, 몇몇 실시예에서, 가상 카드 배틀이 발생하는 다면적인(multifaceted) 상호작용 게임 경험을 제공하는 시스템 및 방법에 관한 것이다. 가상 카드 배틀의 결과는 게임 내의 도전과제(challenge) 또는 활동의 난이도 레벨에 영향을 준다.

**발명의 내용****해결하려는 과제****과제의 해결 수단**

[0002]

몇몇 실시예에 따라, 본 발명은 (a) 가상 카드 배틀을 위한 한 쌍의 가상 카드 중 적어도 하나의 선택을 수신하는 단계; (b) 가상 카드 쌍 각각에 대한 하나 이상의 스킬을 선택하는 단계; (c) 차이 값을 계산하기 위해 하나 이상의 스킬을 이용하여 비교 분석을 수행함으로써 가상 카드 배틀을 실행하는 단계; (d) 가상 활동에 대한 복

수의 속성을 선택하기 위해 차이 값을 이용하는 단계로서, 상기 가상 활동은 가상 카드 배틀의 상호작용 버전인, 상기 가상 활동에 대한 복수의 속성을 선택하기 위해 차이 값을 이용하는 단계; 및 (e) 복수의 속성을 가진 가상 활동을 가상 카드 배틀의 플레이어에게 제공하는 단계를 포함하는 방법에 관한 것이다.

[0003] 몇몇 실시예에 따라, 본 발명은 (a) 제1 파티의 가상 카드의 선택을 수신하는 단계; (b) 제2 파티의 가상 카드의 선택을 수신하는 단계; (c) 가상 카드 배틀을 위한 한 쌍의 가상 카드를 선택하는 단계로서, 가상 카드 쌍은 제1 파티의 가상 카드 중 하나 및 제2 파티의 가상 카드 중 하나를 포함하는 것인, 상기 가상 카드 배틀을 위한 한 쌍의 가상 카드를 선택하는 단계; (d) 가상 카드 쌍 각각에 대한 하나 이상의 스킬을 선택하는 단계; (e) 차이 값을 정립하기 위해 가상 카드 쌍 각각에 대한 하나 이상의 스킬을 이용하여 가상 카드 쌍의 비교 분석을 수행하는 단계; 및 (f) 차이 값을 이용하여 가상 활동에 대한 복수의 속성을 선택하는 단계를 포함하는 방법에 관한 것이고, 여기서 가상 활동은 가상 카드 배틀의 상호작용 버전이다.

[0004] 몇몇 실시예에 따라, 본 발명은 (a) 가상 카드 쌍 각각에 대한 하나 이상의 스킬을 선택하는 단계로서, 하나 이상의 스킬 각각은 스킬 값을 가지는, 상기 가상 카드 쌍 각각에 대한 하나 이상의 스킬을 선택하는 단계; (b) 가상 카드 배틀을 위한 가상 카드 배틀 타입을 랜덤하게 선택하는 단계; (c) 차이 값을 정립하기 위해 가상 카드 쌍 각각에 대한 하나 이상의 스킬을 이용한 가상 카드 쌍의 비교 분석을 포함하는, 가상 카드 쌍을 이용한 가상 카드 배틀을 수행하는 단계; 및 (d) 차이 값을 이용하여, 선택된 난이도 레벨을 가지는 가상 활동을 생성하는 단계를 포함하는 방법에 관한 것이고, 여기서 가상 활동을 가상 카드 배틀의 상호작용 버전이다.

### 도면의 간단한 설명

[0005] 본 발명의 몇몇 실시예들이 첨부된 도면에 도시되어 있다. 이 도면이 반드시 축척에 따른 것은 아니며, 그 세부사항들이 기술의 이해를 위해 필수적인 것이 아니고, 또는 이해하기 어렵게 만드는 다른 세부사항들이 생략될 수도 있음이 이해될 것이다. 본 발명이 필연적으로 여기 도시된 특정 실시예로 제한되는 것은 아님이 이해될 것이다.

도 1은 본 발명의 형태를 실시하기 위해 사용될 수 있는 예시적인 컴퓨팅 아키텍처의 개략적인 도면이다.

도 2는 하나 이상의 가상 카드 배틀을 위한 가상 카드의 선택을 도시하는 예시적인 그래픽 사용자 인터페이스 (GUI)의 스크린샷이다.

도 3은 가상 카드의 다양한 쌍에 대한 가상 카드 배틀의 일부분에 대한 차별적인 점수를 도시하는 예시적인 GUI의 스크린샷이다.

도 4는 각각의 가상 카드에 대한 스킬 값을 보여주는 예시적인 GUI의 스크린샷이다.

도 5는 가상 카드 배틀을 위해 함께 매칭된 모든 가상 카드 쌍에 대한 차별적인 점수를 도시하는 예시적인 GUI의 스크린샷이다.

도 6은 가상 활동 개시 스크린을 보여주는 예시적인 GUI의 스크린샷이다.

도 7은 가상 카드 배틀을 위한 헤드투헤드(head-to-head) 매치업을 보여주는 예시적인 GUI의 스크린샷이다.

도 8은 가상 카드 배틀, 그 가상 카드 배틀을 위해 계산된 차이 값, 그 차이 값을 기초로 가상 활동에 대하여 선택된 난이도 레벨을 기초로 하여 가상 활동에 대하여 선택된 플레이어를 보여주는 예시적인 GUI의 스크린샷이다.

도 9는 가상 활동을 위한 타겟 사이징을 보여주는 예시적인 GUI의 스크린샷이다.

도 10은 가상 활동에 대한 플레이어 참여를 보여주는 예시적인 GUI의 스크린샷이다. 이 가상 활동은 가상 카드 배틀의 차이 값을 기초로 하는 난이도로 구성된다.

도 11은 본 발명의 예시적인 방법의 흐름도이다.

도 12는 본 발명의 다른 예시적인 방법의 흐름도이다.

도 13은 본 발명의 형태를 구현하기 위해 사용될 수 있는 예시적인 컴퓨팅 장치의 개략적인 도면이다.

### 발명을 실시하기 위한 구체적인 내용

[0006] 본 발명이 다수의 상이한 형태의 실시예들이 가능하지만, 몇몇 특수한 실시예들이 도면에 도시되어 있고 지금부

터 상세하게 설명될 것인데, 본 개시물이 본 발명의 원리의 예시로서 간주되어야 하고, 본 발명을 도시된 실시 예로 제한하도록 의도된 것은 아님을 이해해야 한다.

[0007] 여기 사용된 용어는 단지 특정 실시예를 설명할 목적일 뿐이며, 본 발명을 제한할 의도가 아니다. 여기서 사용된 단수 형태 "하나", "하나의", "그"는 문맥에서 분명히 다르게 지시되지 않았다면, 복수의 형태도 포함하도록 의도되었다. 또한, 용어 "포함하다" 및/또는 "포함하는"이 본 명세서에서 사용된 때, 이는 언급된 피처, 정수, 단계, 동작, 엘리먼트, 및/또는 컴포넌트의 존재를 명시한 것이며, 하나 이상의 다른 피처, 정수, 단계, 동작, 엘리먼트, 컴포넌트, 및/또는 이들의 그룹의 존재 또는 추가를 배제하는 것이 아님을 더 이해해야 한다.

[0008] 여기 언급된 유사한 또는 닮은 엘리먼트 및/또는 컴포넌트들이 도면 전체에 걸친 유사한 부재번호를 통해 식별될 수 있음이 이해될 것이다. 또한, 몇몇 도면은 단지 본 발명의 개략적인 표현일 뿐임이 이해될 것이다. 이처럼, 몇몇 컴포넌트는 도식적인 명료함을 위해 그 실제 축적과 다르게 그려져 있을 수 있다.

[0009] 본 발명은 가상 게임 경험을 제공하기 위한 시스템 및 방법에 관한 것이다. 몇몇 실시예에서, 가상 게임 경험은 가상 카드 배틀 및 가상 게임 활동을 모두 포함한다. 몇몇 실시예에 따라, 가상 게임 활동의 초기 조건은 가상 카드 배틀의 결과를, 부분적으로, 기초로 한다. 하나의 실시예에서, 초기 조건은 가상 게임 활동에 대한 난이도 레벨을 나타낸다.

[0010] 도 1은 본 발명의 형태를 실시하기 위한 예시적인 아키텍처를 도시한다. 이 아키텍처는 본 명세서 전반에 걸쳐 더 상세하게 서술되어 있는 다양한 기능을 제공하도록 구성된 서버 시스템(이하 시스템(105)이라 함)을 포함한다. 일반적으로, 시스템(105)은 클라이언트(115)와 같은 클라이언트 장치와 통신하도록 구성되어 있다. 클라이언트(115)는, 예컨대, 스마트폰, 태블릿, 컴퓨터, 또는 다른 유사한 컴퓨팅 장치를 포함할 수 있다. 본 발명에 따라 사용될 수 있는 컴퓨팅 장치의 예는 도 13과 관련지어 더 상세하게 설명된다.

[0011] 시스템(105)은 네트워크(120)와 같은 공중 또는 사설 네트워크를 통해 클라이언트(115)와 통신적으로 연결될 수 있다. 적절한 네트워크는, 예컨대, 로컬 인터넷, PAN(Personal Area Network), LAN(Local Area Network), WAN(Wide Area Network), MAN(Metropolitan Area Network), 가상 사설망(VPN), 기억 영역 네트워크(SAN: storage area network), 프레임 릴레이 연결, AIN(Advanced Intelligent Network) 연결, 동기식 광 통신망(SONET: synchronous optical network) 연결, 디지털 T1, T3, E1 또는 E3 라인, 디지털 데이터 서비스(DDS: Digital Data Service) 연결, 디지털 가입자 회선(DSL: Digital Subscriber Line) 연결, 이더넷 연결, 종합정보통신망(ISDN: Integrated Services Digital Network) 라인, V.90, V.34 또는 V.34비스 아날로그 모뎀 연결과 같은 다이얼업 포트(dial-up port), 케이블 모뎀, 비동기식 전송 방식(ATM: Asynchronous Transfer Mode) 연결, 또는 파이버 분산형 데이터 인터페이스(FDDI: Fiber Distributed Data Interface) 또는 동선 분산형 데이터 인터페이스(CDDI :Copper Distributed Data Interface) 연결 중 임의의 하나 이상을 포함하거나 그와 인터페이싱할 수 있다. 뿐만 아니라, 통신은 또한 무선 응용 프로토콜(WAP: Wireless Application Protocol), GPRS(General Packet Radio Service), GSM(Global System for Mobile Communication), 코드 분할 다중 접근(CDMA: Code Division Multiple Access) 또는 시분할 다중 접근(TDMA: Time Division Multiple Access), 셀룰러 폰 네트워크, GPS(Global Positioning System), 셀룰러 디지털 패킷 데이터(CDPD: cellular digital packet data), RIM (Research in Motion, Limited) 듀플렉스 페이징 네트워크(duplex paging network), 블루투스 라디오(Bluetooth radio), 또는 IEEE 802.11 기반 무선 주파수 네트워크를 포함하는 임의의 다양한 무선 네트워크와의 링크를 포함할 수 있다. 네트워크(120)는 또한 RS-232 직렬 연결, IEEE-1394(파이어와이어(Firewire)) 연결, 파이버 채널 연결, IrDA(적외선) 포트, SCSI(Small Computer Systems Interface) 연결, USB (Universal Serial Bus) 연결 또는 다른 유선 또는 무선, 디지털 또는 아날로그 인터페이스 또는 연결, 메시(mesh) 또는 디지(Digi®) 네트워킹 중 임의의 하나 이상을 포함할 수도 있고 또는 이들과 인터페이싱할 수도 있다.

[0012] 시스템(105)은 일반적으로 프로세서(130), 네트워크 인터페이스(135), 및 메모리(140)를 포함한다. 몇몇 실시 예에 따라, 메모리(140)는 다양한 방법을 수행하기 위해 프로세서(130)에 의해 실행될 수 있는 로직(예컨대, 명령어)(145)를 포함한다. 예를 들어, 로직은 본 명세서에 더 상세하게 설명된 기능을 제공하도록 구성된 비디오 게임 클라이언트 애플리케이션(이하 "게임 애플리케이션(150)"이라 함)을 포함할 수 있다.

[0013] 시스템(105)에 기여하는 여기 서술된 기능들이 클라이언트(115) 내에서도 실행될 수 있음이 이해될 것이다. 즉, 클라이언트(115)는 여기 서술된 기능을 수행하도록 프로그래밍될 수 있다. 다른 경우에, 클라이언트(115)가 시스템(105) 및 클라이언트(115)가 클라이언트/서버 관계로 동작하도록 시스템(105)과 상호작용하는 클라이언트 측 애플리케이션을 포함함으로써, 시스템(105) 및 클라이언트(115)는 여기 서술된 기능을 제공하도록 협력할 수 있다. 복잡한 계산적 피처는 서버(105)에 의해 실행될 수 있고, 데이터 수집 및 데이터 디스플레이와 같

은 더 적은 계산 리소스를 필요로 하는 단순한 오퍼레이션은 클라이언트(115)에 의해 실행될 수 있다.

[0014] 일반적으로, 사용자 인터페이스 모듈(125)은 사용자가 시스템(105)과 상호작용하는 것을 가능하게 하는 다양한 그래픽 사용자 인터페이스(GUI)를 제공하기 위해 시스템(105)에 의해 실행될 수 있다. 몇몇 경우에, GUI는 애플리케이션(150)의 실행에 의해 생성된다. 사용자는, 예컨대, 클라이언트(115)를 이용하여 시스템(105)과 상호작용할 수 있다. 시스템(105)은 클라이언트를 위한 웹 기반 인터페이스를 생성할 수도 있다.

[0015] 게임 애플리케이션(150)(애플리케이션(150)이라고도 함)은 스마트폰 또는 태블릿 장치와 같은 사용자 장치(예컨대, 클라이언트(115)) 상에 로컬식으로 실행될 수 있다. 대안으로서, 애플리케이션(150)은 네트워크를 통해 사용자 장치에 의해 액세스될 수 있다. 그러므로, 애플리케이션(150)은 서버상에서 실행될 수 있고, 브라우저 애플리케이션을 이용하여 사용자 장치에 의해 액세스될 수 있다. 서버는 애플리케이션(150)의 GUI를 표준 또는 모바일 웹사이트의 웹페이지로서 역할하게 할 것이다.

[0016] 몇몇 실시예에서, 게임 애플리케이션(150)은 사용자가 게임플레이에 참여하는 가상 게임 환경과 같은 게임 경험을 사용자에게 제공하도록 구성되어 있다. 가상 게임 환경의 예는 롤플레잉 게임, 일인칭 슈팅 게임, 멀티플레이어 게임, 및 스포츠 게임 등을 포함하지만, 이에 제한되는 것은 아니다.

[0017] 게임 애플리케이션(150)은 가상 카드 배틀, 및 가상 카드 배틀의 결과를 적어도 일부 기초로 하는 가상 게임 활동, 이 둘 모두를 제공하도록 구성된다.

[0018] 아래의 도 2 내지 도 11은 가상 카드 배틀 및 가상 게임 활동 이둘 모두의 예시적인 스크린샷을 포함한다. 이러한 도면은 단지 예시일 뿐이며, 가상 카드 게임 기반의 가상 활동을 가능하게 하는 로직 및/또는 알고리즘들이 다른 가상 게임 경험으로 확장될 수 있음에 주목해야 한다.

[0019] 이제 도 2를 참조하면, 제1 플레이어는 제1 풋볼 팀에 대한 복수의 카드를 제공 받는다. 게임 애플리케이션(150)은, 몇 가지만 나열하자면, 파워, 시력, 인지력(awareness), 속도, 및 민첩도(agility)와 같은 하나 이상의 스킬을 랜덤하게 선택하기 위한 가상 게임을 구현하도록 구성된다. 이 스킬들은 제1 풋볼 팀에 대한 복수의 카드 위에 디스플레이된 제2 풋볼 팀의 플레이어에 대한 유사한 또는 유사하지 않은 스킬에 대항된다. 사용자는 수비수에 대하여 방어(depense), 공격수에 대하여 샷(shot)을 선택함으로써 그 플레이어가 공격 플레이어인지 또는 방어 플레이어인지 선택할 수 있다. 몇몇 실시예에서, GUI(200)는 플레이어가 샷과 방어 선택 간에 전환할 수 있는 선택 가능한 버튼(205)을 포함할 수 있다.

[0020] 더 상세하게는, 가상 카드(210) 테오 에이라스(Teo Eiras)는 파워 45, 시력 43, 인지력 46, 속도 38 및 민첩도 38과 같은 복수의 스킬 값을 가진다. 각각의 가상 카드는 자신의 스킬값 순열(permutation)을 가질 것이다. 이러한 스킬 값은 차이 값을 판정하기 위해 여기 서술된 비교 계산에 사용된다.

[0021] 도 3에서, 가상 카드는 플레이어에 의한 선택을 위해 하일라이팅된다. 플레이어는 몇몇 실시예에서 선택을 위해 카드 전체를 스크롤할 수 있다. 다른 실시예에서, 가상 카드는 애플리케이션(150)에 의해 랜덤하게 선택될 수 있다. 이러한 실시예에서, 플레이어는 제3의 페어링을 위한 그들의 가상 카드를 선택한다. 애플리케이션(150)에 의해 랜덤하게 선택되는 가상 카드는 플레이어가 그 가상 카드를 선택할 때까지 물음표를 단 채로 가려져 있다.

[0022] 2개의 카드 배틀 비교가 수행되었고, 차이 값이 계산되었다. 제1 페어링에서, 차이 값(215)은 -531이고, 제2 페어링에 대한 제2 차이값은 -235이다.

[0023] 도 4는 다수의 가상 카드 배틀 페어를 도시한다. 플레이어는 가상 카드 배틀을 실행하기 위해 킥 오프(kick off) 버튼(220)을 선택할 수 있다.

[0024] 사용자는 가상 카드 배틀을 이용하기 위해 카드를 선택할 수 있다(또는 가상 카드는 시스템에 의해 랜덤하게 선택될 수도 있다. 각 플레이어가 그들의 가상 카드를 선택하면, 팀이 도 4에 도시된 바와 같이 정렬된다. 도 5에서, 각각의 가상 카드 배틀의 결과가 플레이어가 도 2에서 방어 또는 샷을 선택하는 것을 가능하게 하는 선택 가능한 버튼을 대체하여 디스플레이될 수 있다. 플레이어는 그들이 가상 활동을 위한 근거로서 이용하기를 원하는 가상 카드 배틀 페어링을 선택할 수 있다.

[0025] 몇몇 실시예에서, 게임 애플리케이션(150)은 각각의 가능한 가상 카드 매치업을 위한 난이도 계산값을 디스플레이하도록 구성된다. 예를 들어, 제2 풋볼 팀의 제임스 멜바(James Melba)의 가상 카드(225)는 한스 디트리히(Hans Dietrich)의 가상 카드(230)에 대항하도록 매치업될 수 있고, 이는 -235의 점수를 받아 "매우 어려움"의 난이도 계산값(235)을 야기한다. 이러한 난이도 계산값은 제임스 멜바에 대한 인지력 160 및 속도 141의 선택

된 스킬 값과 한스 디트리히와 연관된 시력 27 및 인지력 39의 스킬 값을 비교한 차이이다.

[0026] 그러므로, 한스 디트리히가 제임스 멜바와의 가상 카드 매치에서 이길 가능성은 낮다.

[0027] 제1 사용자가 한스 디트리히에 대한 가상 카드를 선택한다면, 게임 애플리케이션(150)은 난이도 레벨에 대하여 가상 플레이팅 가드에 의해 표현된 각각의 플레이어에 대한 선택된 스킬 값을 이용하여 비교 분석을 수행하도록 구성된다. 난이도 등급(bracket)은 게임 활동에 대한 난이도 레벨과 연관된 차이 값의 범위를 포함한다. 예를 들어, "쉬움(easy)"의 난이도 레벨은 -40 내지 0의 차이 값 범위와 연관될 수 있다. 다른 예에서, 중간 난이도 레벨은 -75 내지 -41의 차이 값 범위와 연관될 수 있다. 다른 예에서, 이러한 범위는 원한다면 계층화(stratify)될 수 있다.

[0028] 고유한 난이도 레벨과 연관된 차이 값의 각각의 이러한 범위는 가상 활동에 대한 활동 속성의 선택에도 유사하게 연결된다. 예시적인 가상 활동은 도 10에 도시되어 있고 그와 연관지어 서술된다.

[0029] 도 7 및 도 8은 가상 카드 배틀 및 대응하는 가상 활동의 개시를 도시한다. 상세하게 말하자면, 도 7은 헤드 투 헤드 가상 카드 매치를 도시한다. 이 가상 카드 매치의 결과가 도 8에 도시되어 있는데, 예측한 바와 같이, 한스 디트리히는 제임스 멜바와의 배틀에서 235 포인트의 차이만큼 졌다. 이러한 실시예가 실제 카드 배틀보다 수행한 난이도 계산과 동등한 차이값을 보여주고 있으나, 사용된 확률 계산은 난이도 계산과 상이한 차이 값을 만들어낼 수도 있음이 이해될 것이다. 예를 들어, 확률을 기반으로, 차이 값을 기초로 했을 때 불리했던 파티가 그럼에도 불구하고 카드 배틀에서 이길수도 있고, 또는 -235보다 큰 더 작은 음의 값을으로 질 수도 있다. 이러한 차이 값의 크기는 가상 활동의 다양한 속성에 대한 난이도 레벨(240)을 선택하기 위해 사용된다.

[0030] 도 10은 가상 카드 배틀의 결과에 기인하여 선택된/영향을 받은 난이도(또는 복잡도) 레벨을 가지는 다양한 속성을 가지는 골 킥 활동과 같은 가상 활동(300)을 보여준다. 예를 들어, 가상 카드 게임의 결과의 음의 차이값으로 인해, 골 킥은 제1 속성에서 엄격한 각도로 놓여진 골대(305)를 포함한다. 부가적으로, 제2 속성은 키커(315) 부근의 3명의 수비수(310)의 위치를 포함한다. 제3 속성은 키커(315)와 골대(305) 사이의 거리를 포함하고, 또한 제4 속성은 풍속계(320)에 의해 지시되는 시간당 14 킬로미터의 역풍을 포함한다.

[0031] 몇몇 실시예에서, 골키퍼(325)의 스킬 또는 행동은 차이값 또는 그 차이값에 대하여 선택된 차이값 범위에 따라 시스템에 의해 선택적으로 조절될 수 있다.

[0032] 그러므로, 다른 속성은 더 공격적인 골키퍼, 또는 빗나간 골대의 관점(view)을 포함할 수도 있다. 다른 속성은 실질적으로 크기 감소된 타겟 영역, 또는 움직이거나 또는 불안정한 타겟 영역을 포함할 수 있다. 몇몇 실시예에서, 난이도 레벨은 상대 플레이어의 수, 환경적 조건, 타겟 영역 또는 크기, 조준 안정도, 타겟 거리, 및 이들의 다양한 조합과 같은 복수의 속성을 포함한다. 이러한 속성 각각은 플레이어에 대한 스킬 값을 특별하게 맞춤 조정될 수도 있다. 예를 들어, 플레이어가 비교적 낮은 시력 스킬 값을 가진다면, 가상 활동에 대한 속성은 골대가 흐리게 나타나게 하는 것을 포함할 수 있다.

[0033] 이와 반대로, 가상 카드 배틀 내의 차이 값이 0이면, 더 낮은 음의 속성이 선택되거나, 음의 몇몇 속성이 크기가 감소된다. 예를 들어, 차이가 -100이였다면, 단지 한명의 중간 수비수만 존재하고 바람이 시간당 5키로미터로 감소될 수 있다. 키커는 또한 골대와 다소 가깝게 위치할 수 있다. 차이 값이 양으로 전환되면, 골키퍼를 제거하고 키커를 골대 바로 앞으로 이동시키는 것과 같이, 속성이 키커에게 호의적으로 변할(skew) 수 있다.

[0034] 이러한 속성들을 맞춤 조절하기 위해 사용되는 알고리즘은 설계 요구사항에 따라 달라질 수 있지만, 임의의 활동은 수행된 가상 카드 배틀의 결과를 기초로 선택된 난이도 속성을 가진다. 몇몇 실시예에서, 가상 카드 배틀은 동일한 타입의 활동에 대한 배틀을 포함할 수 있다. 예를 들어, 시스템은 코너킥 액션과 관련된 2개의 가상 카드 간의 가상 카드 배틀을 수행할 수 있다. 카드 배틀에 이은 가상 활동은 가상 카드 배틀의 한 당사자를 포함하는 코너킥 활동이다.

[0035] 다른 예로서, 차이 값이 쉬움 난이도 레벨에 속한다면, 난이도 레벨은 (1) 플레이어가 골대 중앙과 나란하고; (2) 골대 전방에 느린 골키퍼만 존재하고; 및 (3) 역풍이 없는 가상 활동에 대한 활동 속성과 연관된다. 확실히, 이것은 본 발명의 구현 예일 뿐이고, 제한으로 의도된 것이 아니다.

[0036] 확실히, 난이도 레벨과 연관된 활동 속성의 크기 또는 개수는 난이도가 높아질수록 증가할 수 있다. 예를 들어, "어려움" 난이도 레벨에 대한 활동 속성의 개수가 5개이고, "중간" 난이도 레벨에 대한 활동 속성의 개수는 3개일 수 있다.

[0037] 동일한 원리는 차이 결과값이 음의 난이도 레벨(음의 차이) 또는 양의 난이도 레벨(양의 차이)을 야기하는지 여

부와 관계없이 적용된다.

[0038] 애플리케이션(150)은 활동에 대한 미리 정해진 속성을 이용하거나 가상 활동을 생성하기 위해 랜덤 또는 스킬 기반의 속성을 이용할 수 있다. 예를 들어, 수비수, 골키퍼, 바람, 및 거리에 대한 속성은 그 활동이 코너킥이므로 선택된다. 다른 가상 활동은 카드 게임 배틀 동안 선택된 스킬을 기초로 하여 상이한 속성의 선택을 야기 할 수 있다. 예를 들어, 플레이어가 낮은 시력 스킬 점수를 가지고, 프리킥에 관한 카드 배틀에 졌다면, 프리킥 가상 활동의 속성은 더 작은 골 타겟, 또는 흐리게 나타나는 골대를 포함할 수 있다.

[0039] 도 11은 본 발명의 예시적인 방법의 흐름도이다. 이 방법은 제1 플레이어에 대한 한 세트의 가상 카드 및 제2 플레이어에 대한 한 세트의 가상 카드를 선택하는 단계를 포함할 수 있다. 몇몇 실시예에서, 플레이어 중 한 명은 게임 시스템을 포함할 수 있다. 플레이어에 의해 선택된 카드는 카드당 한번씩 배틀(per-card-battle)하는 경우에 한번에 하나씩 선택될 수도 있고, 또는 대안으로서 플레이어는 시스템에 의한 페어와이즈 매칭(pairwise matching) 이전에 모든 그들의 가상 카드를 선택할 수 있다. 그러므로, 플레이어는 그들이 배틀하고자 하는 모든 가상 카드를 선택하고, 시스템은 플레이어에 의해 선택된 가상 카드와 짹을 이를 가상 카드를 랜덤 선택하게 만듦으로써 응답할 것이다.

[0040] 사용된 프로세스와 무관하게, 가상 카드 배틀은 제1 플레이어로부터 한 장 및 제2 플레이어로부터 한 장씩 한 쌍의 카드를 포함할 수 있다. 그러므로 이 방법은 가상 카드 배틀을 위한 한 쌍의 가상 카드 중 적어도 하나의 선택을 수신하는 단계(1105)를 포함할 수 있다. 다시, 몇몇 실시예는 이 시스템이 플레이어들이 모두 휴면 플레이어일 때 카드 쌍의 두 선택을 모두 수신하는 것을 허용한다. 시스템이 하나의 플레이어로부터 가상 카드의 선택을 수신한 다음 시스템이 그 쌍을 위한 다른 카드를 뽑는 경우에, 이 시스템은 가상 카드 쌍 중 하나의 카드만 수신한다.

[0041] 카드가 선택된 후, 이 방법은 가상 카드 쌍 각각에 대한 하나 이상의 스킬을 선택하는 단계(1110)를 포함한다. 앞서 제공된 예에서, 이 시스템은 민첩도, 시력, 또는 카드와 연관된 다른 스킬과 같은 스킬을 랜덤하게 선택할 수 있다. 또한, 선택된 카드에 대한 스킬들은 그 카드가 속하는 주제(subject matter)에 따라 다양할 수 있다. 예를 들어, 풋볼 선수를 나타내는 가상 카드는 야구 선수를 나타내는 가상 카드와 상이한 스킬을 가질 수 있다.

[0042] 그 다음, 이 방법은 차이 값은 계산하기 위해 하나 이상의 스킬을 이용하여 비교 분석을 수행함으로써 가상 카드 배틀을 실행하는 단계(1115)를 포함한다. 차이 값이 계산된 후, 이 방법은 시스템이 가상 활동에 대한 복수의 속성을 선택하기 위해 그 차이 값을 이용하는 단계(1120)를 포함할 수 있다. 앞서 언급한 바와 같이, 가상 활동은 가상 카드 배틀의 상호 작용 버전이다.

[0043] 유리하게도, 이 시스템은 가상 카드 배틀의 결과가 가상 활동 또는 게임 경험을 생성하기 위해 사용되는 로직을 선택하는 것을 허용함으로써, 가상 게임의 범위(confine) 내에 새로운 매력적인 가상 활동을 생성할 수 있다. 각각의 가상 활동은 카드 배틀 결과를 기초로 선택된 그것의 속성들을 가지는 고유한 게임 인스턴스(instance)가 될 잠재성을 가진다.

[0044] 더 상세하게는, 이 시스템은 차이 값이 증가함에 따라 가상 활동의 난이도 또는 복잡도를 선택적으로 변경할 수 있다. 예를 들어, 차이 값이 증가함에 따라 가상 활동의 속성은 그 개수가 증가하거나 난이도가 증가할 수 있다. 예시의 방법으로, 차이값이 -100일 때 골대가 30미터 거리에 위치한다면, 차이 값이 -200일 때 골대는 50미터에 위치할 수 있다. 대안으로서 또는 부가적으로, 차이 값이 -100에서 -200으로 변한다면, 골키퍼는 그 크기가 증가하거나 시도된 골을 막는 민첩도가 증가할 수 있다.

[0045] 시스템이 차이 값을 이용하여 가상 활동의 초기 조건을 결정한 후, 이 방법은 가상 카드 배틀의 플레이어에게 복수의 속성을 가진 가상 활동을 제공하는 단계(1125)를 포함할 수 있다. 가상 활동의 예시적인 프로비전(provision)은 도 10에 도시되어 있다.

[0046] 도 12는 앞서 언급한 프로세스에서 난이도 값 범위를 이용하는 방법을 도시한다. 맥락에 따라, 한 쌍의 가상 카드들이 상술된 임의의 기술을 이용하여 가상 카드 배틀을 위해 선택되었다고 가정한다. 또한, 이 방법은 카드 배틀의 주제(subject matter)의 시스템에 의한 랜덤한 선택 또는 플레이어에 의한 선택 중 어느 하나를 포함할 수 있다. 앞서 언급한 바와 같이, 이 주제는 방어 또는 샷을 포함할 수 있다. 방어 구성에서, 가상 활동은 키커에 맞서는 골키퍼 또는 수비수로서 가상 활동에 참가하는 플레이어를 포함할 것이다.

[0047] 이 방법은 한 쌍의 가상 카드 중 각각의 카드에 대한 하나 이상의 스킬을 선택하는 단계(1205)를 포함한다. 다시, 하나 이상의 스킬 각각은 그것과 연관된 스킬값을 가진다.

- [0048] 그 다음, 이 방법은 차이 값을 수립하기 위해 가상 카드 쌍의 각각의 카드에 대한 하나 이상의 스킬을 이용하여 그 가상 카드 쌍의 비교 분석을 포함하는, 가상 카드 쌍을 이용한 가상 카드 배틀을 수행하는 단계(1210)를 포함할 수 있다.
- [0049] 차이 값이 계산된 후, 이 방법은 차이 값을 차이 값 범위와 비교함으로써 가상 카드 배틀에 대한 난이도 레벨을 판정하는 단계(1215)를 포함한다. 다시, 차이 값 범위는 복수의 속성을 정의하는 다양한 난이도 레벨과 연관된다. 예를 들어, -100 내지 0의 난이도 레벨 범위에 속하는 차이 값은 "어려움"으로 라벨링될 수 있다. 난이도 값 범위는 가상 활동에 대한 사전 정의된 세트의 속성과 연관될 수 있다. 이러한 사전 정의는 시스템이 차이 값의 크기를 기초로 가상 활동에 대한 속성들을 지능적으로(intelligently) 선택할 수 있는 실시예와 달리, 시스템을 보다 정적인 방식으로 동작하게 만든다. 그러므로, 몇몇 실시예에서, "어려움" 차이 값 범위에 속하는 차이 값을 가지는 모든 가상 카드 배틀은 속성의 동적 선택과 비교하여, 동일한 세트의 가상 활동 속성을 가지는 가상 활동을 야기할 것이다.
- [0050] 이 방법은 또한 선택된 차이 값 범위와 연관된 복수의 속성을 이용하여 가상 활동을 생성하는 단계(1220)를 포함할 수 있다.
- [0051] 몇몇 실시예에서, 본 발명의 시스템은 불연속 확률 밀도 계산, (난이도 레벨과 연관된 차이 값 범위를 포함하는 표를 이용하는) 차이 값 루프 함수, 및 가상 카드 배틀과 연관된 가상 활동의 속성들을 다이내믹하게 선택하기 위한 난수 스크립팅(random number scripting)을 이용할 수 있다. 예컨대, "어려움"의 난이도 레벨을 선택하도록 차이 값이 이용되었다면, 코너킥과 같은, 가상 활동의 파라미터들은 수비수의 수, 킥의 각도, 골대로부터의 거리, 및 풍속 등을 포함할 수 있다. 이 시스템은 각각의 이러한 파라미터에 대한 값을 선택하는 난수 발생기 기능을 구현할 수 있다. 0에서 10까지의 눈금을 이용하여, 시스템에 의해 더 높은 난수가 발생될수록, 주어진 파라미터는 더 어려워진다.
- [0052] 몇몇 실시예에서, 이 방법은 (a) 제1 파티의 가상 카드 선택을 수신하는 단계; (b) 제1 파티의 가상 카드 선택을 수신하는 단계; (c) 가상 카드 배틀을 위한 한 쌍의 가상 카드를 선택하는 단계로서, 이 한 쌍의 가상 카드는 제1 파티의 가상 카드 중 하나 및 제2 파티의 가상 카드 중 하나를 포함하는 것인 상기 가상 카드 배틀을 위한 한 쌍의 가상 카드를 선택하는 단계; (d) 가상 카드 쌍 중 각각의 카드에 대한 하나 이상의 스킬을 선택하는 단계; (e) 차이 값을 수립하기 위해 가상 카드 쌍 중 각각의 카드에 대한 하나 이상의 스킬을 이용하여 가상 카드 쌍의 비교 분석을 수행하는 단계; (f) 차이값의 범위를 포함하는 테이블의 검색(lookup)을 실행하는 단계로서, 각각의 범위는 가상 카드 배틀을 기초로 하는 가상 활동에 대한 난이도 레벨과 연관되어 있고, 상기 검색은 계산된 차이 값이 어떤 차이 값 범위에 속하는지 판정하는 것인, 상기 차이값의 범위를 포함하는 테이블의 검색(lookup)을 실행하는 단계; (g) 플레이어가 가상 활동에 참여하는 방법을 정의하는 복수의 선택된 가상 활동 파라미터 각각에 대하여 불연속 확률 밀도 계산을 수행하는 단계; (h) 복수의 선택된 가상 활동 파라미터 각각에 대한 값을 다이내믹하게 선택하기 위한 난수 스크립팅을 수행하는 단계; 및 (i) 플레이어에 대한 가상 활동을 실행하는 단계를 더 포함한다. 카드 배틀을 위한 난이도 값 범위를 판정하기 위해 사용되는 차이 값의 계산, 및 가상 활동 파라미터의 선택을 보여주는 의사 코드(pseudo code)의 예시적인 실시예가 아래에 제공된다. 이 시스템은 하나의 실시예에서 파이썬 스크립트(Python script)를 이용할 수 있다. 예시적인 파이썬 스크립트는 난이도 값 범위를 포함하는 테이블로부터 값을 검색하기 위한 매치 플로우(match flow)를 보여준다.
- ```

[0053] import json
[0054] import random
[0056] 이 스크립트의 첫 부분은 불연속 확률 밀도를 기초로 하여 값의 랜덤 리스트를 생성할 것이다.
[0057] def DiscretePDF( dist, count=None ):
[0058]     returnlist = True
[0059]     if count == None:
[0060]         returnlist = False
[0061]     count = 1
[0063]     total = sum( dist.values( ) )

```

```

[0064]     result = [ ]
[0065]     i = 0
[0066]     while i < count:
[0067]         val = random.uniform( 0, total )
[0068]         cur = 0
[0069]         for k, v in dist.items( ):
[0070]             cur += v
[0071]             if val <= cur:
[0072]                 result.append( k )
[0073]                 break
[0074]         i += 1
[0075]     if returnlist:
[0076]         return result
[0077]     else:
[0078]         return result[ 0 ]
[0080] def ResolveCardBattle(cardValuePlayer, cardValueOpponent):
[0081]     return cardValuePlayer - cardValueOpponent
[0083] def SelectDifficulty(cardBattleDelta):
[0084]     difficulty = None
[0085]     for difficulty_key, delta_range in tuning[ "Data" ][ "DifficultyLevelRanges" ].items():
[0086]         if cardBattleDelta >= delta_range[0] and cardBattleDelta < delta_range[1]:
[0087]             print "Selected", difficulty_key
[0088]             difficulty = difficulty_key
[0089]         break
[0091]     return difficulty
[0093] 아래의 스크립트 부분은, 이 실시예에서 패스 활동을 포함하는, 활동의 클래스의 선택을 제공한다.
[0094] class Scenario:
[0095]     pass
[0097]     def play_match():
[0098]         card_slots = [
[0099]             (194,35),
[0100]             (100,350),
[0101]             (300,350),
[0102]             (954,423),
[0103]             (234,290)
[0104]         ]

```

[0105]

[0106] 가상 카드 배틀로부터의 델타 값은 아래에서 생성된다.

[0108]

for card\_slot in card\_slots:

[0109]

delta = ResolveCardBattle(\*card\_slot)

[0110]

difficulty = SelectDifficulty(delta)

[0111]

[0112]     scenario = Scenario()

[0113]

scenario.defenders =

[0114]

DiscretePDF(tuning["Data"]["KickingGames"]["DefenderDifficultyTuning"][difficulty])

[0115]

scenario.wind =

[0116]

random.uniform(\*tuning["Data"]["KickingGames"]["WindStrengthRange"][difficulty])

[0117]

scenario.shot\_angle =

[0118]

DiscretePDF(tuning["Data"]["KickingGames"]["ShotAngle"][difficulty])

[0119]

scenario.shot\_distance =

[0120]

DiscretePDF(tuning["Data"]["KickingGames"]["ShotDistance"][difficulty])

[0122]

print "Player Card Value", card\_slot[0]

[0123]

print "Opponent Card Value", card\_slot[1]

[0124]

print "The following are the basic scenario parameters for a Shooting/Passing Gameplay scenario,"

[0125]

print "Based on the above card values"

[0126]

print "These values would feed into the game runtime engine to generate a Shooting/Passing attempt visualization."

[0127]

print "-----"

[0128]

print json.dumps(scenario.\_\_dict\_\_, indent=4)

[0129]

print "-----"

[0131]

아래의 스크립트 부분은 상호작용 게임에 대한 튜닝 파라미터 선택을 제공하는데, 여기서 앞서 선택된 난이도 값을 이용하여(또한 값 테이블의 난이도 값 범위를 이용하여) 뿐만 아니라 각각의 활동 파라미터에 대한 값을 선택하기 위해 난수 발생을 이용하여 속성을 선택한다.

[0133]

tuning = json.loads("""{

[0134]

"Type": "TuningData",

[0135]

"Description": "Tunables used for the interactivity games",

[0136]

"Data":

[0137]

{

[0138]

"FieldMaxX": 33,

[0139]

"FieldMaxZ": 52,

[0140]

"AttemptTimeOutTime": 2,

[0141]

"CardBattleWinBonus": 0,

```

[0142] "DifficultyLevelRanges":  

[0143] {  

[0144]     "VERY_EASY": [175, 1000000],  

[0145]     "EASY": [33, 174],  

[0146]     "MEDIUM": [0, 32],  

[0147]     "HARD": [-224, -1],  

[0148]     "VERY_HARD": [-1000000, -225]  

[0149] },  

[0150]  

[0151] "KickingGames":  

[0152] {  

[0153]     "ChanceToWinOnGoodAction": [0.9, 0.7, 0.4, 0.2, 0.05],  

[0154]     "TargetColors":  

[0155]     {  

[0156]         "BIG": [0.0, 1.0, 0.0],  

[0157]         "MEDIUM": [1.0, 1.0, 0.0],  

[0158]         "SMALL": [1.0, 0.0, 0.0]  

[0159]     },  

[0160]     "TargetColorRanges":  

[0161]     {  

[0162]         "SMALL": [0.0, 0.40],  

[0163]         "MEDIUM": [0.40, 0.75],  

[0164]         "BIG": [0.75, 1.0]  

[0165]     },  

[0166]     "NickingTargetCountsAsHit": false,  

[0167]     "TimerTotalTime":  

[0168]     [  

[0169]         3.0,  

[0170]         3.0,  

[0171]         3.0,  

[0172]         3.0,  

[0173]         3.0  

[0174]     ],  

[0175]     "TimerFlashTime": 0.75,  

[0176]     "TimeToSlowTime": 0.85,  

[0177]     "LowHeightKick": 0.1,

```

```

[0178]     "HighHeightKick": 3.3,
[0179]     "MaxCurveForce": 100000,
[0180]     "BallPastTargetCutOff": 7.5,
[0181]     "MinSwipeSpeedCrossThreshold" : 0.005,
[0183]     "DefenderSpacing": 1.0,
[0184]     "DefenderWallDist": 8.0,
[0185]
[0186]     "DefenderToMarkSpacingMin": 1.75,
[0187]     "DefenderToMarkSpacingMax": 3,
[0188]     "ExtrasMinDistSide": 3.0,
[0189]     "ExtrasMaxDistSide": 6.0,
[0190]     "ExtrasMinDistBall": 0.4,
[0191]     "ExtrasMaxDistBall": 0.7,
[0192]
[0193]     "KickerOffset":
[0194]     [
[0195]         -0.5,
[0196]         0.0,
[0197]         -5.5
[0198]     ],
[0199]     "CameraOffset":
[0200]     [
[0201]         0,
[0202]         2.6,
[0203]         -9.5
[0204]     ],
[0205]     "CameraLookAtOffset":
[0206]     [
[0207]         0,
[0208]         1.5,
[0209]         0
[0210]     ],
[0211]     "PreCameraBlendTime": 1.0,
[0212]     "PreCameraOffset":
[0213]     [
[0214]         0,

```

```

[0215]        4,
[0216]        -20
[0217]    ],
[0218]    "TargetScales": [ 1.1, 0.95, 0.8, 0.7, 0.6 ],
[0219]    "TargetVanishChance": [ 0.0, 0.0, 0, 0, 0 ],
[0220]    "TargetWhenToVanish": 3.25,
[0221]
[0222]    "WindStrengthRange":
[0223]    {
[0224]        "VERY_EASY": [0, 0],
[0225]        "EASY": [2, 5],
[0226]        "MEDIUM": [5, 10],
[0227]        "HARD": [10, 15],
[0228]        "VERY_HARD": [10, 20]
[0229]    },
[0230]
[0231]    "WindChance": [ 0.0, 0.2, 0.5, 0.85, 1.0 ],
[0232]
[0233]    "DefendingScenarios":
[0234]    {
[0235]        "none" : [],
[0236]        "1left" : [ {"x": 1.25, "y": 3.0} ],
[0237]        "1right" : [ {"x": -1.25, "y": 3.0} ],
[0238]        "1closeleft" : [ {"x": 0.4, "y": 1.5} ],
[0239]        "1closeright" : [ {"x": -0.4, "y": 1.5} ],
[0240]        "1centre": [ {"x": 0.0, "y": 1.0} ],
[0241]        "2left": [ {"x": -1.0, "y": 1.0}, {"x": 0.5, "y": 1.0} ],
[0242]        "2farright": [ {"x": -0.90, "y": 0.5}, {"x": 1.10, "y": 3.0} ],
[0243]        "2farleft": [ {"x": -1.10, "y": 3.0}, {"x": 0.90, "y": 0.5} ],
[0244]        "2right": [ {"x": -0.5, "y": 1.0}, {"x": 1.0, "y": 1.0} ],
[0245]        "2closeleft" : [ {"x": -1.45, "y": 4.0}, {"x": 0.25, "y": 0} ],
[0246]        "2closeright" : [ {"x": -0.25, "y": 0}, {"x": 1.45, "y": 4.0} ],
[0247]        "2centreA" : [ {"x": -1.4, "y": 0.5}, {"x": 0.0, "y": 0.0} ],
[0248]        "2centreB" : [ {"x": 0.0, "y": 0.0}, {"x": 1.4, "y": 0.5} ],
[0249]        "2block" : [ {"x": -0.4, "y": 0.75}, {"x": 0.4, "y": 3.75} ],
[0250]        "3left": [ {"x": -0.8, "y": 1.25}, {"x": 0.25, "y": 0.5}, {"x": 2.0, "y": 3.5} ],

```

```

[0251]     "3right": [{"x": -2.0, "y": 3.5}, {"x": -0.25, "y": 0.5}, {"x": 0.8, "y": 1.25}],
[0252]     "3closeleft" : [{"x": 2.5, "y": 5.0}, {"x": 0.0, "y":0.0}, {"x": -1.0, "y": 2.5}],
[0253]     "3closeright" : [{"x": 1.0, "y": 2.5}, {"x": 0, "y": 0.0}, {"x": -2.5, "y": 5.0}],
[0254]     "3centre" : [{"x": -1.2, "y": 3.0}, {"x": 0.0, "y": 5.0}, {"x": 1.2, "y": 2.0}]
[0255] },
[0256]
[0257]     "DefenderDifficultyTuning" :
[0258] {
[0259]     "VERY_EASY" :
[0260] {
[0261]         "none": 50,
[0262]         "1left": 20,
[0263]         "1right": 20,
[0264]         "1closeleft": 5,
[0265]         "1closeright": 5
[0266] },
[0267]     "EASY" :
[0268] {
[0269]         "none": 10,
[0270]         "1left": 30,
[0271]         "1right": 30,
[0272]         "1closeleft": 10,
[0273]         "1closeright": 10,
[0274]         "1centre": 10,
[0275]         "2left": 0,
[0276]         "2right": 0,
[0277]         "2farright" : 0,
[0278]         "2farleft" : 0,
[0279]         "2closeleft" : 0,
[0280]         "2closeright" : 0,
[0281]         "2centreA" : 0,
[0282]         "2centreB" : 0
[0283] },
[0284]     "MEDIUM" :
[0285] {
[0286]         "none": 0,

```

```

[0287]         "1left": 0,
[0288]         "1right" : 0,
[0289]         "1closeleft" : 10,
[0290]         "1closeright" : 10,
[0291]         "1centre": 10,
[0292]         "2left": 20,
[0293]         "2right": 20,
[0294]         "2closeleft" : 20,
[0295]         "2closeright" : 20,
[0296]         "2farright" : 0,
[0297]         "2farleft" : 0,
[0298]         "2centreA" : 15,
[0299]         "2centreB" : 15,
[0300]         "2block" : 15,
[0301]         "3left": 0,
[0302]         "3right": 0,
[0303]         "3closeleft" : 0,
[0304]         "3closeright" : 0,
[0305]         "3centre" : 0
[0306]     },
[0307]     "HARD" :
[0308]     {
[0309]         "none": 0,
[0310]         "1left": 5,
[0311]         "1right" : 5,
[0312]         "1closeleft" : 5,
[0313]         "1closeright" : 5,
[0314]         "1centre": 5,
[0315]         "2left": 10,
[0316]         "2right": 10,
[0317]         "2closeleft" : 10,
[0318]         "2closeright" : 10,
[0319]         "2farright" : 0,
[0320]         "2farleft" : 0,
[0321]         "2centreA" : 10,
[0322]         "2centreB" : 10,

```

```

[0323]         "2block" : 10,
[0324]         "3left": 0,
[0325]         "3right": 0,
[0326]         "3closeleft" : 0,
[0327]         "3closeright" : 0,
[0328]         "3centre" : 0
[0329]     },
[0330]     "VERY_HARD" :
[0331]     {
[0332]         "none": 0,
[0333]         "1left": 0,
[0334]         "1right" : 0,
[0335]         "1closeleft" : 0,
[0336]         "1closeright" : 0,
[0337]         "1centre": 0,
[0338]         "2left": 0,
[0339]         "2right": 0,
[0340]         "2closeleft" : 0,
[0341]         "2closeright" : 0,
[0342]         "2centreA" : 5,
[0343]         "2centreB" : 5,
[0344]         "2block" : 5,
[0345]         "3left": 15,
[0346]         "3right": 15,
[0347]         "3closeleft" : 20,
[0348]         "3closeright" : 20,
[0349]         "3centre" : 15
[0350]     }
[0351] },
[0352] "ShotAngle" :
[0353] {
[0354]     "VERY_EASY" :
[0355]     {
[0356]         "WideLeft": 0,
[0357]         "NearLeft": 10,
[0358]         "NoAngle": 80,

```

```

[0359]         "NearRight": 10,
[0360]         "WideRight": 0
[0361]     },
[0362]     "EASY":
[0363]     {
[0364]         "WideLeft": 0,
[0365]         "NearLeft": 10,
[0366]         "NoAngle": 80,
[0367]         "NearRight": 10,
[0368]         "WideRight": 0
[0369]     },
[0370]     "MEDIUM":
[0371]     {
[0372]         "WideLeft": 10,
[0373]         "NearLeft": 25,
[0374]         "NoAngle": 30,
[0375]         "NearRight": 25,
[0376]         "WideRight": 10
[0377]     },
[0378]     "HARD":
[0379]     {
[0380]         "WideLeft": 20,
[0381]         "NearLeft": 20,
[0382]         "NoAngle": 20,
[0383]         "NearRight": 20,
[0384]         "WideRight": 20
[0385]     },
[0386]     "VERY_HARD":
[0387]     {
[0388]         "WideLeft": 25,
[0389]         "NearLeft": 15,
[0390]         "NoAngle": 10,
[0391]         "NearRight": 15,
[0392]         "WideRight": 25
[0393]     }
[0394] }

```

```

[0395]     "ShotDistance":  

[0396]     {  

[0397]         "VERY_EASY":  

[0398]         {  

[0399]             "Near": 100,  

[0400]             "Mid": 0,  

[0401]             "Far": 0  

[0402]         },  

[0403]         "EASY":  

[0404]         {  

[0405]             "Near": 50,  

[0406]             "Mid": 50,  

[0407]             "Far": 0  

[0408]         },  

[0409]         "MEDIUM":  

[0410]         {  

[0411]             "Near": 0,  

[0412]             "Mid": 50,  

[0413]             "Far": 50  

[0414]         },  

[0415]         "HARD":  

[0416]         {  

[0417]             "Near": 0,  

[0418]             "Mid": 0,  

[0419]             "Far": 100  

[0420]         },  

[0421]         "VERY_HARD":  

[0422]         {  

[0423]             "Near": 0,  

[0424]             "Mid": 0,  

[0425]             "Far": 100  

[0426]         }  

[0427]     }  

[0428] },  

[0429] "ShootingGame":  

[0430] {

```

```

[0431]         "ShotAngleValues" :
[0432]         {
[0433]             "WideLeft": -40,
[0434]             "NearLeft": -20,
[0435]             "NoAngle": 0,
[0436]             "NearRight": 20,
[0437]             "WideRight": 40
[0438]         },
[0439]         "ShotDistanceValues" :
[0440]         {
[0441]             "Near": 20,
[0442]             "Mid": 30,
[0443]             "Far": 42
[0444]         },
[0445]         "ShotSpeeds" :
[0446]         {
[0447]             "Near": 25,
[0448]             "Mid": 40,
[0449]             "Far": 45
[0450]         }
[0451]     }
[0452] }
[0453] }
[0454] """
[0455] play_match()

[0456] 가상 활동의 속성을 확률적 방식으로 판정하기 위해 스크립트가 실행된 후, 가상 활동은 플레이 매치 스크립트 호출(call)에 의해 실행된다.

[0457] 아래의 부분은 가상 카드 배틀의 실행을 통해 선택된 가상 활동을 제공하기 위해 사용되는 예시적인 게임 스크립트를 서술한다.

[0458] 이 스크립트는 다양한 샷(shot) 거리로부터 선택하기 위해 사용된다.

[0459] public enum ShotDistances

[0460] {
[0461]     Near,
[0462]     Mid,
[0463]     Far,
[0464]     Invalid

```

```

[0466]    };
[0467]    이 스크립트는 다양한 샷 각도로부터 선택하기 위해 사용된다.
[0468]    public enum ShotAngles
[0469]    {
[0470]        WideLeft,
[0471]        NearLeft,
[0472]        NoAngle,
[0473]        NearRight,
[0474]        WideRight,
[0475]        Invalid
[0476]    };
[0477]    이 스크립트는 다양한 공 속도로부터 선택하기 위해 사용된다.
[0478]    public enum BallSpeed
[0479]    {
[0480]        Slow,
[0481]        Medium,
[0482]        Fast,
[0483]        Invalid
[0484]    };
[0485]    전체적인 활동 난이도 레벨이 계산되고 디스플레이된다.
[0486]    //for difficulty meter
[0487]    public enum DifficultyLevel
[0488]    {
[0489]        VERY_EASY = -2,
[0490]        EASY,
[0491]        MEDIUM,
[0492]        HARD,
[0493]        VERY_HARD,
[0494]        Count = 5,
[0495]        Invalid = -100
[0496]    };
[0497] }
[0498] 이 스크립트는 또한 이 예에서 키킹(kicking) 게임을 포함하는 활동 스크립트의 클래스를 선택하기 위해 사용된다.
[0499] public class KickingGameScript
[0500] {
[0501]     //-----

```

```

[0503]      //
[0504]      //
[0505]      //
[0506]      //-----
[0507]      public static DifficultyLevel GetDifficultyLevel(int playerDifference, DifficultyLevel
diffLevel = DifficultyLevel.Count)
[0508]      {
[0509]          if (diffLevel == DifficultyLevel.Count)
[0510]          {
[0511]              if (Interactivity.Util.Tuning == null)
[0512]              {
[0513]                  Interactivity.Util.RefreshTuning();
[0514]              }
[0516]              foreach (KeyValuePair<DifficultyLevel, List<int>> pair in
Interactivity.Util.Tuning.DifficultyLevelRanges)
[0517]              {
[0518]                  List<int> range = pair.Value;
[0519]                  if (range[0] <= playerDifference && playerDifference <= range[1])
[0520]                  {
[0521]                      for (int i = (int)DifficultyLevel.VERY_EASY; i <=
(int)DifficultyLevel.VERY_HARD; i++)
[0522]                      {
[0523]                          if (pair.Key == ((DifficultyLevel)i))
[0524]                          {
[0525]                              diffLevel = pair.Key;
[0526]                          }
[0527]                      }
[0528]                  }
[0529]              }
[0530]          }
[0531]          return diffLevel;
[0532]      }
[0533]      public int GetCurrentPlayerDifference()
[0534]      {
[0535]          return user.player[current_slot_index] - opponent.player[current_slot_index];
[0536]      }

```

```

[0537]     public DifficultyLevel GetDifficultyLevel()
[0538]     {
[0539]         return MatchFlowUtil.GetDifficultyLevel(GetCurrentPlayerDifference(),
tut_GetDifficultyLevelOverride());
[0540]     }
[0541]     protected ShotDistances GetShotDistance()
[0542]     {
[0543]         if (eCurrentShotDistance == ShotDistances.
[0544]             Invalid)
[0545]         {
[0546]             eCurrentShotDistance = random.DiscretePDF(Util.Tuning.
[0547] KickingGames.ShotDistance[GetDifficultyLevel()][0];
[0548]             if (tut_GetOverrideShotDistance() != ShotDistances.
[0549]             Invalid)
[0550]             {
[0551]                 eCurrentShotDistance = tut_GetOverrideShotDistance();
[0552]             }
[0553]             if (eOverrideShotDistance != ShotDistances.Invalid)
[0554]             {
[0555]                 eCurrentShotDistance = eOverrideShotDistance;
[0556]             }
[0557]         }
[0558]     }
[0559]     return eCurrentShotDistance;
[0560] }
[0561] }
[0562]     protected ShotAngles GetShotAngle()
[0563]     {
[0564]         if (eCurrentShotAngle == ShotAngles.Invalid)
[0565]         {
[0566]             {
[0567]                 eCurrentShotAngle = random.DiscretePDF(Util.Tuning.KickingGames.ShotAngle[GetDifficultyLevel()][0];
[0568]                 if (tut_GetOverrideShotAngle() != ShotAngles.Invalid)
[0569]                 {
[0570]                     eCurrentShotAngle = tut_GetOverrideShotAngle();
[0571]                 }
[0572]                 if (eOverrideShotAngle != ShotAngles.Invalid)
[0573]                 {
[0574]                     {

```

```

[0575]             eCurrentShotAngle = eOverrideShotAngle;
[0576]         }
[0577]     }
[0578]     return eCurrentShotAngle;
[0579] }
[0580]     override protected float GetShotDistanceValue()
[0581] {
[0582]     return Util.Tuning.ShootingGame.ShotDistanceValues[GetShotDistance()];
[0583] }
[0584]     override protected float GetShotAngleValue()
[0585] {
[0586]     return Util.Tuning.ShootingGame.ShotAngleValues[GetShotAngle()];
[0587] }
[0588]     override protected float GetShotSpeed()
[0589] {
[0590]     return Util.Tuning.ShootingGame.ShotSpeeds[eCurrentShotDistance];
[0591] }
[0592] // This is where we take the values
[0593]     protected override void SetupNewAttempt()
[0594] {
[0595]     ResetCharacterInstances();
[0596]     EnableAllInteractiveCharacters(false);
[0597]     PopulateBackgroundCharacters(playerInitData, nonKeeperSlots, teamPlayerInstances,
[0598]     mInputData.SelectedPlayers);
[0599]     PopulateBackgroundCharacters(opponentInitData, nonKeeperSlots, defenderInstances,
[0600]     mInputData.SelectedOpponents);
[0601]     CharacterInitData playerKeeperInitData = new CharacterInitData();
[0602]     playerKeeperInitData.teamSlot = TeamSlotType.K;
[0603]     playerKeeperInitData.playerGO = PlayerGoalie;
[0604]     playerInitData.Add(playerKeeperInitData);
[0605]     ReplaceCharacterModels(Kicker, OpponentGoalie);
[0606]     //Distance
[0607]     eCurrentShotDistance = GetShotDistance();
[0608]     float fDistance = GetShotDistanceValue();
[0609]     fShotAngle = GetShotAngleValue();
[0610] }

```

```

[0614]         float x = fDistance * Mathf.Sin(Mathf.Deg2Rad * fShotAngle);
[0615]         float z = fDistance * Mathf.Cos(Mathf.Deg2Rad * fShotAngle);
[0617]         v3MomentOrigin = v3KickToPos + new Vector3(x, BallPrefab.
[0618]         transform.lossyScale.x / 2.0f, -z);
[0620]         //Kicker
[0621]         Vector3 kickerPosition = new Vector3(v3MomentOrigin.x, 0.
[0622]         of, v3MomentOrigin.z);
[0623]         Kicker.transform.position = kickerPosition;
[0624]         Kicker.transform.LookAt(v3KickToPos);
[0626]         //Rotate the kicker's offset too.
[0628]         Vector3 kickerOffset = new Vector3(Util.Tuning.KickingGames.KickerOffset[0],
[0629]         Util.Tuning.KickingGames.KickerOffset[1], Util.Tuning.KickingGames.KickerOffset[2]);
[0630]         Vector3 kickerRotatedOffset = Kicker.transform.rotation * kickerOffset;
[0631]         Vector3 correctedKickerPosition = kickerPosition + kickerRotatedOffset;
[0632]         Kicker.transform.position = correctedKickerPosition;
[0633]         // create the ball at the kicker, so it doesn't move around when attached
[0634]         Quaternion identityRotation = Quaternion.identity;
[0635]         Instantiate(BallPrefab, correctedKickerPosition, identityRotation);
[0637]         GameObject ball = Interactivity.Ball.GetInstance();
[0638]         Debug.DrawLine(ball.transform.position, ball.transform.
[0639]         position + Vector3.up * 2.0f, Color.red, 10.0f);
[0641]         ball.transform.parent = tempObjects;
[0642]         ball.GetComponent<Ball>().TrajectoryPredictedEvent += new
[0643]         TrajectoryPredictedEventHandler(TrajectoryPredicted);
[0644]         //Set the kick speed.
[0645]         float fKickSpeed = GetShotSpeed();
[0646]         BallKicker2 ballKickScript = Kicker.GetComponent<BallKicker2>();
[0647]         if (ballKickScript != null)
[0648]         {
[0649]             ballKickScript.
[0650]             SetKickSpeed(fKickSpeed);
[0651]         }
[0652]         //////////////////////////////
[0654]         // Goalie
[0655]         Vector3 netToBall = v3MomentOrigin - v3KickToPos;
[0656]         Vector3 v3GoaliePosition = v3KickToPos + netToBall.normalized * GoalieDistOffGoalLine;

```

```

[0657]         OpponentGoalie.transform.position = v3GoaliePosition;
[0658]         OpponentGoalie.transform.LookAt(new Vector3(v3MomentOrigin.
[0659]             x, 0.0f, v3MomentOrigin.z));
[0660]         if (OpponentGoalie != null)
[0661]         {
[0662]             AIKickInterceptorScript goalieScript = OpponentGoalie.
[0663]             GetComponent<AIKickInterceptorScript>();
[0664]             if (goalieScript != null)
[0665]             {
[0666]                 goalieScript.Reset(v3GoaliePosition, v3MomentOrigin);
[0667]                 goalieScript.TouchBallEvent += GoalieTouchedBall;
[0668]             }
[0669]         }
[0670]     }
[0671]     base.SetupNewAttempt();
[0672]     mCameraBlender.StartBlend();
[0673] }
[0674] Team user;
[0675] Team opponent;
[0676] int current_slot_index;
[0677] void PlayGame(Team user, Team opponent)
[0678] {
[0679]     this.user = user;
[0680]     this.opponent = opponent;
[0681]     int num_players = 5;
[0682]     for(int i = 0; i < num_players; i++)
[0683]     {
[0684]         this.current_slot_index = i;
[0685]         SetupNewAttempt();
[0686]         WaitForAttemptToComplete();
[0687]     }
[0688] }
[0689] }
```

[0690] 도 13은 본 발명의 시스템 및 방법의 실시예를 구현하기 위해 사용될 수 있는 예시적인 컴퓨팅 장치(1)(시스템(1) 및 컴퓨터 시스템(1)이라고도 함)를 도시한다. 도 13의 시스템(1)은 컴퓨팅 장치, 네트워크, 서버, 또는 이들의 조합 등의 환경에서 구현될 수 있다. 도 13의 컴퓨팅 장치(1)는 하나 이상의 프로세서(10) 및 메모리(20)를 포함한다. 메모리(20)는 프로세서(10)에 의해 실행되는 명령어 및 데이터를 부분적으로 저장한다. 메모리(20)는 동작 시 실행 가능한 코드를 저장할 수 있다. 도 13의 시스템(1)은 대용량 저장 장치(30), 휴대용

저장 장치(40), 출력 장치(50), 사용자 입력 장치(60), 디스플레이 시스템(70), 및 주변 장치(들)(80)을 더 포함한다.

[0695] 도 13에 도시된 컴포넌트들은 단일 버스(90)를 통해 연결되어 있는 것으로 도시되어 있다. 이 컴포넌트들은 하나 이상의 데이터 전송 수단을 통해 연결될 수 있다. 프로세서(10) 및 메모리(20)는 로컬 마이크로프로세서 버스를 통해 연결 될 수도 있고, 대용량 저장 장치(30), 주변 장치(80), 휴대용 저장 장치(40) 및 디스플레이 시스템(70)은 하나 이상의 입/출력(I/O) 버스를 통해 연결될 수 있다.

[0696] 자기 디스크 드라이브 또는 광 디스크 드라이브로 구현될 수 있는 대용량 저장 장치(30)는 프로세서(10)에 의해 사용될 데이터 및 명령어를 저장하기 위한 비휘발성 저장 장치이다. 대용량 저장 장치(30)는 메모리(20)에 소프트웨어를 로딩할 목적으로 본 발명의 실시예를 구현하기 위한 시스템 소프트웨어를 저장할 수 있다.

[0697] 휴대용 저장 장치(40)는 도 13의 컴퓨터 시스템(10)에 데이터 및 코드를 입력하거나 그로부터 출력하기 위해 플로피 디스크, 콤팩트 디스크, 디지털 비디오 디스크, 또는 USB 저장 장치와 같은 휴대용 비휘발성 저장 매체와 함께 동작한다. 본 발명의 실시예를 구현하기 위한 시스템 소프트웨어는 이러한 휴대용 매체에 저장되어 휴대용 저장 장치(40)를 통해 컴퓨터 시스템(1)으로 입력될 수 있다.

[0698] 사용자 입력 장치(60)는 사용자 인터페이스의 일부분을 제공한다. 사용자 입력 장치(60)는 알파벳 숫자 및 다른 정보를 입력하기 위한 키보드와 같은 알파벳 숫자 키패드, 또는 마우스, 트랙볼, 스타일러스와 같은 포인팅 장치, 또는 커버 방향 키를 포함할 수 있다. 추가적인 사용자 입력 장치(60)는 음성 인식 장치, 얼굴 인식 시스템, 모션 기반 입력 시스템, 및 제스처 기반 시스템 등과 같은 장치를 포함할 수 있으나, 이에 제한되는 것은 아니다. 예를 들어, 사용자 입력 장치(60)는 터치 스크린을 포함할 수 있다. 또한, 도 2에 도시된 시스템(1)은 출력 장치(50)를 포함한다. 적절한 출력 장치는, 스피커, 프린터, 네트워크 인터페이스, 및 모니터를 포함한다.

[0699] 디스플레이 시스템(70)은 액정 디스플레이(LCD) 또는 다른 적절한 디스플레이 장치를 포함할 수 있다. 디스플레이 시스템(70)은 텍스트 및 그래픽 정보를 수신하고 디스플레이 장치로 그 정보를 출력하기 위해 프로세싱한다.

[0700] 주변 장치(들)(80)는 컴퓨터 시스템에 추가 기능을 추가하기 위한 임의의 타입의 컴퓨터 지원 장치를 포함할 수 있다. 주변 장치(들)(80)는 모뎀 또는 라우터를 포함할 수 있다.

[0701] 도 2의 컴퓨터 시스템(1)에 제공된 컴포넌트들은 본 발명의 실시예와 함께 사용하기에 적합할 수 있는 컴퓨터 시스템 내에서 전형적으로 찾을 수 있는 것이고, 당업계에 공지된 그러한 컴퓨터 컴포넌트의 넓은 카테고리를 나타내도록 의도되었다. 그러므로, 도 2의 컴퓨터 시스템(1)은 퍼스널 컴퓨터, 휴대용 컴퓨팅 장치, 전화기, 모바일 컴퓨팅 장치, 워크스테이션, 서버, 미니컴퓨터, 메인프레임 컴퓨터, 또는 임의의 다른 컴퓨팅 장치일 수 있다. 또한, 컴퓨터는 다양한 버스 구성, 네트워크형 플랫폼, 다중 프로세서 플랫폼 등을 포함할 수 있다. 유닉스, 리눅스, 윈도우즈, Mac OS, 팜(Palm) OS, 안드로이드, iOS(2010년 6월 이전에는 아이폰(iPhone) OS로도 알려짐), QNX 및 다른 적절한 운영체제를 포함하는 다양한 운영체제가 사용될 수 있다.

[0702] 여기 서술된 프로세싱을 수행하는데 적합한 임의의 하드웨어 플랫폼이 여기 제공된 시스템 및 방법과 함께 사용하기에 적합함을 이해해야 한다. 컴퓨터 판독 가능한 저장 매체는 중앙 처리 장치(CPU), 프로세서 또는 마이크로컨트롤러 등에 명령어를 제공하는데 참여하는 임의의 매체 또는 매체들을 의미한다. 이러한 매체는 각각 광 또는 자기 디스크 및 동적 메모리와 같은 비휘발성 및 휘발성 매체를 포함하는 형태를 취할 수 있으나, 이에 제한되지는 않는다. 컴퓨터 판독 가능한 저장 매체의 일반적인 형태는 플로피 디스크, 플렉시블 디스크, 하드 디스크, 자기 테이프, 임의의 다른 자기 저장 매체, CD-ROM, 디지털 비디오 디스크(DVD), 임의의 다른 광 매체, RAM, PROM, EPROM, FLASHROM, 임의의 다른 메모리 칩 또는 카트리지를 포함한다.

[0703] 본 발명의 형태에 대한 오퍼레이션을 수행하기 위한 컴퓨터 프로그램 코드는 자바, 스마트토크(Smalltalk) 또는 C++과 같은 객체 지향 프로그래밍언어, 및 "C" 프로그래밍 언어 또는 유사한 프로그래밍 언어와 같은 종래의 절차적 프로그래밍 언어를 포함하는 하나 이상의 프로그래밍 언어 중 하나 이상의 임의의 조합으로 작성될 수 있다. 프로그램 코드는 전적으로 사용자의 컴퓨터상에서, 사용자의 컴퓨터에서 부분적으로, 사용자의 컴퓨터상에서 부분적으로 스탠드얼론 소프트웨어 패키지로서, 그리고 원격 컴퓨터상에서 부분적으로, 또는 원격 컴퓨터 또는 서버 상에서 전적으로 실행될 수도 있다. 후자의 시나리오에서, 원격 컴퓨터는 로컬 영역 네트워크(LAN), 또는 광역 네트워크(WAN)를 포함하는 임의의 타입의 네트워크를 통해 사용자의 컴퓨터와 연결될 수 있고, 또는 (예컨대, 인터넷 서비스 제공자를 이용하여 인터넷을 통해) 외부 컴퓨터에 이러한 연결이 이루어질 수도 있다

[0704]

아래의 청구항에서 모든 수단 또는 단계 플러스 기능 엘리먼트의 대응 구조, 재료, 행동, 및 동등물은 구체적으로 청구된 다른 청구된 엘리먼트와 결합하여 그 기능을 수행하기 위한 임의의 구조, 재료 또는 행동을 포함하도록 의도되었다. 본 발명의 설명은 예시와 설명의 목적으로 제공된 것이고, 본 발명을 개시된 형태로 제한하거나 그것이 완벽한(exhaustive) 설명이 되도록 의도된 것은 아니다. 본 발명의 범위 및 정신을 벗어나지 않는 다양한 수정 및 변형이 당업자들에게 명백할 것이다. 예시적인 실시예는 본 발명의 원리 및 그 실제적 응용을 가장 잘 설명하도록 그리고 고려된 특정 용도에 적합한 다양한 수정을 포함하는 다양한 실시예에 대하여 당업자들이 본 발명을 이해할 수 있도록 선택되고 설명되었다.

[0705]

본 발명의 형태는 본 발명의 실시예에 따른 방법, 장치 및 컴퓨터 프로그램 프로덕트의 흐름도 도면 및/또는 블록도를 참조하여 앞서 설명되었다. 흐름도 도면 및/또는 블록도의 각각의 블록 및 흐름도 도면 및/또는 블록도 내의 블록의 조합은 컴퓨터 프로그램 명령어로 구현될 수 있음이 이해될 것이다. 이러한 컴퓨터 프로그램 명령어는 컴퓨터의 프로세서 또는 다른 프로그래밍 가능한 데이터 프로세싱 장치를 통해 실행되는 명령어들이 흐름도 및/또는 블록도 블록 또는 블록들에 명시된 기능/동작을 구현하는 명령어들을 포함하는 제조품(article of manufacture)을 만들어내도록 하는 특정 방식으로, 컴퓨터, 다른 프로그래밍 가능한 데이터 처리 장치, 또는 다른 장치들을 기능하게 만들 수 있는 컴퓨터 저장 매체에 저장될 수 있다.

[0706]

또한, 이러한 컴퓨터 프로그램 명령어는 컴퓨터 판독 가능한 매체에 저장된 명령어가 흐름도 및/또는 블록도 블록 또는 블록들에 명시된 기능/동작을 구현하는 명령어들을 포함하는 제조품(article of manufacture)을 만들어내도록 하는 특정 방식으로, 컴퓨터, 다른 프로그래밍 가능한 데이터 처리 장치, 또는 다른 장치들을 기능하게 만들 수 있는 컴퓨터 저장 매체에 저장될 수 있다.

[0707]

또한, 컴퓨터 프로그램 명령어는 컴퓨터, 다른 프로그래밍 가능한 장치, 또는 다른 장치가 컴퓨터로 구현되는 프로세스를 산출하게 만들기 위해 컴퓨터, 다른 프로그래밍 가능한 데이터 처리 장치, 또는 다른 장치에 로딩되고, 그로 인해 컴퓨터 또는 다른 프로그래밍 가능한 장치상에서 실행되는 명령어들이 흐름도 및/또는 블록도 블록 또는 블록들에 명시된 기능/동작을 구현하기 위한 프로세스를 제공할 수 있다.

[0708]

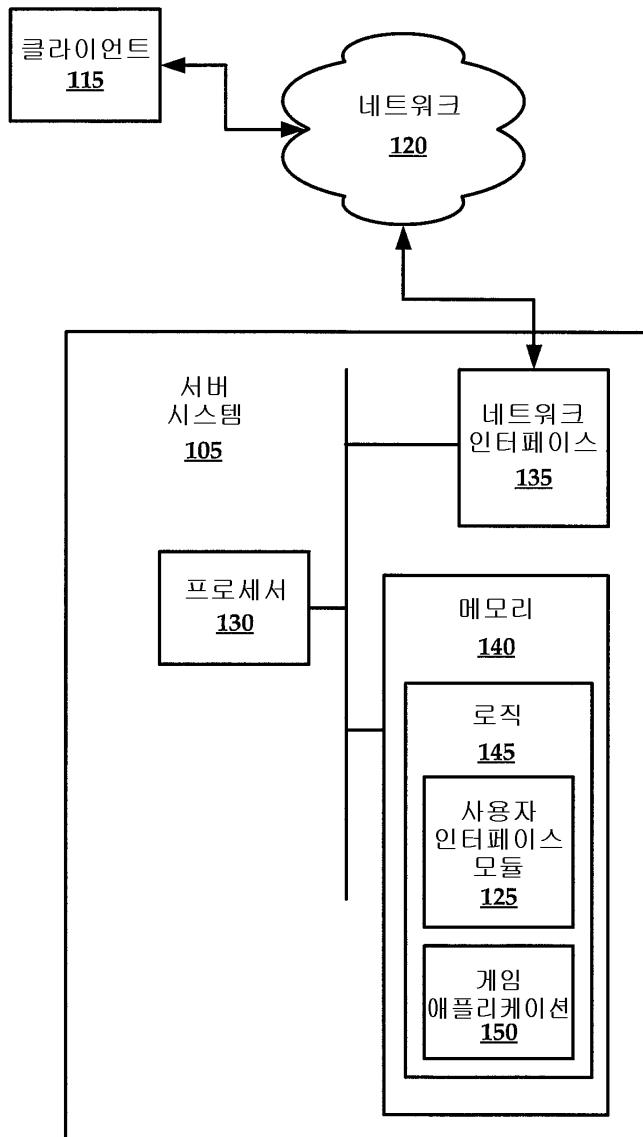
도면의 흐름도 및 블록도는 본 발명의 다양한 실시예에 따른 시스템, 방법, 및 컴퓨터 프로그램 프로덕트의 가능한 구현 방법의 아키텍처, 기능, 및 동작을 보여준다. 이와 관련하여, 흐름도 또는 블록도 내의 각각의 블록은 명시된 논리적 기능(들)을 구현하기 위한 하나 이상의 실행 가능한 명령어를 포함하는 모듈, 세그먼트, 또는 코드의 일부분을 나타낼 수 있다. 또한, 몇몇 대안의 구현 방법에서, 블록 내에 언급된 기능들이 도면에 나열된 순서와 다르게 실행될 수도 있음을 이해해야 한다. 예를 들어, 순차적으로 도시된 2개의 블록은 실제로 실질적으로 동시에 수행될 수도 있고, 또는 이 블록들은 포함된 기능에 따라 종종 역순으로 실행될 수도 있다. 또한, 블록도 및/또는 흐름도의 각각의 블록, 및 블록도 및/또는 흐름도의 블록의 조합이 특수한 기능 또는 동작을 수행하는 특수 목적 하드웨어 기반 시스템, 또는 특수 목적 하드웨어 및 컴퓨터 명령어의 조합에 의해 구현될 수 있음이 이해될 것이다.

[0709]

다양한 실시예들이 상술되었으나, 이들은 단지 예로서 제공된 것일 뿐이며, 제한이 아님을 이해해야 한다. 이러한 설명은 본 발명의 범위를 여기 나열된 특정 형태로 제한하도록 의도된 것이 아니다. 그러므로, 바람직한 실시예의 폭 및 범위는 임의의 앞서 언급된 예시적인 실시예에 의해 제한되어서는 안 된다. 상기 설명은 설명을 위한 것이고 제한이 아님을 이해해야 한다. 이와 달리, 본 설명은 청구항에 의해 정의되고 당업자들에 의해 이해된 본 발명의 정신 및 범위 내에 포함될 수 있는, 그러한 대안, 수정 및 동등물을 커버하도록 의도되었다. 그러므로, 본 발명의 범위는 상기 설명을 참조하여 정해져서는 안되고, 그 대신 첨부된 청구항 및 그 동등물의 전체 범위를 고려하여 결정되어야 한다.

도면

도면1



도면2

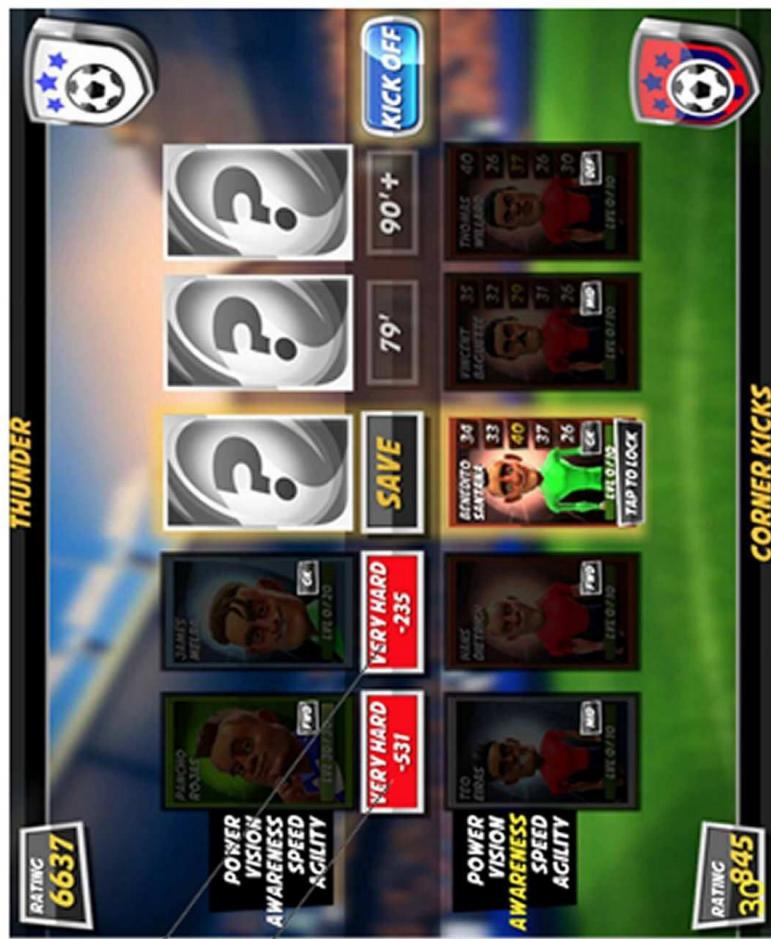


200

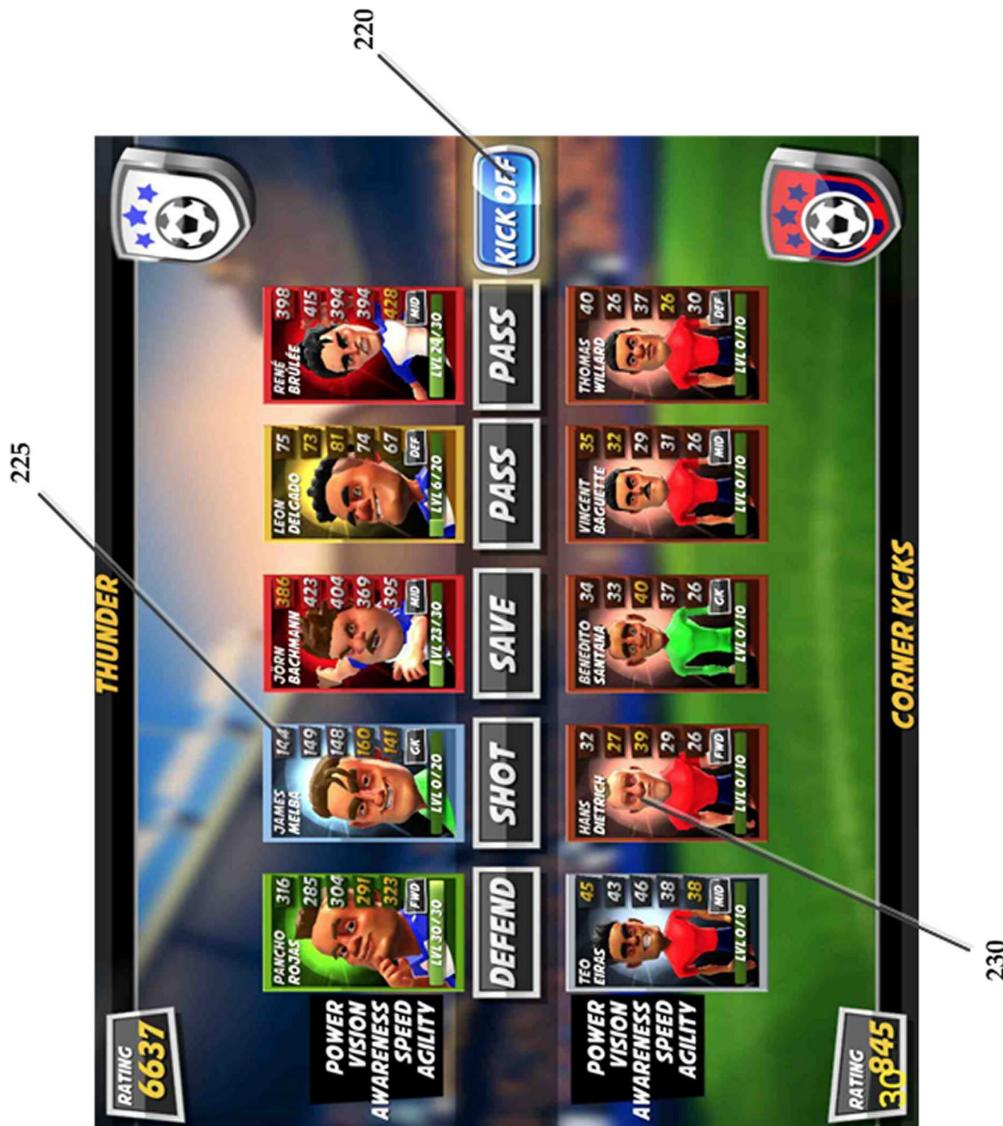
205

210

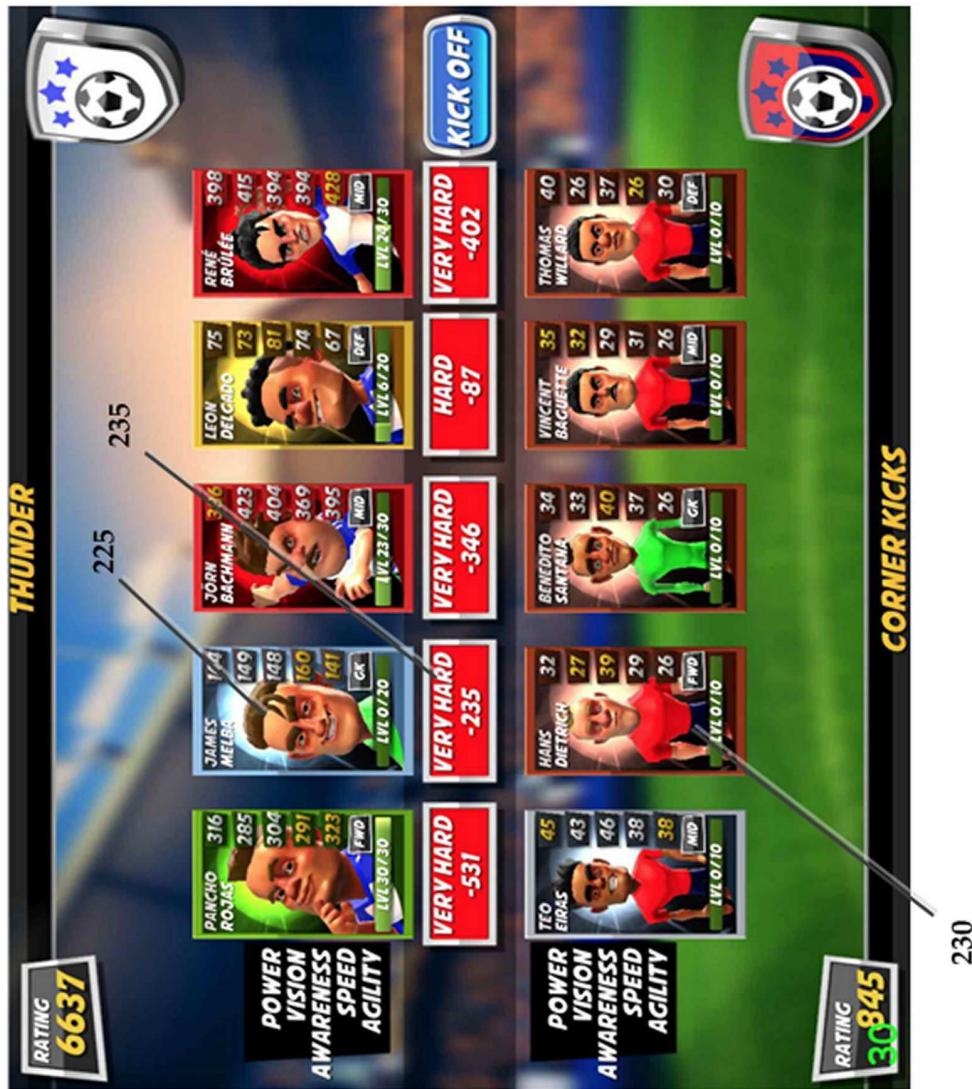
도면3



도면4



도면5



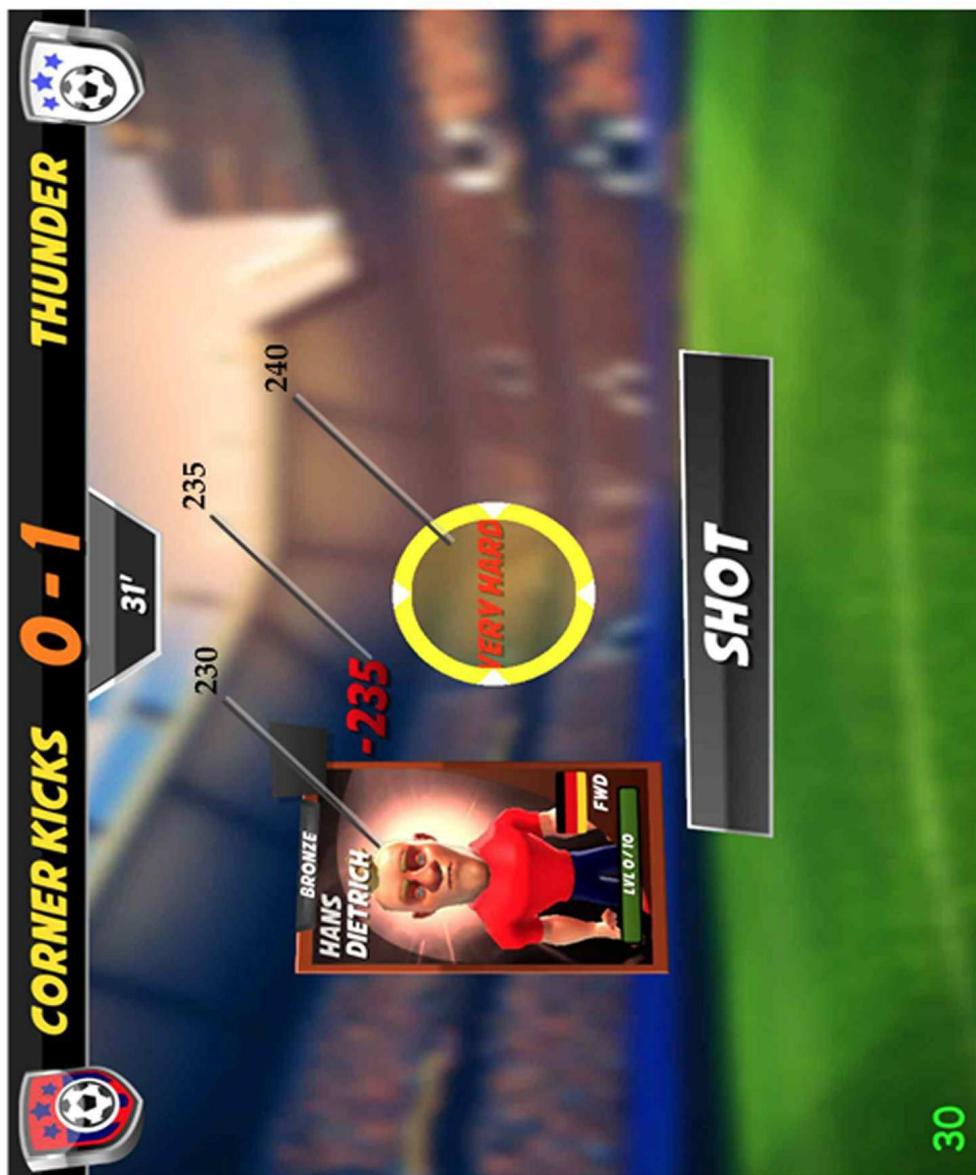
## 도면6



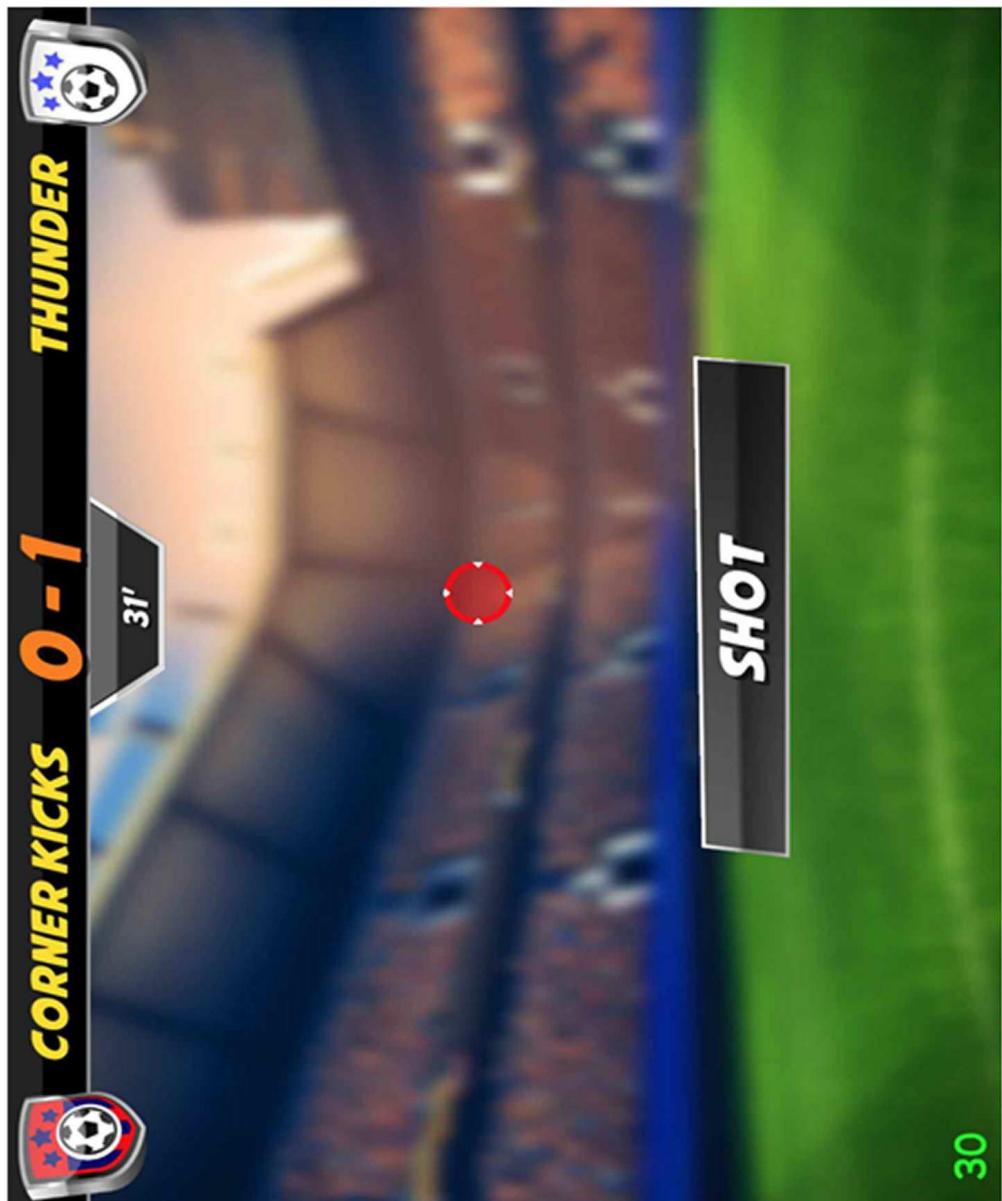
도면7



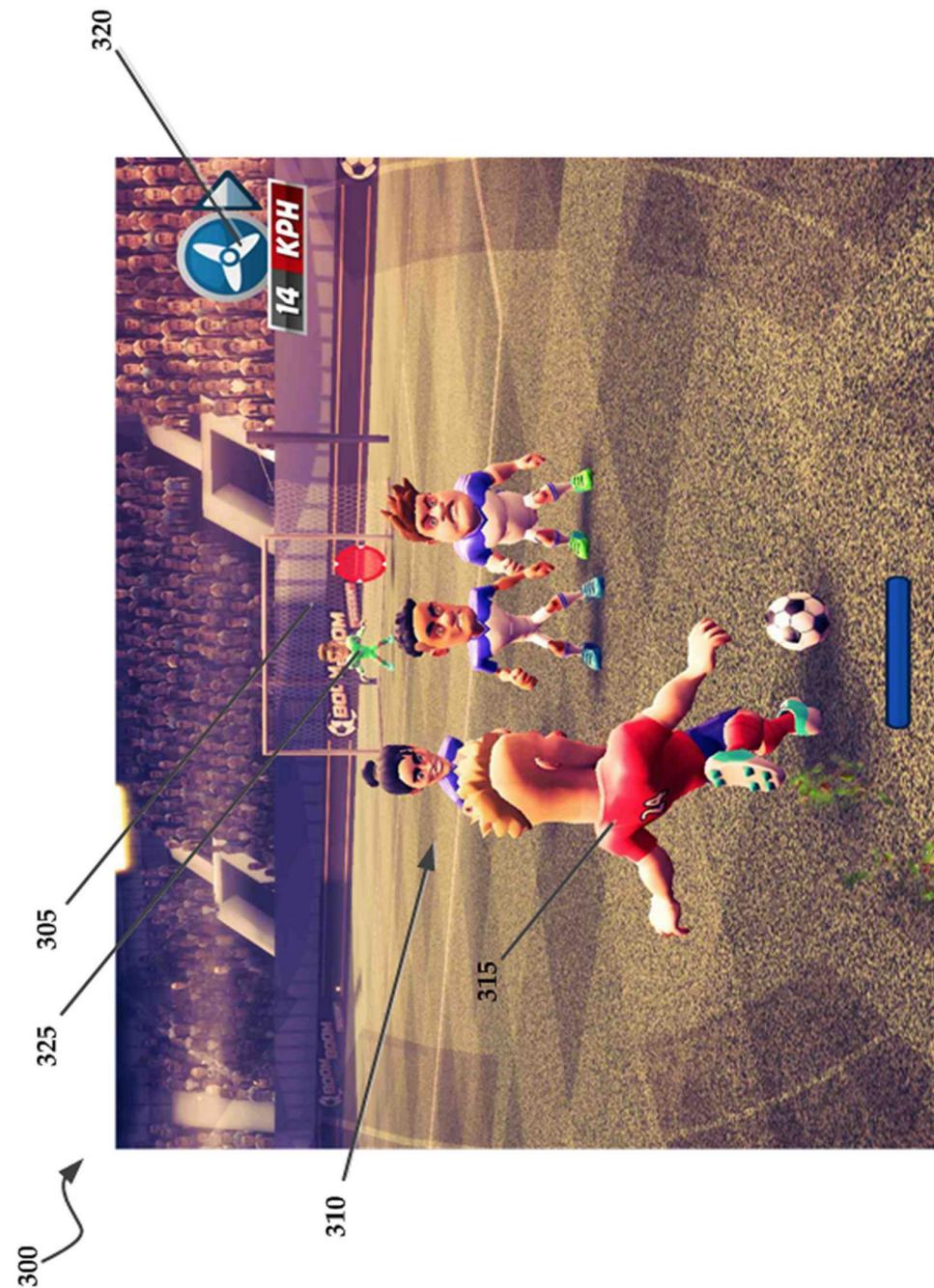
도면8



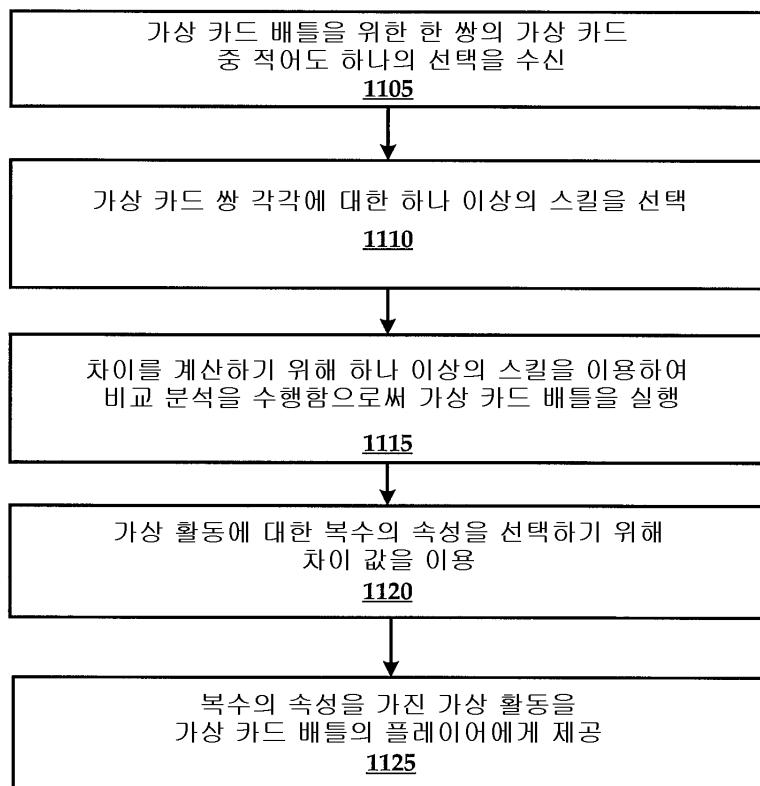
도면9



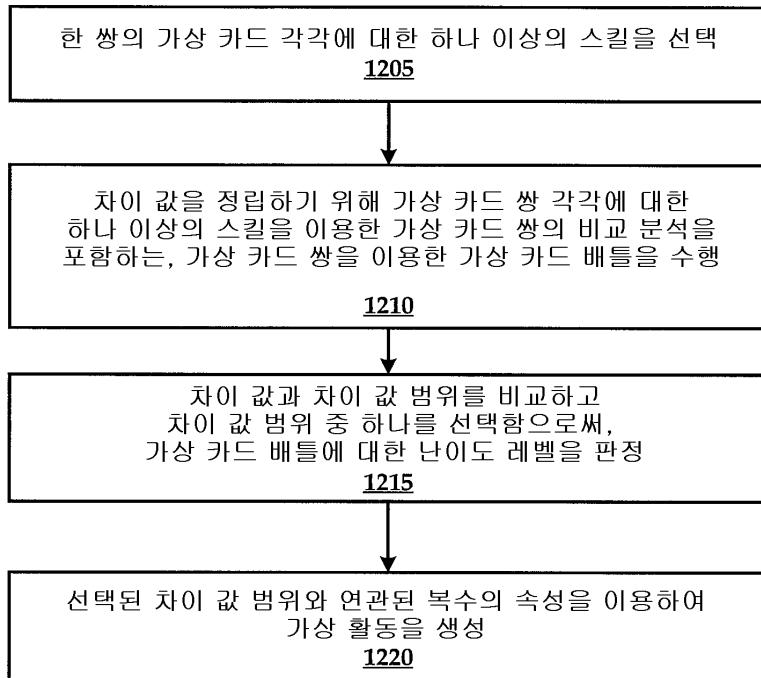
도면10



## 도면11



## 도면12



## 도면13

