SHAKE RESPONSIVE PORTABLE COMPUTING DEVICE FOR SIMULATING A RANDOMIZATION OBJECT USED IN A GAME OF CHANCE

Inventor: Louis B. Rosenberg, Arroyo Grande, CA (US)

Correspondence Address:
LAW OFFICE OF PHILIP A STEINER
846 HIGUERA STREET
SUITE 4
SAN LUIS OBISPO, CA 93401 (US)

Assignee: Outland Research, LLC, Pismo Beach, CA (US)

Appl. No.: 11/563,610
Filed: Nov. 27, 2006

Related U.S. Application Data
Continuation-in-part of application No. 11/539,598, filed on Oct. 6, 2006.
Continuation-in-part of application No. 11/555,784, filed on Nov. 2, 2006.

Provisional application No. 60/750,490, filed on Dec. 15, 2005.

Publication Classification
Int. Cl.
A63B 7/00 (2006.01)
U.S. Cl. .................................................. 273/138.1

ABSTRACT
An apparatus, method and computer program product for a shake responsive portable computing device used in simulating a randomization object for playing a game of chance. The portable computing device includes a processor, a motion sensor functionally coupled to the processor and configured to detect shaking motions imparted on the portable computing device by a user, a display coupled to the processor, a memory coupled to the processor and a simulation program operatively loaded into the memory. The simulation program includes instructions executable by the processor to identify deliberately imparted characteristic shaking motions from the detected shaking motions, generate a randomized gaming result in dependence on the identified characteristic shaking motion signals and output the randomized gaming result to the display as a graphical representation of a randomization object.
FIG. 2
FIG. 4
Start 500

Couple motion sensor to a processor 505

Configure motion sensor to detect manual shaking motions (MSM) imparted by a user 515

Operatively load simulation program in memory coupled to the processor 520

Read motion sensor 530

MSM Detect

No 535

ID Characteristic shake event (CSM) 540

Output (dynamically) randomized gaming result as a graphical representation of a gaming instrument 550

Update player status 560

Yes

PDA, Cellphone, Portable media player, Handheld game player 510

Setup, Configuration, Gaming instrument selection 525

Time, Number of events, Magnitude, Freq 545

One or more dice 555

Win, Loss, Draw 565

FIG. 5
SHAKE RESPONSIVE PORTABLE COMPUTING DEVICE FOR SIMULATING A RANDOMIZATION OBJECT USED IN A GAME OF CHANCE

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a non-provisional application claiming benefit and priority under 35 U.S.C. § 119(e) from co-pending U.S. provisional application Ser. No. 60/750,490 filed on Dec. 15, 2005 to a common assignee and the instant inventor;

[0002] this application is also a continuation-in-part of co-pending U.S. patent application Ser. Nos. 11/539,598 filed on Oct. 6, 2006 and 11/555,784 filed on Nov. 2, 2006 also to the common assignee and inventor;

[0003] all of the aforementioned patents applications are hereby incorporated by reference in their entirety as if fully set forth herein.

FEDERALLY SPONSORED RESEARCH AND DEVELOPMENT

[0004] Not Applicable

REFERENCE TO A MICROFICHE APPENDIX

[0005] Not Applicable

FIELD OF INVENTION

[0006] The present inventive embodiments relates generally to portable computing devices and more specifically to shake-responsive portable computing devices for simulating the shaking and tossing of a randomization object used in a game of chance.

BACKGROUND

[0007] As the size and cost of processor driven devices have declined in recent years, portable computing device have become popular consumer products due to their highly portable nature and interconnectivity with existing computer networks. For example, personal digital assistant (PDAs), portable gaming systems, cellular telephones, and portable media players are all examples of common portable computing device that have gained widespread adoption over recent years. In fact, a large percentage of people in America have at least one portable computing device on their person when they leave their home and generally use such devices for a wide range of applications. For example, cellular telephones have come to take on additional computing functions such as gaming, scheduling, picture taking, and media playing.

[0008] In the relevant art, portable computing devices enable users to play games of chance, for example games that require a user to roll simulated dice and make moves based upon the outcome of the dice roll. For example, “,” is a popular dice game offered for Cingular mobile phones that is based on one of the most popular casino games in Asia called Sic Bo. Another common dice game for Motorola and Nokia mobile phones is “Baccarat,” a game in which players place bets and press a button to toss the simulated dice. In general, such dice games work by automatically rolling the dice for the user automatically or based upon a simple button press, which differs significantly from the real world experience in dice rolling. In the real world, there is a causal relationship between physically tossing dice and observing the outcome that has made dice games so popular over the centuries.

[0009] Unfortunately the modern instantiation of dice games upon portable computing device have removed this physically causal relationship and reduced the appeal of dice games.

[0010] Unless otherwise indicated herein, the approaches described in this section are not prior art to the claims in this application and are not admitted to be prior art by inclusion in this section.

SUMMARY

[0011] The various exemplary embodiments described herein address the limitations in the relevant art and provides mechanisms which allows a user to physically utilize a portable computing device to simulate the shaking and tossing of a randomization object for playing a game of chance. In the real world, when a container of some physical substance needs to be mixed, it is common human action to pick up the container and shake it. As examples, a can of spray paint, salad dressing, orange juice, and other consumer products that often separate into layers requiring agitation to become mixed. Thus shaking is a common metaphor that used to mix or randomize the distribution of the contents within a container. The various inventive embodiments, leverages this metaphor and establishes a unique user interface for a portable computing device in which a user can simulate the physical motions used in playing games of chance in which an actual physically shaking action is performed to initiate the randomization process and result in a randomized outcome.

[0012] In an exemplary apparatus embodiment, a shake responsive portable computing device for simulating a randomization object used in a game of chance is provided. This exemplary apparatus embodiment comprises a processor, a motion sensor functionally coupled to the processor and configured to detect shaking motions imparted on the portable computing device by a user, a display coupled to the processor, a memory coupled to the processor and a simulation program operatively loaded into the memory.

[0013] The simulation program includes instructions executable by the processor to identify deliberately imparted characteristic shaking motions from the detected shaking motions, generate a randomized gaming result in dependence on the identified characteristic shaking motion signals and output the randomized gaming result to the display as a graphical representation of the randomization object.

[0014] In a first related exemplary apparatus embodiment, the randomization object comprises one or more simulated dice, and the simulation program further includes instructions to simulate a tumbling movement of the one or more simulated dice in dependence on the identified characteristic shaking motions.

[0015] In a second related exemplary apparatus embodiment, the simulation program further includes instructions to simulate the tumbling movement of the one or more dice in approximate coordination with the identified characteristic shaking motions.
In a third related exemplary apparatus embodiment, the simulation program further includes instructions to dynamically depict the tumbling movement of the one or more simulated dice coming to rest to reveal the generated gaming result.

In a fourth related exemplary apparatus embodiment, the display is encompassed by a housing in common with the processor.

In a fifth related exemplary apparatus embodiment, the display is associated with a separate processor in processing communications over a wireless link with the processor.

In a sixth related exemplary apparatus embodiment, the tumbling movement of the simulated dice come to rest in response to a detected cessation of the identified characteristic shaking motions.

In a seventh related exemplary apparatus embodiment, the portable computing device is one of, a personal data assistant, a cellular telephone, a portable media player, and a handheld game player.

In an eighth related exemplary apparatus embodiment, the generated gaming result is one of; a number, a word, a sentence, an image, a color and any combination thereof.

In a ninth related exemplary apparatus embodiment, the generated gaming result is used to update a status of play in the game of chance.

In a tenth related exemplary apparatus embodiment, the generated gaming result is dependent upon at least one dynamic characteristic of the identified characteristic shaking motion.

In an eleventh related exemplary apparatus embodiment, the dynamic characteristic of the identified characteristic shaking motion is one of; a time duration, a magnitude, a number of distinct shake events, a frequency of distinct shake events and any combination thereof.

In an exemplary method embodiment, a method for providing a shake responsive portable computing device for simulating a randomization object used in a game of chance is provided. This exemplary method embodiment comprises coupling a motion sensor to a processor associated with the portable computing device, configuring the motion sensor to detect shaking motions imparted on the portable computing device by a user, and loading a simulation program into a memory coupled to the processor.

The simulation program includes instructions executable by the processor for identifying deliberately imparted characteristic shaking motions from detected shaking motions, generating a randomized gaming result in dependence on the identified characteristic shaking motion signals and outputting the randomized gaming result to a display coupled to the processor as a graphical representation of the randomization object.

In a first related exemplary method embodiment, further instructions executable by the processor are provided to simulate a tumbling movement of one or more simulated dice in dependence on the identified characteristic shaking motions.

In a second related exemplary method embodiment, the simulation program further includes instructions executable by the processor for simulating the tumbling movement of the one or more dice in approximate coordination with the identified characteristic shaking motions.

In a third related exemplary method embodiment, the simulation program further includes instructions executable by the processor for dynamically depicting the tumbling movement of the one or more simulated dice coming to rest to reveal the gaming result.

In a fourth related exemplary method embodiment, the simulation program further includes instructions executable by the processor for generating the randomized gaming result in dependence upon one or more dynamic characteristics of the identified characteristic shaking motions selected from the group consisting essentially of, a time duration, a magnitude, a number of distinct shake events, a frequency of the distinct shake events, and any combination thereof.

In a fifth related exemplary method embodiment, the tumbling movement of the simulated dice coming to rest in response to a detected cessation of the identified characteristic shaking motions.

In a sixth related exemplary method embodiment, the display is associated with a separate processor in processing communications over a wireless link with the processor.

In an exemplary computer program product (CPP) embodiment, a CPP embodied in a tangible form comprising instructions executable by a processor associated with a shake responsive portable computing device to receive shaking motion signals from an operatively coupled motion sensor, identify deliberately imparted characteristic shaking motions from the detected shaking motions signals, generate a randomized gaming result in dependence on the identified characteristic shaking motion signals and output the randomized gaming result on an operatively coupled display as a graphical representation of a randomization object.

In a first related exemplary CPP embodiment, further instructions executable by the processor are provided to simulate a tumbling movement of one or more simulated dice in dependence on the identified characteristic shaking motions.

In a second related exemplary CPP embodiment, further instructions executable by the processor are provided for simulating the tumbling movement of the one or more dice in approximate coordination with the identified characteristic shaking motions.

In a third related exemplary CPP embodiment, further instructions executable by the processor are provided to dynamically depict the tumbling movement of the one or more dice coming to rest to reveal the gaming result.

In a fourth related exemplary CPP embodiment, the tangible form may be one or more of, a logical media, a magnetic media and an optical media.

The various exemplary apparatus, methodic and computer program product embodiments described above are provided in related exemplary numeric embodiments for
convenience only. No limitation to the various exemplary embodiments disclosed is intended.

BRIEF DESCRIPTION OF DRAWINGS

[0039] The features and advantages of the various exemplary embodiments will become apparent from the following detailed description when considered in conjunction with the accompanying drawings. Where possible, the same reference numerals and characters are used to denote like features, elements, components or portions of the various inventive embodiments. It is intended that changes and modifications can be made to the described embodiment without departing from the true scope and spirit of the subject inventive embodiments as generally defined by the claims.

[0040] FIG. 1—depicts a generalized and exemplary block diagram of a portable computing device.

[0041] FIG. 2—depicts an exemplary signal trace of a motion sensor.

[0042] FIG. 3—depicts a generalized and exemplary block diagram of an embodiment of the portable computing device.

[0043] FIG. 4—depicts an alternate embodiment where the portable computing device is in processing communications with a second computing device over a wireless communications link.

[0044] FIG. 5—depicts an exemplary process flow chart of the various exemplary embodiments.

DETAILED DESCRIPTION

[0045] An apparatus, method and computer program product are described herein for a shake responsive portable computing device. The portable computing device is configured to simulate a randomization object for use in playing a game of chance. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the various inventive embodiments. It will be apparent to one skilled in the art that the present inventive embodiments may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the inventive embodiments.

[0046] The simulation of the randomization object is performed in response to a user imparting a physical shaking motion upon a suitably equipped portable computing device. Various embodiments provide for the dynamic simulation of the shaking of the randomization object until a randomized result is obtained and visually outputted to a user. The various embodiments utilize one or more parameters to simulate the shaking and/or coming to rest of the simulated randomization object including a magnitude, duration, and/or time-varying profile of the shaking activity imparted by the user upon the portable computing device. Where necessary, computer programs, algorithms and routines are envisioned to be programmed in a high level language object oriented language, for example Java (™) C++, C#, CORBA or Visual Basic (™).

[0047] Referring to FIG. 1, a generalized block diagram of a portable computing device 100 is depicted. The portable computing device 100 includes a communications infrastructure 90 used to transfer data, memory addresses where data items are to be found and control signals among the various components and subsystems associated with the portable computing device 100. A processor 5 is provided to interpret and execute logical instructions stored in the main memory 10. The main memory 10 is the primary general purpose storage area for instructions and data to be processed by the processor 5. The main memory 10 is used in its broadest sense and includes RAM, EEPROM and ROM. A timing circuit 15 is provided to coordinate activities within the portable computing device 100 in near real time and to make time-based assessments of sensor data collected by intrinsic or extrinsic sensors 75 coupled to the portable computing device 100. The processor 5, main memory 10 and timing circuit 15 are directly coupled to the communications infrastructure 90.

[0048] A display interface 20 is provided to drive a display 25 associated with the portable computing device 100. The display interface 20 is electrically coupled to the communications infrastructure 90 and provides signals to the display 25 for visually outputting both graphics and alphanumeric characters. In an embodiment, the display 25 visually outputs a graphical representation of a simulated randomization object 305 (FIG. 3.) The display 25 may also output gaming applications that utilize the simulated randomization objects 305 for play. The display 25 may also be used to visually output various controls and/or menus for interacting with the software and/or firmware installed on the portable computing device 100. In another embodiment, the display interface 20 may be used to drive a remote display 25A over a wireless network 85. This arrangement allows networked gaming applications to be communicated to other computing devices over the wireless network 85. In this way, a user may utilize the portable computing device 100 as a remote randomization object peripheral coupled to another computer via the wireless network 85.

[0049] The display interface 20 may include a dedicated graphics processor and memory to support the displaying of graphics intensive media. The display 25 may be of any type (e.g., cathode ray tube, gas plasma) but in most circumstances will usually be a solid state device such as liquid crystal display (LCD) or polymeric display.

[0050] A secondary memory subsystem 30 is provided which houses retrievable storage units such as a hard disk drive 35, a removable storage drive 40, an optional logical media storage drive 45 and an optional removal storage unit 50. One skilled in the art will appreciate that the hard drive 35 may be replaced with flash memory. The secondary memory 30 may be used to store a plurality of media files, gaming applications, and/or other files, programs, or data. The removable storage drive 40 may be a replaceable hard drive, optical media storage drive or a solid state flash RAM device. The logical media storage drive 45 may include a flash RAM device, an EEPROM encoded with playable media, or optical storage media (CD, DVD). The removable storage unit 50 may be logical, optical or of an electromechanical (hard disk) design.

[0051] A communications interface 55 subsystem is provided which allows for standardized electrical connection of peripheral devices to the communications infrastructure 90 including, serial, parallel, USB, and Firewire (™) connec-
tivity. For example, a user interface 60 and a transceiver 65 are electrically coupled to the communications infrastructure 90 via the communications interface 55. For purposes of this specification, the term user interface 60 includes the hardware and operating software by which a user interacts with the portable computing device 100 and the means by which the portable computing device conveys information to the user and may include the display 25.

[0052] The transceiver 65 facilitates the remote exchange of data and synchronizing signals between the portable computing device 100 and other devices in processing communications 85 with the portable computing device 100. The transceiver 65 is envisioned to be of a radio frequency type normally associated with computer networks for example, wireless computer networks based on BlueTooth (TM) or the various IEEE standards 802.11 x, where x denotes the various present and evolving wireless computing standards, for example 802.11N, WiMax 802.16 and WRANG 802.22.

[0053] Alternately, digital cellular communications formats compatible with for example GSM, 3G and evolving cellular communications standards. Both peer-to-peer (PPP) and client-server models are envisioned for implementation of the various inventive embodiments. In a third alternative embodiment, the transceiver 65 may include hybrids of computer communications standards, cellular standards and evolving satellite radio standards.

[0054] The user interface 60 employed on the portable media play 100 may include a pointing device (not shown) such as a mouse, thumbwheel or track ball, an optional touch screen (not shown); one or more push-button switches 60A, 60B; one or more sliding or circular potentiometer controls (not shown), one or more switches (not shown), and/or one or more tactile feedback units (not shown). The user interface 60 provides interrupt signals to the processor 5 that may be used to interpret user interactions with the portable computing device 100 and may be used in conjunction with the display 25.

[0055] An optional tactile feedback unit 80 may be used to simulate tactile sensations corresponding to a user’s interaction with a simulated randomization object 305 to further increase the realism of the simulated randomization object experience.

[0056] The user interface 60 further includes a specialized shake interface 70. The shake interface 70 is configured to detect if and when a user imparts characteristic shake motions (CSM) 205 (FIG. 2) upon the portable computing device 100 by shaking the portable computing device 100 in his or her hand. The motion sensor(s) 75 are configured to detect when a user deliberately imparts a manual shaking motion upon the portable computing device 100 by shaking the device back and forth and/or up and down in his or her hand. Various manual shaking motions may be performed which allows for differentiation of the type of interaction with the portable computing device 100 the user is seeking. The manual shaking motions are performed in a manner roughly equivalent to how a user would shake a bottle of salad dressing to mix the bottle’s contents. Analogously, a user may simulate a roll of the dice by shaking the portable computing device 100. More specifically, a user may simulate dice 305 (FIG. 3) to be shaken by physically shaking the portable computing device 100 in a characteristic way and may cause simulated dice to be tossed and tumble to rest by ceasing to shake the portable computing device 100 in the characteristic way.

[0057] The shake interface 70 generally includes one or more motion sensors 75 coupled thereto for detecting a shaking action imparted by the user upon the portable computing device 100. The shake interface 70 may include analog to digital conversion circuitry, signal conditioning circuitry, and/or anti-coincidence circuitry. A variety of motion sensors 75 may be employed to detect characteristic shaking motions imparted by the hand of a user upon the casing of a portable computing device 100.

[0058] The most common motion sensor(s) 75 to be used are typically accelerometers. Accelerometers are operative to detect accelerations imparted by the user upon the portable computing device 100. An accelerometer 75 may be oriented to detect accelerations in one or more degrees of freedom 365 (FIG. 3). However, for cost, size and power consumption considerations, a miniature accelerometer is preferred. For example, Analog Devices, Inc. manufactures many integrated Micro Electro Mechanical System (MEMS) accelerometers suitable for the various exemplary embodiments described herein. Information regarding the MEMS accelerometers may be downloaded from the URL www.analog.com.

[0059] One or more accelerometers 75 may be oriented to detect accelerations in one or more degrees of freedom (axes). Alternately, a multi-axis accelerometer 75 may be used. In a simple embodiment, a single axis accelerometer 75 is employed; the axis of detection of the accelerometer being preferably oriented along a long axis of the portable computing device 100. In this way, the accelerometer 75 is configured to detect motions imparted upon the portable computing device 100 by a user shaking it up and down generally in parallel to the long axis of the portable computing device 100.

[0060] The motion sensor(s) 75 are coupled to the sensor interface 70 which allows the one or more sensors 75 to be operatively coupled to the communications infrastructure 90. The sensor interface 70 may monitor interactions with the user interface 60. For example, the sensor interface 70 may be used to monitor a user’s interaction with the one or more push-button switches 60A, 60B. An interrupt circuit may be incorporated into the hardware supporting the communications infrastructure 90. The motion sensor(s) 75 is generally installed within the case embodying the electronics which comprise the portable computing device 100.

[0061] An audio processing subsystem 65 is electrically coupled to the communications infrastructure 90 and provides for the play of sound effects, background music, and the playback and recording of digital media, for example, multi or multimedia encoded in any of the exemplary formats MP3, AVI, WAV, MPG, QT, WMA, AIFF, AU, RAM, RA, MOV, MIDI, etc. The audio processing subsystem 65 may include a microphone input port 65A for input of voice commands and a headphone, headset, ear buds or speaker output 65B. Connection of the microphone 65A and/or headphones 65B includes both traditional cable and wireless arrangements such as BlueTooth(TM) are known in the relevant art.

[0062] As referred to in this specification, “media” refers to video, audio, streaming and any combination thereof. The
audio processing subsystem 65 may also be configured to generate and/or play back sound effects which simulates the sounds produced using a particular randomization object 305. The sound effects are generally output to the audio device 100 in order to enhance the overall gaming experience. In addition, the audio processing subsystem 65 is envisioned to optionally include features such as graphic equalization, volume, balance, fading, base and treble controls, surround sound emulation, and noise reduction. One skilled in the art will appreciate that the above cited list of file formats is not intended to be all inclusive.

[0063] The portable computing device 100 includes an operating system, the necessary hardware and software drivers necessary to fully utilize the devices coupled to the communications infrastructure 90 and at least one simulation program 520 (FIG. 5) loaded into the main memory 10. The simulation program 520 may perform multiple functions, for example performing dice rolling simulations 305 (FIG. 3) in response to characteristic shaking motions 205 imparted by the user.

[0064] The simulation program 520 may also control one or more gaming applications executing upon the portable computing device 100 which utilize the randomized results of the simulated randomization object 305. The simulation program 520 may also communicate the results of the simulated randomization object 305 to one or more other computing devices 400 (FIG. 4) over the communications network 85. The simulation program 520 provides the user with a configuration menu by which he or she may select the type, number and associated parameters of a randomization object 305 to be played. For example, the number of dice to be rolled, the number of faces on each dice to be rolled, and the selectable items upon each face of the dice to be rolled.

[0065] In another embodiment, the simulation program 520 may also provide a menu which allows the user to select displays images, sounds, and/or tactile feedback corresponding to the simulated randomization object 305 and/or games to be played by the portable computing device 100. For example, the simulation program 520 may be programmed to monitor the shake interface 70 by reading the associated motion sensor(s) 75 and storing data from the motion sensors 75 in memory 30 over time. The simulation program 520 may also read data from the timing circuit 15. The simulation program 520 processes the time varying profile of the sensor data from the motion sensor(s) 75 and determines based upon the time varying characteristics of the sensor data whether or not the user has imparted a characteristic shaking action upon the portable computing device 100. In addition, the simulation program 520 may process the time varying profile of the sensor data to determine based upon the time varying characteristics of the sensor data whether or not a previously imparted characteristic shaking action has been ceased by the user.

[0066] References to the simulation program 520 may be made in both singular and plural form. No limitation is intended by such grammatical usage as one skilled in the art will appreciate that multiple programs, objects, subprograms, routines, algorithms, applets, contexts, etc. may be implemented programmatically to implement the inventive embodiments.

[0067] FIG. 2 provides an exemplary signal trace 230 of an accelerometer arranged as a motion sensor 75 suitable for use in the various envisioned embodiments. This particular type of motion sensor 75 provides an analog voltage output signal 230 which varies as a function of the acceleration(s) that the motion sensor 75 is experiencing. For this particular type of motion sensor 75 (accelerometer), a steady state voltage signal of 2000 millivolts (mV) is output from the motion sensor 75 when no accelerations are being detected. When undergoing either a positive or negative acceleration, the voltage output from the accelerometer increases or decreases as functions of the magnitude M205 being experienced by the portable computing device 100. When sampled over a time domain t 210, various motion dependent parameters may be determined which are useful for simulating a randomization object 305.

[0068] In general, deliberate manual shaking motions may be identified from the time-varying motion sensor data 230 based on at least in part upon the presence of a roughly cyclic signal profile that falls within certain bounds and/or exceeds certain limits. Such bounds and/or limits may include frequency parameters f 215, magnitude parameters M 205, count parameters C 220, and/or timing parameters t 210. As shown by the time-varying motion sensor signal 230, a characteristic shaking motion includes a set of repeated acceleration impulses N, N2, N3220,A,B,C each corresponding with a single manual shake motion imparted by the user. Because the manual shaking motions imparted by the user will generally include a plurality of shake impulses N, N2, N3220,A,B,C, each being imparted such that it exceeds a certain magnitude thresholds M+, M−205,A,B and collectively being imparted within certain timing t1, 2210,A,B and frequency characteristics f1, 215,A,B, characteristic shake events E1, E2, E3225,A,B,C may be identified based upon the count impulses NA, N2, N3220,A,B,C, magnitude M+, −M−205,A,B, timing t1, 2210,A,B, and/or frequency of the acceleration impulses 11, 2215,A,B present in the motion sensor data.

[0069] More specifically, the characteristic shake events E1, E2, E3225,A,B,C may be identified based on the presence of impulse events that exceed certain magnitude limits, meet certain duration bounds, meet certain frequency bounds, meet certain temporal spacing requirements, and/or contain a count of impulses that falls within certain numerical bounds.

[0070] Thus, an analysis of the time-varying acceleration profile of motion data 230 may be performed by the simulation program 520 to determine if a deliberate manual shaking motion has been imparted by the user upon the portable computing device 100 based upon magnitude M205, frequency f1 2215,A,B, timing t1, 2210,A,B, and/or count characteristics NA, N2, N3220,A,B,C. In general, this analysis includes an identification of impulse events E1, E2, E3225,A,B,C and a determination as to whether the magnitude M205 of the impulse events, frequency of impulse events, temporal spacing of impulse events, and/or numerical count of impulse events meet certain bounds and/or exceed certain limits.

[0071] In some exemplary embodiments, the simulation program 520 is configured to identify a characteristic shake event 225 being imparted by a user when a portion of the time varying motion sensor signal 230 meets a certain magnitude M205 and/or temporal bounds Δ1 210,C. For example, a characteristic shake event E 225 may be identi-
fied when a portion of the time varying sensor signal 230 exceeds both a positive and negative magnitude threshold M+, M–\(\text{205A,B}\) within a certain time duration \(\Delta t\text{210C}\).

[0072] The ability to set a required threshold, timing, and/or sequence the simulation program 520 may be configured to determine that the user imparted a deliberate shaking action upon the portable computing device 100 and generate a specific randomization object 305 response specific to the detected event.

[0073] For example, a user simulating a pair of dice to be rolled, may provide three characteristic shake events E1, E2, E3,225A,B,C followed by a release characteristic shake event E4, 225D which provides a different acceleration profile indicating that the simulated dice have been released. Alternately the release may be triggered by the user ceasing to impart a new characteristic shake event upon the portable computing device 100 within a certain time interval of a previous characteristic shake event. The simulation program 520 generates a randomized result for the dice which is visually output to the user. The simulation program 520 may also be configured to output sound effects of the sound of shaking a pair of dice in response to the characteristic shake events E1, E2, E3,225A,B,C and a sound of the dice being released in response to the different characteristic shake of the release event E4, 225D (or in response to a determined cessation of the characteristic shaking motion.) The sound effects and visual images are intended to be output in near-real-time with the identified characteristic shake events E,225A-D to provide the user with a convincing illusion that he or she has actually shaken and rolled a pair of dice.

[0074] In an embodiment, a time duration \(\Delta t\text{210C}\), magnitude M\(\text{205}\), number of characteristic shake events N \(\text{220}\), and/or frequency of characteristic shake events \(\text{F215}\) may be used to simulate the time of flight, spin time and/or roll time before a randomized result is obtained for the simulated randomization objects 305. Additionally, the various motion dependent parameters may be determined in multiple geometric axes 325 (Fig. 3) which may be useful for discriminating against unintentional accelerations detected by the motion sensor 75 from being processed by the processor 5. In addition, ambient accelerations imparted due to the constant force of gravity and/or due to a moving reference frame of the user may be filtered from and/or otherwise differentiated from the intentionally imparted shaking motion signal 230 data using standard signal processing techniques.

[0075] With respect to the identification of a single characteristic shake event N1, N2, or N3\(\text{220A,B,C}\), such an event can generally be identified as a rapidly changing acceleration signal that exceeds a certain upper threshold M\(+\text{205A}\) and exceeds a certain lower threshold M\(-\text{205B}\), and accomplishes both within a certain small time period \(\Delta t\text{215C}\). The threshold values used for M\(+\), M–\(,\text{205A,B}\) and \(\Delta t\text{215C}\) may vary and are generally selected based upon the size and weight of the portable computing device 100 itself and the characteristics of the accelerometer employed. For the sensor 75 that generated the signal profile 230, the event duration \(\Delta t\text{210C}\) was defined as 100 milliseconds, although smaller time intervals may be used for greater impulse event discrimination. The upper threshold M\(+\text{205A}\) is approximately 3000 mV, the lower threshold M\(-\text{205B}\) is approximately 1000 mV. In some exemplary embodiments a duration \(\text{11,2210A,B}\) and/or \(\Delta t\text{210C}\) may be determined for a characteristic shake event E \(\text{225}\) the duration being the time during which the magnitude of the signal peaks exceeds one or more thresholds before decaying to a level that does not exceed the thresholds M\(+\), M–\(,\text{205A,B}\).

[0076] In some exemplary embodiments, a magnitude M\(\text{205}\) is determined for a characteristic shake event E \(\text{225}\), the magnitude M\(\text{205}\) being an average of the absolute value of the signal levels over the duration of the characteristic shake event E \(\text{225}\). Alternately, the magnitude M\(\text{205}\) may be determined from the difference between the maximum positive peak M\(+\text{205A}\) and maximum negative peak M–\(\text{205B}\) over the duration of the characteristic shake event E \(\text{225}\). In some exemplary embodiments the magnitude M\(\text{205}\) may be determined as the absolute value of the maximum peak detected over the duration of the characteristic shake event.

[0077] As the acceleration (motion) signals 230 are processed over time starting from t\(=0\) and extending forward along time axis \(\text{t210}\), there are no characteristic shaking events identified by the simulation program 520 until the time profile reaches approximately 1400 milliseconds in time \(\text{t210}\). At this time, a positive voltage transient exceeds the upper threshold M\(+\text{205A}\). This voltage transient corresponds with a user moving the portable computing device 100 with sufficient force to exceed the upper threshold M\(+\text{205A}\) of acceleration. A negative voltage transient which exceeds the lower threshold M–\(\text{205B}\) immediately following the positive voltage transient. This corresponds with the user moving the portable computing device 100 with sufficient force in the opposite direction to exceed the lower threshold M–\(\text{205B}\). Both thresholds are crossed within a time span of less than approximately 100 milliseconds (i.e. within the limits of the defined \(\Delta t\text{210C}\)). This means that the user imparted a sufficient acceleration upon the portable computing device 100 in a first direction to exceed the positive threshold M\(+\text{205A}\) and then immediately reversed direction and within the required time duration \(\Delta t\text{210C}\), imparted a sufficient acceleration upon the portable computing device 100 in the reverse direction to exceed the negative threshold M–\(\text{205B}\).

[0078] As such, the simulation program 520, may determine that the voltage transients exceeded both the positive M\(+\text{205A}\) and negative thresholds M–\(\text{205B}\) within certain predefined time duration \(\Delta t\text{210C}\) (e.g. 100 milliseconds), to elucidate that a characteristic shake event E \(\text{225}\) has occurred. The simulation program 520 then generates and outputs a simulated randomization object 305 in near-real-time relative to the detected shaking motions. By near real-time it is meant that to the user, it is perceived as approximately simultaneous with the imparted characteristic shake event(s) E \(\text{225}\). Ideally a time delay of less than 50 milliseconds after the completion of the shaking motions, for such a delay sufficiently small such that it is generally perceived as near real-time to a user.

[0079] The simulation program 520 may further process the data, identifying subsequent impulse events N2, N3, . . . . Nn, in a similar manner, and output simulations of randomization objects 305 and dynamic movements thereof. It should be noted that while positive threshold M\(+\text{205A}\) and negative threshold M–\(\text{205B}\) are shown in Fig. 2 to be equally distant from the nominal no-acceleration signal profile 230, this need not be the case in all embodiments. In
some exemplary embodiments, a larger threshold may be employed in one direction as compared to the opposite direction. In fact, this is often desired because a human user generally imparts greater acceleration upon an object he is shaking when extending arm muscles as compared to when flexing arm muscles during a reciprocating characteristic shaking motion.

[0080] In some exemplary embodiments, a sequence of characteristic shake events NJ, N2, and N3220A,B,C may be specifically required by the simulation program 520 in order for a user to indicate a particular intention to the portable computing device 100. For example, the simulation program 520 may be configured to determine that a characteristic shaking motion has been imparted by the user only if a particular sequence of characteristic shake impulses NJ, N2, N3220A,B,C are detected in the sensor signals 230. For example, in some exemplary embodiments, the portable computing device 100 may be configured such that the imparted impulses NJ, N2, N3220A,B,C are used to determine if the user is deliberately shaking the portable computing device 100; for example, if three or more characteristic single-shake acceleration impulses are detected in sequence within the acceleration data, each single-shake acceleration profile being separated by a certain minimum amount of time (for example 250 milliseconds) T1, T210A, B, and all three of the single-shake acceleration profiles being imparted upon the portable computing device 100 within a certain maximum amount of time (e.g., 2000 milliseconds), each of the three characteristic single-shake acceleration impulses NJ, N2, N3220A,B,C indicating a intentional user shake as a result of exceeding the positive M+205A and negative acceleration threshold M–205B.

[0081] In this way, deliberate shaking motions may be more easily distinguished from noise and/or unintentional movements of the portable computing device 100. For example, the portable computing device 100 may be configured such that randomization object simulations do not begin being output until at least three characteristic shake events E1, E2, E3225A,B,C are detected with certain timing parameters as described above. In this way, a user makes his or her intention clear to the portable computing device 100 that he or she is deliberately performing a gaming instrument randomization object simulation which avoids spurious movements from being detected when not intended by the user.

[0082] Referring to FIG. 3, an exemplary embodiment of a portable computing device 100 equipped with a motion sensor 75 is depicted. The various embodiments of the portable computing device 100 may be implemented in many common electronic devices including but not limited to a cellular telephone, a portable media player, a personal digital assistant (PDA) and a handheld game player (e.g., GameBoy™). The various inventive embodiments leverages a shaking metaphor and establishes a unique user interface for the portable computing device 100 in which a user can cause simulated gaming object to be shaken and outputted to a display 25 of the portable computing device 100 by physically shaking the portable computing device 100 in a characteristic manner.

[0083] In general, the portable computing device 100 is held in the palm of the user’s hand such that the display 25 is readily visible. For example, the user holds the portable computing device 100 in the palm of one hand, between the fingers and thumb, by grabbing the sides of the portable computing device 100 to provide control and support during the randomization object simulations.

[0084] The gaming simulations are initiated by the user imparting a shaking motion by shaking the portable computing device 100 back and forth generally along a defined axis or direction(s) 205, 365. The sensing axis of the motion sensor 75 is orientated to detect accelerations imparted by the user along the long axis 205 of the portable computing device 100. The data from the motion sensor 75 may be processed by the simulation program 520 executing in the portable computing device 100 to determine if it meets the characteristic event profile requirements described above. In an embodiment, an external port 355 is provided to operatively couple an external motion sensor 75 to the portable computing device 100. The external port 355 is electrically connected to the communications interface 55 and/or the shake interface 70.

[0085] In an embodiment, the simulated randomization objects 305 are visually outputted to the display 25 associated with the portable computing device 100. The simulated randomization objects 305 may be output dynamically 310 to simulate for example the rolling times for a pair of traditional dice, as shown, or the rolling times for various numbers of non-traditional dice with differing numbers of faces and facial content. The results of the simulated randomization object 330 may be outputted numerically 315; in text 320, as a phrase 325, graphically 305 and/or by color (not shown.)

[0086] The motion of the simulated randomization objects 305 may be performed in a number of ways. In an embodiment, the motion displayed to the user is computed over time based in part upon the shaking motions of the user, thereby reflecting the input of the user as he or she shakes the portable computing device 100. Such embodiments generally employ computer animation techniques for rendering the changing location and orientation of the randomization objects 305 over time.

[0087] In another embodiment, simulated randomization objects 305 are displayed using prerecorded video or animations that depicts the motion of the randomization objects 305 generally independent from the characteristic shaking motions of the user other than their being outputted coincidentally with the user’s shaking motions.

[0088] In another embodiment, the displayed simulated shaking motion is substantially coordinated in time with the user imparting the characteristic shaking motions, the displayed simulated release (i.e. dice toss) and the rolling to rest (i.e. dice tumbling) is substantially coordinated in time with the user ceasing to impart the characteristic shaking motions.

[0089] FIG. 4 depicts an alternate embodiment where the portable computing device 100 is configured as a remote peripheral of a second computing device 400. In an embodiment, the portable computing device 100 is processing communications with the second computing device 400 over a wireless communications link 85. A user begins the randomization object simulation by shaking the portable computing device 405 in a characteristic manner 205. The gaming simulation is then visually outputted on a display 25A associated with the second computing device 400.
In one embodiment, the simulation program 520 executes primarily on the portable computing device 100 and only the simulation of the randomization object 305 and gaming result 320 are output on the display 25. A associated with the second computing device 400. In an alternate embodiment, the gaming result is displayed upon both the portable computing device 100 and the second computing device 400. In an alternate embodiment the shaking motion of the simulated randomization object is displayed upon the portable computing device 100, the release and coming to rest of the simulated randomization object is displayed upon the second computing device 400. This creates the illusion to the user that the randomization object (e.g., dice) were first shaken in the hand of the user (as displayed upon the handheld screen of the user), and then were tossed and tumbled to rest onto the screen of the second computing device 400. This illusion is strengthened by substantially coordinating the simulated toss and coming to rest of the randomization object with the actual physical ceasing of the characteristic shaking motions by the user as described previously. One skilled in the art will appreciate that various other arrangements, distributions of processing and display may be accomplished between the portable computing device 100 and the second computing device 400.

Referring to FIG. 5, a process flow chart is depicted which provides an exemplary method for implementing the various inventive embodiments described herein. The process is initiated by coupling a motion sensor 75 to a processor 505 associated with a portable computing device 100. The portable computing device 100 may be a personal data assistant (PDA), cellular telephone, portable media player, and/or a handheld game player 510. The motion sensor 75 is configured to detect manual shaking motions (MSM) imparted on the portable computing device 100 by a user 515. A simulation program is then operatively loaded into a memory coupled to the processor 520. The simulation program includes various menus to allow the user to properly set up the portable computing device 100, configure the operating conditions of the simulation program 520 and select the randomization object of choice 555.

Once the portable computing device 100 has been properly configured 525, the simulation program 520 being reading signals generated by the motion sensor 530. If spurious motion signals are detected, the simulation program ignores the spurious motion signals and continues to monitor the motion sensor for deliberate manual shaking motions 535. If deliberate manual shaking motion signals are detected 535, for example as a sequence of deliberate shaking events, the simulation program 520 identifies any characteristic shake motions 540 based on preestablished parameters 545 entered during the set-up and configuration process 525.

The preestablished parameters may include time duration, number of events, magnitude of the events and frequency of the events 545. Any combination of these parameters may be used to identify the intended characteristic shake motion 540. While the characteristic shaking motion is in progress, the simulation program 520 simulates the shaking of the randomization objects. Once the characteristic shaking motion is determined to have ceased, the simulation program 520 simulates the release and coming to rest of the randomization objects. The simulation program 520 then outputs a randomized gaming result 320 as a graphical representation of a randomization object 305 to a display 25. Thus in related embodiments, the simulation program 520 outputs the randomization object 305 dynamically until the gaming result 320 is determined. In this way, for example, a user is given the illusion of shaking and then releasing a randomization object, for example dice, the gaming result seeming physically responsive to the user's physical motion of shaking and then ceasing to shake.

Once the gaming result 320 has been determined, the status of the player may be updated by the simulation program 560. The status of the player includes for example, advancing pieces in a simulated board-game by an amount prescribed by the gaming result, or winning, losing, drawing or other gaming outcomes 565, in a simulated game of chance. The gaming result 550 may take a variety of forms, for example numerical, textual, color-based, and/or pictorial. The simulation program 520 then resumes monitoring the motion sensor 75 for manual shaking motions 530 as described above. In this way the portable computing device may be manually shaken as a means of rolling dice in games that utilize traditional dice such as Backgammon, Monopoly (TM), Yahtzee (TM) and Craps, as well as games that utilize non-traditional dice with varying numbers of faces and varying facial content, such as Word Yahtzee (TM), Mickey Mouse Yahtzee (TM), and Dungeons-and-Dragons (TM).

The foregoing described exemplary embodiments are provided as illustrations and descriptions. They are not intended to limit an inventive scope to any precise form described. In particular, it is contemplated that functional implementation of the instrument accompaniment program described herein may be implemented equivalently in hardware, software, firmware, and/or other available functional components or building blocks. Other variations and embodiments are possible in light of above teachings, and it is not intended that this Detailed Description limit the scope of inventive embodiments, but rather by the Claims following herein.

What is claimed:

1. A shake responsive portable computing device for simulating a randomization object used in a game of chance comprising:
   a. a processor;
   b. a motion sensor functionally coupled to the processor and configured to detect shaking motions imparted on the portable computing device by a user;
   c. a display coupled to the processor;
   d. a memory coupled to the processor;
   e. a simulation program operatively loaded into the memory including instructions executable by the processor to:
      i. identify deliberately imparted characteristic shaking motions from the detected shaking motions;
      ii. generate a randomized gaming result in dependence on the identified characteristic shaking motion signals; and,
      iii. output the randomized gaming result to the display as a graphical representation of the randomization object.

2. The portable computing device according to claim 1 wherein the randomization object comprises one or more simulated dice and wherein the simulation program further
includes instructions to simulate a tumbling movement of the one or more simulated dice in dependence on the identified characteristic shaking motions.

3. The portable computing device according to claim 2 wherein the simulation program further includes instructions to simulate the tumbling movement of the one or more simulated dice in approximate coordination with the identified characteristic shaking motions.

4. The portable computing device of claim 4 wherein the simulation program further includes instructions to dynamically depict the tumbling movement of the one or more simulated dice coming to rest to reveal the generated gaming result.

5. The portable computing device according to claim 1 wherein the display is encompassed by a housing in common with the processor.

6. The portable computing device according to claim 1 wherein the display is associated with a separate processor in processing communications over a wireless link with the processor.

7. The portable computing device according to claim 4 wherein the simulation program depicts the tumbling movement of the simulated dice coming to rest in response to a detected cessation of the identified characteristic shaking motions.

8. The portable computing device according to claim 1 wherein the portable computing device is one of, a personal data assistant, a cellular telephone, a portable media player and a handheld game player.

9. The portable computing device of claim 1 wherein the generated gaming result is one of; a number, a word, a sentence, a phrase, a color, an image and any combination thereof.

10. The portable computing device of claim 1 wherein the generated gaming result is used to update a status of play in the game of chance.

11. The portable computing device of claim 1 wherein the generated gaming result is dependent upon at least one dynamic characteristic of the identified characteristic shaking motion.

12. The portable computing device of claim 11 wherein the dynamic characteristic of the identified characteristic shaking motion is one of, a time duration, a magnitude, a number of distinct shake events, a frequency of the distinct shake events and any combination thereof.

13. A method for providing a shake responsive portable computing device for simulating a randomization object used in a game of chance comprising:

   coupling a motion sensor to a processor associated with the portable computing device;

   configuring the motion sensor to detect shaking motions imparted on the portable computing device by a user;

   loading a simulation program into a memory coupled to the processor, the simulation program including instructions executable by the processor for;

   identifying deliberately imparted characteristic shaking motions from detected shaking motions;

   generating a randomized gaming result in dependence on the identified characteristic shaking motion signals; and,

   outputting the randomized gaming result to a display coupled to the processor as a graphical representation of the randomization object.

14. The method according to claim 13 wherein the randomization object comprises one or more simulated dice and wherein the simulation program further includes instructions executable by the processor for simulating a tumbling movement of the one or more simulated dice in dependence on the identified characteristic shaking motions.

15. The method according to claim 14 wherein the simulation program further includes instructions executable by the processor for dynamically depicting the tumbling movement of the one or more simulated dice coming to rest to reveal the gaming result.

16. The method according to claim 14 wherein the simulation program further includes instructions executable by the processor for depicting the tumbling movement of the simulated dice coming to rest in response to a detected cessation of the identified characteristic shaking motions.

17. The method according to claim 13 wherein the simulation program further includes instructions executable by the processor for generating the randomized gaming result in dependence upon one or more dynamic characteristics of the identified characteristic shaking motions selected from the group consisting essentially of; a time duration, a magnitude, a number of distinct shake events, a frequency of distinct shake events and any combination thereof.

18. The method according to claim 16 wherein the simulation program further includes instructions executable by the processor for depicting the tumbling movement of the simulated dice coming to rest in response to a detected cessation of the identified characteristic shaking motions.

19. The method according to claim 13 wherein the display is associated with a separate processor in processing communications over a wireless link with the processor.

20. A computer program product embodied in a tangible form comprising instructions executable by a processor associated with a shake responsive portable computing device to;

   receive shaking motion signals from an operatively coupled motion sensor,

   identify deliberately imparted characteristic shaking motions from the detected shaking motions signals;

   generate a randomized gaming result in dependence on the identified characteristic shaking motion signals; and,

   output the randomized gaming result on an operatively coupled display as a graphical representation of a randomization object.

21. The computer program product according to claim 20 wherein the randomization object comprises one or more simulated dice and wherein the computer program product further including instructions executable by the processor to simulate a tumbling movement of one or more simulated dice in dependence on the identified characteristic shaking motions.

22. The computer program product according to claim 21 further including instructions executable by the processor for simulating the tumbling movement of the one or more dice in approximate coordination with the identified characteristic shaking motions.
23. The computer program product according to claim 21 further including instructions executable by the processor to dynamically depict the tumbling movement of the one or more simulated dice coming to rest to reveal the gaming result.

24. The computer program product according to claim 20 wherein the tangible form is one of; a logical media, a magnetic media and an optical media.