



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년11월02일
(11) 등록번호 10-1791951
(24) 등록일자 2017년10월25일

(51) 국제특허분류(Int. Cl.)
G06F 21/50 (2013.01) G06F 3/041 (2006.01)
(52) CPC특허분류
G06F 21/50 (2013.01)
G06F 3/041 (2013.01)
(21) 출원번호 10-2016-0115529
(22) 출원일자 2016년09월08일
심사청구일자 2016년09월08일
(56) 선행기술조사문헌
KR101202414 B1*
KR1020130109590 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
정윤혁
서울특별시 노원구 성북로 139, 113동 1603호 (공릉동, 공릉풍림아파트)
김정현
인천광역시 중구 서해대로464번길 6 202호 (신흥동2가, 미성홈타운)
(74) 대리인
김등용, 김홍석
(뒷면에 계속)

전체 청구항 수 : 총 4 항

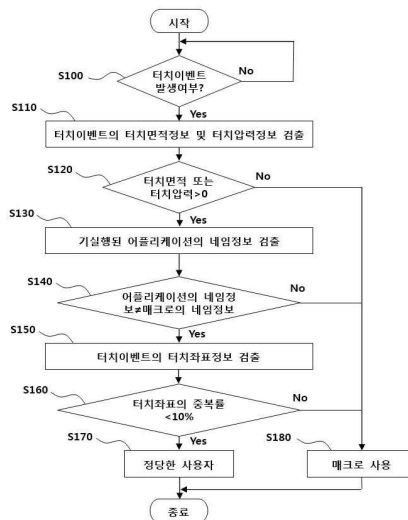
심사관 : 문남두

(54) 발명의 명칭 안드로이드 OS 기반의 매크로 탐지방법

(57) 요약

본 발명에 따른 매크로 탐지방법은 사용자로부터 입력된 터치이벤트를 검출하는 제 1 단계, 상기 터치이벤트로부터 산출된 터치면적정보 또는 터치압력정보를 기설정된 기준값과 비교하여, 상기 터치이벤트의 매크로에 의한 발생여부를 판단하는 제 2 단계 및 상기 터치이벤트가 검출된 경우, 기설정되고 있는 어플리케이션의 네임정보와 기설정된 매크로 어플리케이션의 네임정보를 비교하여, 상기 터치이벤트의 매크로 사용여부를 판단하는 제 3 단계를 포함한다.

대표도 - 도1



(52) CPC특허분류

G06F 2221/2109 (2013.01)

(72) 발명자

최진영

서울특별시 노원구 동일로248길 16, 506동 1003호
(상계동, 수락파크빌아파트)

이상진

서울특별시 중로구 사직로8길 20, 101동 1503호 (내수동, 파크팰리스)

이 발명을 지원한 국가연구개발사업

과제고유번호 H85011610120001002

부처명 미래창조과학부

연구관리전문기관 정보통신기술진흥센터

연구사업명 대학ICT연구센터지원사업

연구과제명 고품질 융합 소프트웨어 센터

기 여 율 1/1

주관기관 고려대학교 산학협력단

연구기간 2016.01.01 ~ 2016.12.31

공지예외적용 : 있음

명세서

청구범위

청구항 1

사용자로부터 입력된 터치이벤트를 검출하는 제 1 단계;

상기 터치이벤트로부터 산출된 터치면적정보 또는 터치압력정보를 기설정된 기준값과 비교하여, 상기 터치이벤트의 매크로에 의한 발생여부를 판단하는 제 2 단계; 및

상기 터치이벤트가 검출된 경우, 기실행되고 있는 어플리케이션의 네임정보와 기설정된 매크로 어플리케이션의 네임정보를 비교하여, 상기 터치이벤트의 매크로 사용여부를 판단하는 제 3 단계를 포함하고,

상기 제 3 단계는

상기 터치이벤트가 검출된 경우, 상기 터치이벤트가 입력된 프로세스 상에서 실행되고 있는 어플리케이션의 네임정보를 검출하는 제 3-1 단계;

상기 어플리케이션의 네임정보를 기설정된 매크로 어플리케이션의 네임정보와 비교하는 제 3-2 단계; 및

상기 어플리케이션의 네임정보가 상기 매크로 어플리케이션의 네임정보와 동일한 경우, 상기 터치이벤트가 상기 매크로 어플리케이션에 의해 발생한 것으로 판단하는 제 3-3 단계를 포함하는 매크로 탐지방법.

청구항 2

청구항 1에 있어서,

상기 제 2 단계는

상기 터치이벤트가 검출된 경우, 상기 터치이벤트로부터 발생된 터치면적정보 와 터치압력정보를 검출하는 제 2-1 단계;

상기 터치면적정보와 터치압력정보를 기설정된 기준값과 비교하는 제 2-2 단계; 및

상기 터치면적정보 또는 터치압력정보 중 어느 하나가 상기 기준값인 '0'과 같은 경우, 상기 터치이벤트가 매크로에 의해 발생한 것으로 판단하는 제 2-3 단계를 포함하는 매크로 탐지방법.

청구항 3

삭제

청구항 4

청구항 1에 있어서,

상기 터치이벤트가 검출된 경우, 상기 터치이벤트에 대응하는 터치좌표정보를 검출하는 단계;

상기 터치좌표정보와 먼저 발생된 터치이벤트에 대응하는 전체의 터치좌표정보를 비교하여, 상기 터치좌표정보의 중복률을 산출하는 단계; 및

상기 터치좌표정보의 중복률과 기설정된 기준률을 비교하여, 상기 터치좌표정보의 중복률이 상기 기준률 이상인 경우, 상기 터치이벤트가 매크로에 의해 발생한 것으로 판단하는 단계를 더 포함하는 매크로 탐지방법.

청구항 5

청구항 1에 있어서,

상기 터치이벤트가 검출된 경우, 상기 터치이벤트에 대응하는 터치좌표정보를 검출하는 단계;

상기 터치좌표정보와 먼저 발생된 터치이벤트에 대응하는 전체의 터치좌표정보를 비교하는 단계;

상기 전체의 터치좌표정보 중 상기 터치좌표정보의 중복된 발생횟수를 카운트를 하는 단계; 및

상기 터치좌표정보의 발생횟수가 기설정된 기준횟수 이상인 경우, 상기 터치이벤트가 매크로에 의해 발생한 것으로 판단하는 단계를 더 포함하는 매크로 탐지방법.

발명의 설명

기술 분야

[0001] 본 발명은 안드로이드 OS 상의 매크로 탐지방법에 관한 것이다.

배경 기술

[0003] 스마트폰의 보급이 증가함에 따라 게임산업은 PC 온라인 게임에서 모바일 게임으로 트렌드가 변화하고 있다. 그리고 모바일 게임시장이 성장함에 따라 보안문제도 중요한 이슈가 되었다. 모바일 게임 보안은 부정행위와 해킹으로부터 사용자의 개인정보와 사이버 자산을 보호하는 것이다. 게임 도중 획득한 게임 아이템과 머니를 다른 플레이어에게 팔거나 현금으로 교환할 수 있는 PC 온라인 게임처럼 모바일 게임 또한 현금 거래가 이루어지고 있다.

[0004] 모바일 게임의 매크로(자동사냥)는 PC 온라인 게임의 게임봇과 같이 정해놓은 규칙에 의해 화면을 터치하는 프로그램으로서 안드로이드 애플리케이션과 PC응용프로그램 등 다양한 형태로 만들어져 사용된다. 매크로는 시간의 제약없이 플레이가 가능하기 때문에 게임 아이템, 머니 등과 같은 사이버 자산과 임무의 완수, 적이나 장애물의 극복, 룰플레이의 성공에 대한 보상으로 주어지는 경험치 등을 일반 사용자보다 빠르게 얻을 수 있다. 이러한 매크로 사용은 정직한 사용자에게 박탈감을 주고 흥미를 잃게 하여 게임을 떠나게 만들고 게임수명을 단축시킨다.

선행기술문헌

특허문헌

[0006] (특허문헌 0001) 대한민국 등록특허 제1005593호

발명의 내용

해결하려는 과제

[0007] 본 발명은 안드로이드 OS 기반의 어플리케이션에 있어서, 매크로를 이용한 자동화 침해를 방지하기 위해, 사용자의 터치이벤트 발생시, 매크로의 실행을 탐지할 수 있는 방법을 제공하기 위함이다.

과제의 해결 수단

[0009] 본 발명에 따른 매크로 탐지방법은 사용자로부터 입력된 터치이벤트를 검출하는 터치이벤트 검출한 제 1 단계, 상기 터치이벤트로부터 산출된 터치면적정보 또는 터치압력정보를 기설정된 기준값과 비교하여, 상기 터치이벤트의 매크로에 의한 발생여부를 판단하는 제 2 단계 및 상기 터치이벤트가 검출된 경우, 기실행되고 있는 어플리케이션의 네임정보와 기설정된 매크로 어플리케이션의 네임정보를 비교하여, 상기 터치이벤트의 매크로 사용여부를 판단하는 제 3 단계를 포함한다.

[0011] 또한, 상기 제 2 단계는 상기 터치이벤트가 검출된 경우, 상기 터치이벤트로부터 발생된 터치면적정보 와 터치압력정보를 검출하는 제 2-1 단계, 상기 터치면적정보 와 터치압력정보를 기설정된 기준값과 비교하는 제 2-2 단계 및 상기 터치면적정보 또는 터치압력정보 중 어느 하나가 상기 기준값인 '0' 과 같은 경우, 상기 터치이

벤트가 매크로에 의해 발생한 것으로 판단하는 제 2-3 단계를 포함한다.

[0013] 또한, 상기 제 3 단계는 상기 터치이벤트가 검출된 경우, 상기 터치이벤트가 입력된 프로세스 상에서 실행되고 있는 어플리케이션의 네임정보를 검출하는 제 3-1 단계 및 상기 어플리케이션의 네임정보를 기설정된 매크로 어플리케이션의 네임정보와 비교하는 제 3-2 단계 및 상기 어플리케이션의 네임정보가 상기 매크로 어플리케이션의 네임정보와 동일한 경우, 상기 터치이벤트가 상기 매크로 어플리케이션에 의해 발생한 것으로 판단하는 제 3-3 단계를 포함한다.

[0015] 또한, 상기 터치이벤트가 검출된 경우, 상기 터치이벤트에 대응하는 터치좌표정보를 검출하는 단계, 상기 터치좌표정보와 먼저 발생된 터치이벤트에 대응하는 전체의 터치좌표정보를 비교하여, 상기 터치좌표정보의 중복률을 산출하는 단계 및 상기 터치좌표정보의 중복률과 기설정된 기준률을 비교하여, 상기 터치좌표정보의 중복률이 상기 기준률 이상인 경우, 상기 터치이벤트가 매크로에 의해 발생한 것으로 판단하는 단계를 더 포함한다.

[0017] 또한, 상기 터치이벤트가 검출된 경우, 상기 터치이벤트에 대응하는 터치좌표정보를 검출하는 단계, 상기 터치좌표정보와 먼저 발생된 터치이벤트에 대응하는 전체의 터치좌표정보를 비교하는 단계, 상기 전체의 터치좌표정보 중 상기 터치좌표정보의 중복된 발생횟수를 카운트를 하는 단계 및 상기 터치좌표정보의 발생횟수가 기설정된 기준횟수 이상인 경우, 상기 터치이벤트가 매크로에 의해 발생한 것으로 판단하는 단계를 더 포함한다.

발명의 효과

[0019] 본 발명에 따른 안드로이드 OS 기반의 매크로 탐지방법은 사용자의 터치이벤트로부터 발생된 터치면적정보와 터치압력정보를 기초로 하여, 정당한 사용자와 불법적인 매크로 사용을 구별하여 탐지 할 수 있다.

[0021] 그리고, 본 발명에 따른 안드로이드 OS 기반의 매크로 탐지방법은 사용자의 터치이벤트에 기초한 탐지방법과 함께 보완적으로 기실행되고 있는 어플리케이션의 네임정보를 이용하여, 특정 매크로 어플리케이션의 실행여부를 탐지함으로써, 불법적인 행동을 통해 서비스 침해를 가하는 사용자를 구별할 수 있다.

[0023] 또한, 본 발명에 따른 안드로이드 OS 기반의 매크로 탐지방법은 사용자의 터치이벤트에 기초한 터치좌표정보를 이용하여, 상기 터치좌표정보의 중복률 또는 발생횟수의 카운트 정보를 이용하여, 보다 효율적으로 매크로의 부정사용을 탐지할 수 있다.

도면의 간단한 설명

[0025] 도 1은 본 발명의 일실시예에 따른 안드로이드 OS 기반의 매크로 탐지방법을 나타낸 흐름도이다.

도 2는 본 발명의 다른 실시예에 따른 안드로이드 OS 기반의 매크로 탐지방법을 나타낸 흐름도이다.

도 3a 와 3b는 본 발명의 일실시예에 따른 사용자의 터치이벤트에 대응하는 터치정보를 이용하여 매크로를 탐지하는 단계에 있어서, 도 3a는 상기 단계를 수도코드(Pseudo code)로 나타낸 것이며, 도 3b는 Java code를 이용해 나타낸 도면이다.

도 4a 내지 도 4c는 본 발명의 일실시예에 따른 기실행되고 있는 어플리케이션의 네임정보와 기설정된 매크로 어플리케이션의 네임정보를 비교하여, 매크로를 탐지하는 단계에 있어서, 도 4a는 안드로이드 OS 상에서의 ADB(Android debug bridge)의 shell 명령어를 통해, 현재 실행되고 있는 어플리케이션의 정보를 나타낸 도면이며, 도 4b는 상기 단계를 수도코드(Pseudo code)로 나타낸 것이며, 도 4c는 Java code를 이용해 나타낸 도면이다.

도 5a 내지 5d는 본 발명의 일실시예에 따른 터치이벤트에 대응하는 터치좌표정보를 이용하여, 매크로를 탐지하는 단계에 있어서, 도 5a 와 5b는 터치좌표정보의 중복률을 이용하는 단계를 나타낸 도면이며, 도 5c 와 5d는 터치좌표정보의 중복된 발생횟수를 이용하는 단계를 나타낸 도면이다.

발명을 실시하기 위한 구체적인 내용

[0026] 본 발명의 목적, 특정한 장점들 및 신규한 특징들은 첨부된 도면들과 연관되어지는 이하의 상세한 설명과 바람직한 실시예들로부터 더욱 명백해질 것이다. 본 명세서에서 각 도면의 구성요소들에 참조번호를 부가함에 있어서, 동일한 구성 요소들에 한해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 번호를 가지도록 하고 있음에 유의하여야 한다. 또한, "일면", "타면", "제1", "제2" 등의 용어는 하나의 구성요소를 다른 구성요소로부터 구별하기 위해 사용되는 것으로, 구성요소가 상기 용어들에 의해 제한되는 것은 아니다.

- [0027] 본 발명을 설명함에 있어서, 본 발명의 요지를 불필요하게 흐릴 수 있는 관련된 공지 기술에 대한 상세한 설명은 생략하며, 실시예가 다른 경우에도 동일 번호는 동일한 부품 또는 소재를 나타낸다.
- [0029] 그리고, 본 발명은 안드로이드 OS를 기반으로 작동하는 어플리케이션 환경에서, 매크로를 이용하는 사용자, 루팅된 안드로이드 OS에서 매크로 어플리케이션을 사용하는 사용자 및 윈도우기반 OS에서 안드로이드 에뮬레이터와 매크로 프로그램을 사용하는 사용자를 전제로 한다. 그러나, 본 발명의 권리범위가 OS의 종류에 제한되지 않음은 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 자명하며, 기반이 되는 OS는 안드로이드 OS 이외에 iOS 및 윈도우(window) 등을 더 포함할 수 있다.
- [0031] 매크로(자동사냥)는 일련의 작업을 수행하기 위해 조작자가 매번 명령을 입력해 주지않고, 특정한 매크로 기능을 가진 키를 눌러줌으로써, 자동적으로 일이 수행되게 하는 프로그램을 말하며, 모바일 게임 매크로는 사용환경에 따라 안드로이드 애플리케이션과 Window 응용 프로그램으로 분류할 수 있다.
- [0033] 예를들면, 안드로이드 애플리케이션 매크로는 HiroMacro, Finger Replayer(FRep) 등 일 수 있으며, HiroMacro는 루트권한이 필요하고 FRep는 루트권한이 없더라도 ADB(Android Debug Bridge)도구를 통해, 재부팅되기 전까지 일시적으로 사용할 수 있다. 또한, Window 응용 프로그램 매크로는 에뮬레이터를 이용하여, PC 환경에서 안드로이드 애플리케이션을 사용하며, 지니모션, 블루스택, 윈드روی 등일 수 있다.
- [0035] 이하, 첨부된 도면을 참고하여, 본 발명에 따른 안드로이드 OS 기반의 매크로 탐지방법에 대하여, 보다 상세히 설명할 것이다.
- [0037] 도 1은 본 발명의 일실시예에 따른 안드로이드 OS 기반의 매크로 탐지방법을 나타낸 흐름도이며, 도 2는 본 발명의 다른 실시예에 따른 안드로이드 OS 기반의 매크로 탐지방법을 나타낸 흐름도이다. 여기에서, 본 발명의 다른 실시예는 일실시예와 중복되는 부분의 설명은 생략한다.
- [0039] 본 발명의 일실시예에 따른 안드로이드 OS 기반의 매크로 탐지방법은 사용자로부터 입력된 터치이벤트를 검출하는 제 1 단계, 상기 터치이벤트로부터 산출된 터치면적정보 또는 터치압력정보를 기설정된 기준값과 비교하여, 상기 터치이벤트의 매크로에 의한 발생여부를 판단하는 제 2 단계 및 상기 터치이벤트가 검출된 경우, 기설정되고 있는 어플리케이션의 네임정보와 기설정된 매크로 어플리케이션의 네임정보를 비교하여, 상기 터치이벤트의 매크로 사용여부를 판단하는 제 3 단계를 포함한다.
- [0041] 도 1에 도시된 바와 같이, 본 발명의 일실시예에 따른 안드로이드 OS 기반의 매크로 탐지방법은 안드로이드 운영체제를 탑재한 스마트폰에서 화면터치와 여러센서 등을 이용할 수 있는 안드로이드 모바일 게임에 적용가능하며, 모바일 게임을 위해 필수적으로 사용되는 사용자의 터치이벤트의 발생여부를 판단한다(S100,S200). 여기에서, 사용자가 디바이스의 화면을 터치할 시 안드로이드는 터치 이벤트 정보를 Motion event 객체로 반환하며, 애플리케이션 개발자는 Motion event 객체로부터 터치좌표, 스크린에 닿은 손가락 면적, 압력, 시간정보 등을 확인할 수 있다.
- [0043] 그리고, 상기 터치이벤트로부터 산출된 터치면적정보 또는 터치압력정보를 기설정된 기준값과 비교하여, 상기 터치이벤트의 매크로에 의한 발생여부를 판단한다(제2단계).
- [0045] 즉, 사용자로부터 입력된 상기 터치이벤트가 검출된 경우, 상기 터치이벤트로부터 발생된 터치면적정보와 터치압력정보를 검출하며(S110,S210)(제 2-1 단계). 상기 터치면적정보와 터치압력정보를 기설정된 기준값과 비교한다(S120,S220)(제 2-2 단계).
- [0046] 그리고, 상기 터치면적정보 또는 터치압력정보 중 어느 하나가 상기 기준값인 '0'과 같은 경우, 상기 터치이벤트가 매크로에 의해 발생한 것으로 판단한다(제 2-3 단계).
- [0048] 보다 구체적으로, 1) 정당한 사용자에게 의해 상기 터치이벤트가 발생된 경우, 상기 터치이벤트의 터치면적정보 또는 터치압력정보는 유동적인 값을 갖는 반면, 2) 상기 터치이벤트가 매크로에 의해 발생된 경우, 상기 터치이벤트의 터치면적정보 또는 터치압력정보는 고정값('0')을 갖는다.
- [0049] 따라서, 상기 터치이벤트의 터치면적정보 또는 터치압력정보 중 어느 하나가 상기 기준값('0')과 같은 경우, 상기 터치이벤트가 매크로에 의해 발생한 것으로 판단할 수 있으나, 에뮬레이터 환경에서 사용되는 매크로는 탐지를 할 수 없다.
- [0051] 예를 들면, 도 3a에 도시된 바와 같이, 상기 터치이벤트에 대응되는 상기 터치압력정보가 기준값('0')과 동일한 경우, ' Macro detection '이라는 문구를 출력하도록 수도(pseudo code)코드로 나타낼 수 있으며, 도 3b에 도

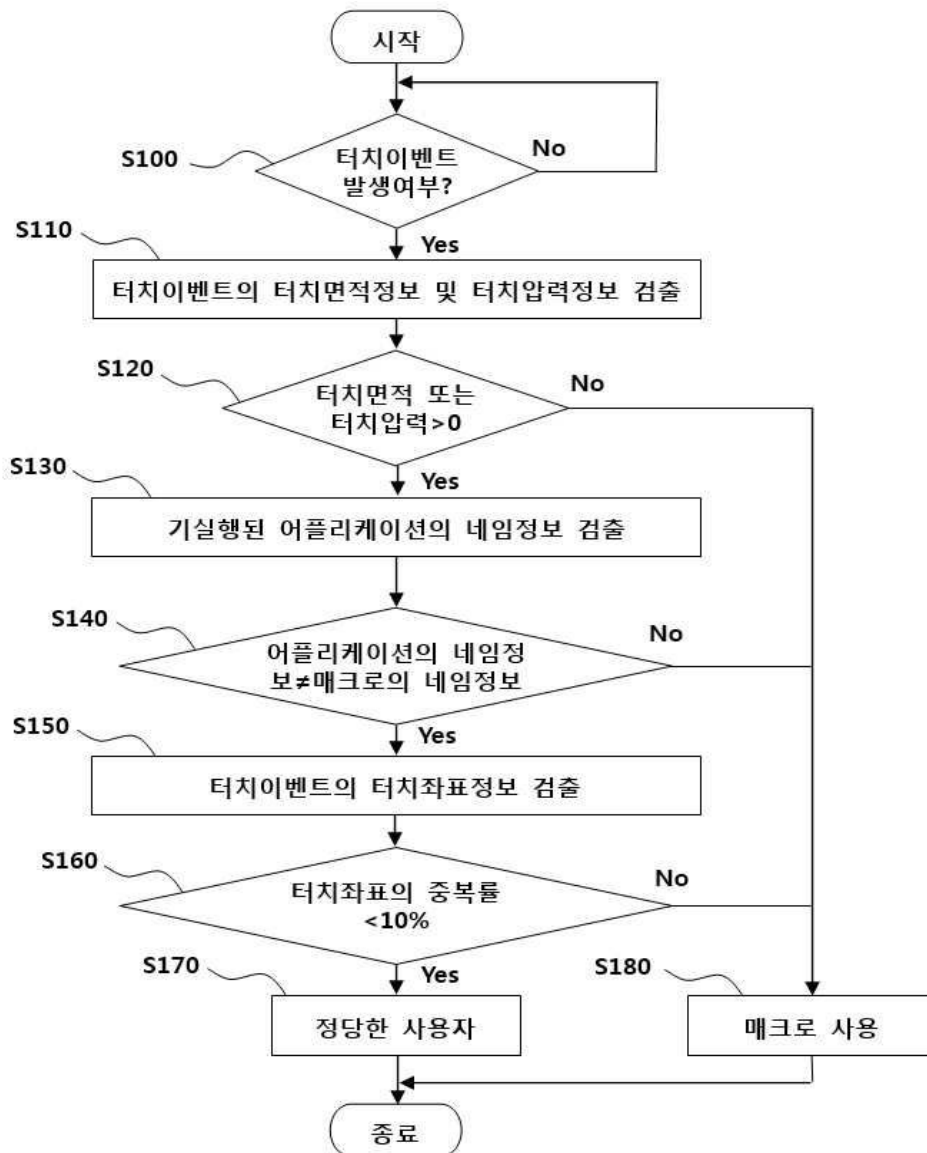
시된 바와 같이, Java code를 이용하여, 상기 터치압력정보가 기준값('0')과 동일한 경우, 사용자에게 Toast message를 통해, 매크로가 탐지됨을 알려줄 수 있도록 코딩될 수 있으나, 이에 한정되는 것은 아니다.

- [0053] 다음으로, 상기 터치이벤트가 검출된 경우, 상기 터치이벤트가 입력된 프로세스 상에서 실행되고 있는 어플리케이션의 네임(name)정보를 검출하며(제 3-1 단계)(S130, S230), 상기 어플리케이션의 네임(name)정보를 기설정된 매크로 어플리케이션의 네임(name)정보와 비교하고(제 3-2 단계)(S140, S240), 상기 어플리케이션의 네임정보가 상기 매크로 어플리케이션의 네임정보와 동일한 경우, 상기 터치이벤트가 상기 매크로 어플리케이션에 의해 발생한 것으로 판단한다(제 3-3 단계).
- [0054] 여기에서, 어플리케이션의 네임정보를 이용하는 방법은 전술한 터치면적정보등을 이용한 방법이 에뮬레이터에서의 매크로 사용을 탐지할 수 없는 단점을 보완하지만, Windows 응용 매크로는 탐지를 할 수 없다.
- [0056] 즉, 도 4a에 도시된 바와 같이, 안드로이드는 Linux 커널 기반의 OS 이기에 Linux 와 동일한 정보를 제공한다. 안드로이드 OS 상에서, ADB의 SHELL에서 PS명령어를 내리면, 현재 프로세스에서 어떤 어플리케이션이 실행되고 있는지 와 상기 어플리케이션의 여러정보를 확인할 수 있다.
- [0058] 예를 들면, 'PID'는 어플리케이션의 실행 시 부여받는 ID 이며, 'PPID'는 부모 프로세스 ID 이고, 'NAME'은 어플리케이션의 이름이며, 프로세스의 여러정보 중 매번 실행시에 유일하게 변하지 않는 정보가 네임(NAME)정보이다.
- [0060] 나아가, 도 4b에 도시된 바와 같이, 상기 터치이벤트가 검출된 경우, 상기 터치이벤트가 입력된 프로세스 상에서 실행되고 있는 어플리케이션의 네임(name)정보가 'Monkey'인 경우, 'Macro detection'이라는 문구를 출력하도록 수도(pseudo code)코드로 나타낼 수 있다.
- [0062] 또한, 도 4c에 도시된 바와 같이, JAVA CODE를 이용하여, 상기 터치이벤트가 발생한 경우, 'shell' 명령을 통해 프로세스에 매크로에 해당하는 어플리케이션 네임정보가 있는지 확인 한 후, 'Monkey'라는 매크로 어플리케이션 네임정보가 있는 경우, 사용자에게 Toast message를 통해, 매크로 탐지가 됐음을 알리도록 구현될 수 있다.
- [0064] 그 다음으로, 상기 터치이벤트가 검출된 경우, 상기 터치이벤트에 대응하는 터치좌표정보를 검출하며(S150, S250), 상기 터치좌표정보와 먼저 발생된 터치이벤트에 대응하는 전체의 터치좌표정보를 비교하여, 상기 터치좌표정보의 중복률을 산출할 수 있다.
- [0065]
- [0066] 그리고, 상기 터치좌표정보의 중복률과 기설정된 기준률을 비교하여, 상기 터치좌표정보의 중복률이 상기 기준률 이상인 경우, 상기 터치이벤트가 매크로에 의해 발생한 것으로 판단한다(S160). 여기에서, 상기 중복률은 10%가 될 수 있으나, 이에 한정되는 것은 아니다.
- [0068] 즉, 도 5a에 도시된 바와 같이, 실제 사용자와 매크로가 특정게임을 하면서 발생시킨 터치이벤트의 좌표값을 점으로 나타낸 것이며, 게임에서 사용하는 버튼과 동일한 크기의 버튼을 500회 터치하였다. 특정 버튼을 누르기 위해 실제 사용자는 여러 좌표를 터치한 반면, 매크로는 하나의 좌표만을 터치했다.
- [0069] 도 5b에 도시된 바와 같이, 게임에서 사용하는 버튼과 동일한 크기의 버튼을 500회 터치하는 실험을 10번 반복한 뒤의 평균은 한 버튼을 500회 터치할 경우, 전체의 터치좌표에 대해 약 10%가 중복되었고(a), 이는 터치횟수에 비례했다.
- [0071] 또한, 본 발명에 따른 다른 실시예는 앞서 설명한 일실시에 중 사용자로부터 입력된 터치이벤트를 검출하는 터치이벤트 검출하는 제 1 단계, 상기 터치이벤트로부터 산출된 터치면적정보 또는 터치압력정보를 기설정된 기준값과 비교하여, 상기 터치이벤트의 매크로에 의한 발생여부를 판단하는 제 2 단계 및 상기 터치이벤트가 검출된 경우, 기실행되고 있는 어플리케이션의 네임정보와 기설정된 매크로 어플리케이션의 네임정보를 비교하여, 상기 터치이벤트의 매크로 사용여부를 판단하는 제 3 단계를 포함하며, 이에 대한 설명은 생략한다.
- [0073] 나아가, 상기 터치이벤트가 검출된 경우, 상기 터치이벤트에 대응하는 터치좌표정보를 검출하며, 상기 터치좌표정보와 먼저 발생된 터치이벤트에 대응하는 전체의 터치좌표정보를 비교하고, 상기 전체의 터치좌표정보 중 상기 터치좌표정보의 중복된 발생횟수를 카운트를 한다(S260).
- [0075] 그리고, 상기 터치좌표정보의 발생횟수가 기설정된 기준횟수 이상인 경우, 상기 터치이벤트가 매크로에 의해 발생한 것으로 판단하며(S270), 매크로 탐지를 단순히 터치횟수를 기준으로 잡았기 때문에 터치좌표정보의 중복률을 계산하는 프로세스보다 모바일 기기의 자원을 덜 소모할 수 있다.

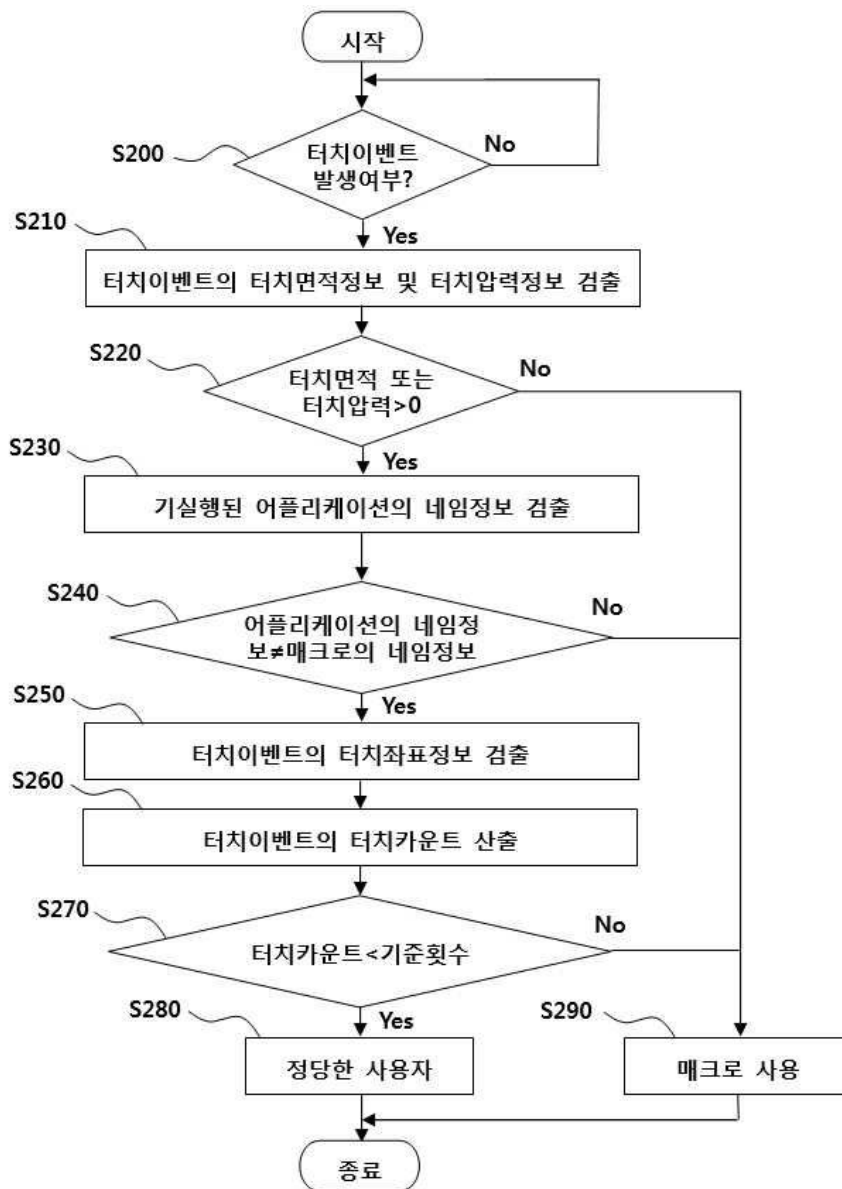
- [0077] 여기에서, 상기 기준횟수는 각 어플리케이션마다 동일한 값을 잡을 수 없기 때문에 각각 사용자가 평균적으로 어느 정도의 플레이 시간이 있는지 확인해야 하며, 플레이 시간동안 각 좌표마다 최대 카운트가 얼마였는지 통계치를 추산해서 정해야 한다.
- [0079] 예를들면, 도 5c에 도시된 바와 같이, 상기 터치이벤트의 발생횟수 카운트가 '30'을 초과하는 경우, 'Macro detection'이라는 문구를 출력하도록 수도(pseudo code)코드를 이용하여, 구현될 수 있으며, 도 5d에 도시된 바와 같이, Java code를 이용하여, 상기 터치이벤트의 발생횟수 카운트가 '2'를 초과하는 경우, Toast message 를 통해, 매크로가 탐지됨을 알리도록 구현될 수 있으나, 이에 한정되는 것은 아니다.
- [0081] 상기에서 살펴본 바와 같이, 사용자의 터치이벤트에 기초한 터치좌표정보의 중복률 또는 발생횟수의 카운트 값을 이용한 방법은 진술한 터치면적정보 등을 이용한 방법과 어플리케이션 네임정보를 이용한 방법이 애플레이터에 의한 매트릭스 사용과 윈도우 응용 매크로 사용을 탐지 못하는 단점을 보완한 것으로서, 다양한 매크로 사용에 대해 보다 효율적으로 탐지할 수 있다.
- [0083] 또한, 본 발명에 의한 매크로 탐지방법은 태블릿 PC, 노트북(notebook), 넷-북(net-book), e-리더(e-reader), PDA(personal digital assistant), PMP(portable multimedia player), MP3 플레이어 또는 MP4 플레이어와 같은 데이터 처리 장치로 구현되거나, 모바일폰(mobile phone), 스마트폰(smart phone)과 같은 핸드헬드 장치(handheld device)에서 수행될 수 있다.
- [0085] 이상 본 발명을 구체적인 실시예를 통하여 상세히 설명하였으나, 이는 본 발명을 구체적으로 설명하기 위한 것으로, 본 발명에 따른 안드로이드 OS 기반의 매크로 탐지방법은 이에 한정되지 않으며, 본 발명의 기술적 사상 내에서 당해 분야의 통상의 지식을 가진 자에 의해 그 변형이나 개량이 가능함은 명백하다고 할 것이다.
- [0086] 본 발명의 단순한 변형 내지 변경은 모두 본 발명의 영역에 속하는 것으로 본 발명의 구체적인 보호범위는 첨부된 특허청구범위에 의하여 명확해질 것이다.

도면

도면1



도면2



도면3a

```

if event.Pressure is 0 then
  print "Macro detection"
end if
  
```

도면3b

```

@Override
public boolean onTouchEvent(MotionEvent event) {
    super.onTouchEvent(event);

    if (event.getPressure() == 0) {
        Toast("압력값으로 매크로 탐지");
    }
    return false;
}

private void Toast(String msg) {
    Toast.makeText(MainActivity.this, msg, Toast.LENGTH_SHORT).show();
}

```

도면4a

```

shell@ciskt:/ $ ps
USER      PID     PPID  VSZ   RSS   WCHAN    PC      NAME
root        1         0  1244   428   ffffffff 00000000 S /init
root        2         0      0      0   ffffffff 00000000 S kthreadd
root        3         2      0      0   ffffffff 00000000 S ksoftirqd/0
root        5         2      0      0   ffffffff 00000000 S kworker/u:0
root        6         2      0      0   ffffffff 00000000 S migration/0
root        7         2      0      0   ffffffff 00000000 S watchdog/0
root       20         2      0      0   ffffffff 00000000 S khelper
root       21         2      0      0   ffffffff 00000000 S kworker/u:1
root       27         2      0      0   ffffffff 00000000 S sync_system_wor
root      429         2      0      0   ffffffff 00000000 S sync_supers
root      431         2      0      0   ffffffff 00000000 S bdi-default
root      433         2      0      0   ffffffff 00000000 S kblockd
root      452         2      0      0   ffffffff 00000000 S khubd
root      470         2      0      0   ffffffff 00000000 S irq/365-max7769
root      491         2      0      0   ffffffff 00000000 S irq/359-max7768
root      535         2      0      0   ffffffff 00000000 S cfg80211
root      623         2      0      0   ffffffff 00000000 S kpegasusq
root      657         2      0      0   ffffffff 00000000 S khungtaskd
root      658         2      0      0   ffffffff 00000000 S kswapd0
root      710         2      0      0   ffffffff 00000000 S fsnotify_mark
root      730         2      0      0   ffffffff 00000000 S ecryptfs-kthrea
root      738         2      0      0   ffffffff 00000000 S crypto
root      756         2      0      0   ffffffff 00000000 S s3c-fb
root      762         2      0      0   ffffffff 00000000 S s3c-fb-vsync

```

도면4b

```

val process
val result
process is "application's process name" //example name = "commands.monkey"
result <- execute shell process
if result is not "" then
    print "Macro detection"
end if

```

도면4c

```

@Override
public boolean onTouchEvent(MotionEvent event) {
    super.onTouchEvent(event);

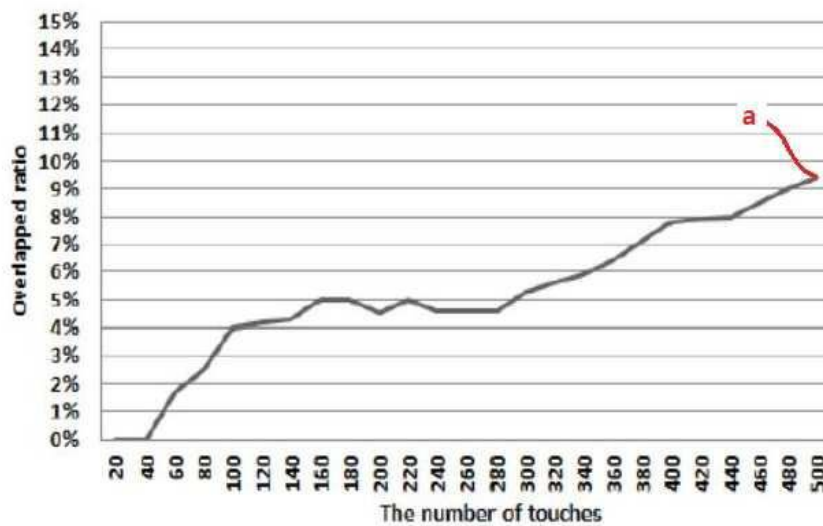
    ProcessDetection();

    return false;
}

private void ProcessDetection(){
    Runtime runtime = Runtime.getRuntime();
    Process process;
    try {
        process = runtime.exec("ps name commands.monkey");
        BufferedReader br = new BufferedReader((new InputStreamReader(process.getInputStream())));
        int i = 0;
        String line;
        String[] readline = new String[2];
        while ((line = br.readLine()) != null) {
            readline[i] = line;
            i++;
        }
        String[] words = readline[i].split(" ");
        if (words[1].toString() != "") {

```

도면5a



도면5b



정당사용자



매크로 사용

도면5c

```

val query
val result
query is "insert into table
        value(
            coordinate = event.coordinate
execute query
query is "select count(*)
        from table
        where coordinate = event.coordinate
        group by coordinate"
result <- execute query
if result > 30 then
    print "Macro detection"
end if
    
```

도면5d

```

    super.onTouchEvent(event);

    int x = (int) event.getX();
    int y = (int) event.getY();

    String coordi = String.valueOf(x) + String.valueOf(y);

    insertData(coordi);
    queryData(coordi);

    return false;
}

ivate void Toast(String msg) {
    Toast.makeText(MainActivity.this, msg, Toast.LENGTH_SHORT).show();

ivate void insertData(String coordi) {
    String sql = "insert into " + tableName + "(coordi) values('" + coordi + "')";
    database.execSQL(sql);

ivate void queryData(String coordi) {
    String sql = "select count(*) from " + tableName + " where coordi = '" + coordi + "'";
    Cursor result = database.rawQuery(sql, null);
    if (result.moveToFirst()) {
        int count = result.getInt(0);
        if (count > 2){
            Toast("카운트 값으로 매크로 탐지");
            //2는 임의로 정한 카운트 횟수

```