

(43) International Publication Date
12 September 2013 (12.09.2013)(51) International Patent Classification:
G10H 1/02 (2006.01)(21) International Application Number:
PCT/US2013/029466(22) International Filing Date:
6 March 2013 (06.03.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

61/607,585	6 March 2012 (06.03.2012)	US
13/787,738	6 March 2013 (06.03.2013)	US
13/787,731	6 March 2013 (06.03.2013)	US
13/787,734	6 March 2013 (06.03.2013)	US

(71) Applicant: APPLE INC. [US/US]; 1 Infinite Loop, M/s 169-3ip1, Cupertino, CA 95014 (US).

(72) Inventors: LITTLE, Alexander, H.; 1 Infinte Loop, M/S 77-1GL, Cupertino, CA 95014 (US). MANJARREZ, Eli T.; 1 Infinte Loop, M/S 77-1GL, Cupertino, CA 95014 (US).

(74) Agents: EUREK, Justin et al.; KILPATRICK TOWNSEND & STOCKTON LLP, Eighth Floor, Two Embarcadero Center, San Francisco, CA 94111 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

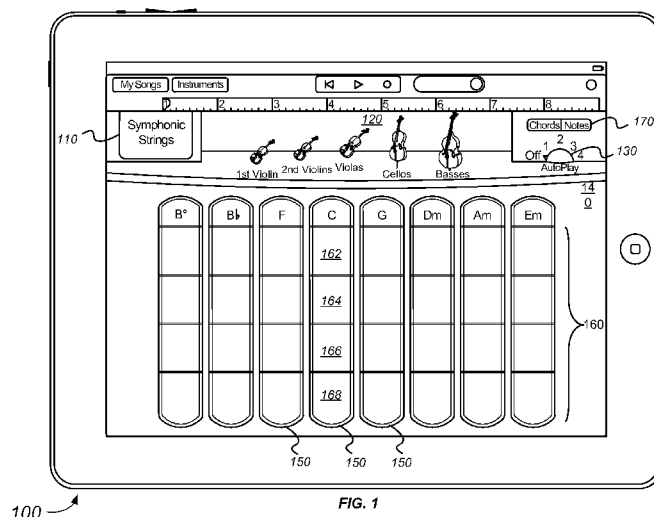
Declarations under Rule 4.17:

— as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:

— without international search report and to be republished upon receipt of that report (Rule 48.2(g))

(54) Title: DETERMINING THE CHARACTERISTIC OF A PLAYED NOTE ON A VIRTUAL INSTRUMENT



(57) Abstract: A user interface implemented on a touch-sensitive display for a virtual musical instrument comprising a plurality of chord touch regions configured in a predetermined sequence, each chord touch region corresponding to a chord in a musical key and being divided into a plurality of separate touch zones, the plurality of chord touch regions defining a predetermined set of chords, where each of the plurality of separate touch zones in each region is associated with one or more preselected MIDI files stored in a computer-readable medium. Each of the plurality of touch zones is configured to detect one or more of a plurality of touch gesture articulations including at least one of a legato articulation, a pizzicato articulation, or a staccato articulation. The one or more of the plurality of touch gesture articulations determines the preselected MIDI file associated with each of the plurality of separate touch zones.

DETERMINING THE CHARACTERISTIC OF A PLAYED NOTE ON A VIRTUAL INSTRUMENT

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present non-provisional application claims benefit under 35 U.S.C. § 119 of
5 U.S. Provisional Patent Application No. 61/607,585, filed on March 6, 2012, and entitled
“DETERMINING THE CHARACTERISTIC OF A PLAYED NOTE ON A VIRTUAL
INSTRUMENT,” which is herein incorporated by reference in its entirety for all purposes.

BACKGROUND

[0002] Virtual musical instruments, such as MIDI-based or software-based keyboards,
10 guitars, basses, and the like, typically have user interfaces that closely resemble the actual
instrument. For example, a virtual piano will have an interface configured as a touch-
sensitive representation of a keyboard, or a virtual guitar will have an interface configured as
a touch-sensitive fret board. While these types of interfaces may be intuitive, they require
that the user understands how to play notes, chords, chord progressions, etc., on a real
15 musical instrument in order to implement them on the virtual musical instrument, such that
the user is able to produce pleasing melodic or harmonic sounds from the virtual instrument.
Such requirements may be problematic for musicians with little to no experience with the
particular corresponding instrument or in music composition in general. Furthermore, some
instruments may not lend themselves to virtual replicas requiring similar input articulations
20 on the user interface. For example, brass or orchestral instruments may require air pressure
or handheld instruments in order to play a note, much less a chord. MIDI keyboards are
typically used as samplers to recreate these orchestral sounds and, as discussed above, require
some level of keyboard proficiency and musical knowledge to produce pleasing melodic or
harmonic phrases.

25 [0003] Not all users who would enjoy playing a virtual instrument are musicians who know
how to form chords, construct chord progressions, or compose diatonic harmony within a
musical key. Furthermore, users who do know how to form chords and play chord
progressions on a real instrument may find it difficult to recreate music with the same
musical proficiency on the user interface due to the lack of tactile stimulus (e.g., weighted
30 keys, strings, bow, etc.), which the user may be accustomed to.

[0004] These problems lead to frustration and make a system less useful, less enjoyable, and less popular. Therefore, a need exists for a system that strikes a balance between simulating traditional musical instruments and providing an optimized user interface that allows effective musical input and performance, and that allows even non-musicians to experience a musical performance on a virtual instrument.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] FIG. 1 shows a schematic illustration of a user interface for a virtual musical instrument, according to an embodiment of the invention.

[0006] FIG. 2 illustrates a plurality of gestural legato articulations over a plurality of musically defined touch regions, according to an embodiment of the invention.

[0007] FIG. 3 illustrates a two-finger auto-legato gestural articulation over a plurality of musically defined touch regions, according to an embodiment of the invention.

[0008] FIG. 4 illustrates a pizzicato gestural articulation in a touch region of a touch-sensitive musical interface, according to an embodiment of the invention.

[0009] FIG. 5 illustrates a staccato gesture articulation in a touch region of a touch-sensitive musical interface, according to an embodiment of the invention.

[0010] FIG. 6 is a simplified flow diagram illustrating aspects of a method of determining a characteristic of a note outputted by a virtual instrument, according to an embodiment of the invention.

[0011] FIG. 7 shows an illustration of a user interface for a virtual musical instrument, according to an embodiment of the invention.

[0012] FIG. 8 illustrates a legato gestural articulation on a touch region of a virtual musical instrument interface, according to an embodiment of the invention.

[0013] FIG. 9 depicts a graph illustrating the cross-fading of two samples of a virtual instrument, according to an embodiment of the invention.

[0014] FIG. 10 illustrates a stereo field mapping configuration of a cross-faded composite sample, according to an embodiment of the invention.

[0015] FIG. 11 depicts a dual flywheel mass modeling system, according to an embodiment of the invention.

[0016] FIG. 12 illustrates an embodiment of a note lifecycle, according to an embodiment of the invention.

[0017] FIG. 13 is a simplified state machine illustrating aspects of modeling a lifecycle of a note played on a virtual instrument, according to an embodiment of the invention.

5 [0018] FIG. 14 illustrates an example of a musical performance system that can enable a user to compose and create music with a number of virtual instruments on a music application, according to an embodiment of the invention.

[0019] FIG. 15 illustrates a computer system according to an embodiment of the present invention.

10 [0020] FIG. 16 illustrates a chord touch region including four touch zones with chord configurations having successively wider harmonic ranges, according to an embodiment of the invention.

[0021] FIG. 17 illustrates a chord touch region including four touch zones with chord configurations having harmonic ranges successively higher in pitch, according to an
15 embodiment of the invention.

[0022] FIG. 18 illustrates a chord touch region including four touch zones with chord configurations that change by a minimum number of notes between chord inversions, according to an embodiment of the invention.

[0023] FIG. 19 illustrates a chord touch region including four touch zones with chord
20 configurations that change by a minimum number of notes between chords, according to an embodiment of the invention.

[0024] FIG. 20 is a simplified flow diagram illustrating aspects of a method of selecting a virtual instrument on a user interface, according to an embodiment of the invention.

[0025] FIG. 21 is a simplified flow diagram illustrating aspects of a method of modulating
25 a lifecycle of a musical note or chord, according to an embodiment of the invention.

DETAILED DESCRIPTION

[0026] The following disclosure describes systems, methods, and products for musical performance and/or input. Various embodiments can include or communicatively couple with a wireless touch screen device. A wireless touch screen device including a processor

can implements the methods of various embodiments. Many other examples and other characteristics will become apparent from the following description.

[0027] A musical performance system can accept user inputs and audibly sound one or more tones. User inputs can be accepted via a user interface. A musical performance system, therefore, bears similarities to a musical instrument. However, unlike most musical instruments, a musical performance system is not limited to one set of tones. For example, a classical violin or a classical cello can sound only one set of tones, because a musician's interaction with the physical characteristics of the instrument produces the tones. On the other hand, a musical performance system can allow a user to modify one or more tones in a set of tones or to switch between multiple sets of tones. A musical performance system can allow a user to modify one or more tones in a set of tones by employing one or more effects units. A musical performance system can allow a user to switch between multiple sets of tones. Each set of tones can be associated with a patch (e.g., channel strip (CST) file).

[0028] FIG. 14 illustrates an example of a musical performance system that can enable a user to compose and create music with a number of virtual instruments on a music application, according to an embodiment of the invention. Musical performance system 1400 can include multiple subsystems such as a display 1405, one or more processing units 1410, and a storage subsystem 1415. One or more communication paths can be provided to enable one or more of the subsystems to communicate with and exchange data with one another. The various subsystems in FIG. 14 can be implemented in software, in hardware, or combinations thereof. In some embodiments, the software can be stored on a transitory or non-transitory computer readable storage medium and can be executed by one or more processing units.

[0029] It should be appreciated that musical performance system 1400 as shown in FIG. 14 can include more or fewer components than those shown in FIG. 14, can combine two or more components, or can have a different configuration or arrangement of components. In some embodiments, musical performance system 1400 can be a part of a portable computing device, such as a tablet computer, a mobile telephone, a smart phone, a desktop computer, a laptop computer, a kiosk, etc.

[0030] Display 1405 in some embodiments can provide an interface that allows a user to interact with musical performance system 1400. Display 1405 can be a monitor or a screen

in some embodiments. Through the interface, the user can view and interact with a GUI 1420 of a musical performance system 1400. In some embodiments, display 1405 can include a touch-sensitive interface (also sometimes referred to as a touch screen) that can both display information to the user and receive inputs from the user. Processing unit(s) 1410 can include one or more processors that each have one or more cores. In some embodiments, processing unit(s) 1410 can execute instructions stored in storage subsystem 1415. System 1400 may also include other types of user input and output mechanisms such as allowing a user to provide an input based on received accelerometer or gyroscope sensor readings (internal to system 1400) or provide output such as haptic output based on a desired musical characteristic.

[0031] Storage subsystem 1415 can include various memory units such as a system memory 1430, a read-only memory (ROM) 1440, and a permanent storage device 1450. The system memory can be a read-and-write memory device or a volatile read-and-write memory, such as dynamic random access memory. The system memory can store some or all of the instructions and data that the processor needs at runtime. The ROM can store static data and instructions that are needed by processing unit(s) 1410 and other modules of system 1400. The permanent storage device can be a read-and-write memory device. Some embodiments of the invention can use a mass-storage device (such as a magnetic or optical disk or flash memory) as a permanent storage device. Other embodiments can use a removable storage device (e.g., a floppy disk, a flash drive) as a permanent storage device.

[0032] Storage subsystem 1415 can store a touch gesture library 1415 that includes a number of system recognizable touch gestures 1432 on the GUI 1420, MIDI-controlled audio samples 1434 for storing data relating to music played on the virtual instruments, and virtual instrument data 1436 for storing information about each virtual instrument. Further detail regarding system architecture and auxiliary components thereof are not discussed in detail so as not to obfuscate the focus on the invention and would be understood by those of ordinary skill in the art.

[0033] Chord View Interface

[0034] FIG. 1 shows a schematic illustration of a user interface 100 for a virtual musical instrument, according to an embodiment of the invention. FIG. 1 shows the user interface displayed on a tablet computer such as the Apple iPad®; however the interface could be used

on any touch screen or touch-sensitive computing device (e.g., smart phone, personal digital assistant, and the like). The interface 100 includes channel strip file (CST) sound browser 110, a virtual instrument selector (VIS) 120, an autoplay selector 130, a chord player region 140, and a chord/notes switch 170. Alternatively, other sound files, or “patches” can be used instead of, or in addition to, the CST files, which can be accessed with the sound browser 110.

[0035] The CST sound browser 110 can be configured to select a suite of virtual instruments, where each suite or ensemble is suited for a different musical preference. Selecting a given suite (e.g., Symphonic Strings, String Quartet, Baroque Strings, etc.) loads the appropriate CST and autoplay grooves specifically suited to the selected ensemble. For example, the Symphonic CST can include a first violin, a second violin, a viola, a cello, and a bass. In contrast, the String Quartet CST may include a first a second violin, a viola, and cello. Any number and configuration of virtual instruments can be grouped together in any preferred combination. In some embodiments, MIDI sequences can be used instead of, or in addition to, the autoplay grooves.

[0036] The interface 100 includes a number of chord touch regions (chord strips) 150, shown for example as a set of eight adjacent columns or strips. Each touch region 150 can correspond to a pre-defined chord (base chord) within one or more particular keys, with adjacent regions configured to correspond to different chords and progressions within the key or keys. For example, the key of C major includes the diatonic chords of C major (I), D minor (ii), E minor (iii), F major (IV), G major (V), A minor (vi), and B diminished (vii), otherwise known as the Tonic, Supertonic, Mediant, Subdominant, Dominant, Submediant, and Leading Tone chords. In the example shown in Fig. 1, an additional non-diatonic chord of B-flat major is included with the key of C major, and the chords are arranged sequentially in accordance with the circle of fifths. This arrangement allows a user to easily create sonically pleasing sequences by exploring adjacent touch regions. A user can touch or initiate gestures (e.g., swipe gestures) on the various touch regions 150 to create sounds that are related to the chord that is assigned to the particular touch region. The types of touches, sounds, and/or gestures associated with the touch regions and features therein are further described below. It should be noted that any set of pitches (collectively known as a chord) can be defined for any chord touch region and is not limited to the conventional harmony described in this particular example.

[0037] Each chord touch region is divided into a number of touch zones referred to collectively as touch zones 160. Referring to FIG. 1, each touch region 150 comprises four individual touch zones 162, 164, 166, 168. Each of the touch zones 160 can correspond to various harmonic voicings (e.g., chord inversions) of the base chord assigned to the touch region 150 that it belongs to. The chords assigned to each touch zone 160 can be associated with MIDI-controlled audio samples (e.g., signals, streams, etc.). Touching or initiating touch gestures on any touch zone 160 in a region 150 plays the chord audio sample assigned to that touch zone.

[0038] MIDI (Musical Instrument Digital Interface) is an industry-standard protocol defined in 1982 that enables electronic musical instruments, such as keyboard controllers, computers, and other electronic equipment, to communicate, control, and synchronize with each other. For example, MIDI can be used to trigger playback of an audio sample to create a sound. In other words, MIDI is an instruction communications protocol typically used in electronic musical instruments. It should be noted that the term “audio signal” can also include a MIDI-controlled audio sample or digital signal processor (DSP) generated audio stream. Other audio processing systems and protocols may be used as would be known by one of ordinary skill in the art.

[0039] Virtual Instrument Selection in Chords View

[0040] A user can select a virtual instrument via the VIS 120. In some cases, the virtual instruments can include one or more violins, violas, cellos, basses, or the like. When a user selects one or more instruments with the VIS 120, the system loads the appropriate patch files (e.g., CST files) for the selected instruments. In some embodiments, the VIS 120 displays up to 5 instruments or more, which when selected (e.g., by tapping) will either activate or deactivate a note being played by that particular instrument. For example, while performing gestural inputs on the touch zone region corresponding to a given chord sequence, a user may select a violin and a bass to play (i.e., output an audible musical output) the given chord sequence. During the input of gestures to create the chord sequence, the user can deselect the bass in real-time, leaving only the violin sample to be voiced during the chords sequence. Likewise, reselecting the bass thereafter will reintroduce the bass sample’s voice into the chord sequence.

[0041] Grooves / Auto-play Selection

[0042] In certain embodiments, the interface 110 includes various chord sequence auto-play features. An auto-play selector 130 is configured to loop one or more MIDI-controlled audio samples (e.g., audio signals/streams, audio loops, etc.) that include a number of predefined accompanying rhythms when selected by a user. In response to a user selection, a groove plays for the chord being touched by the user. In some embodiments, the groove rhythm latches or stops when the user touches the same chord region again. The groove rhythm can switch to a new chord by selecting a different touch zone 150. The playback tempo can be locked during playback or manually adjustable as required. Variations of the auto-play grooves can be selected by one or more taps on the auto-play selector 130. In some cases, the auto-play grooves can be a MIDI sequence. Furthermore, the auto-play grooves (or MIDI sequences) will also correspond to the chord voicing selected for each touch zone 160.

[0043] Chord Inversions

[0044] In certain embodiments, each of the touch zones 160 of a touch region 150 can include various inversions of the base chord. For example, the top most touch zone 162 can correspond to a root position of the base chord, such as a C Major triad C-E-G with C in the bass. The next adjacent touch zone 164 can correspond to a first inversion of the base chord, or the C Major triad C-E-G with E in the bass. The following adjacent touch zone 166 can correspond to a second inversion of the base chord, or the C Major triad C-E-G with G in the bass. The next adjacent touch zone 168 can correspond to a third inversion of the base chord, or the C Major triad C-E-G with an additional 7th in the bass (e.g., Cmaj7 with B in the bass or a C7 with a Bb in the bass). Alternatively, the touch zone 168 may include a 10th in the bass, or E-C-G, where the E note is one octave higher than the E note in the first inversion E. Optional adaptations of the first, second, and third inversions can be utilized.

[0045] In some embodiments, each of the touch zones 160 of a touch region 150 can include various inversions of the base chord that are successively higher in relative pitch than the base chord. To illustrate, the top most touch zone 162 may correspond to a root position of the base chord, such as a C Major triad C-E-G with C in the bass. The next adjacent touch zone 164 may correspond to a first inversion of the base chord, or the C Major triad C-E-G with E in the bass, where at least one note of the chord (e.g., the top note) is higher than any note of the base chord. Similarly, the following touch zone 166 may include a second inversion of the base chord, where at least one note of the second inversion chord (e.g., the top note) is higher in pitch than any note of the base chord or first inversions. Finally, the last

adjacent touch zone 168 can include a third inversion of the base chord, where at least one note of the third inversion chord is higher in pitch than any note of the base chord or first inversions. Configuring inversions in this manner can make cycling through the touch zones 160 sounds as if the relative pitch is moving from one register to a higher register.

- 5 [0046] In further embodiments, each of the touch zones 160 of a touch region 150 can include various chord voicings (i.e., the harmonic arrangement of notes) that have successively wider harmonic ranges. To illustrate, the top most touch zone 162 may correspond to the base chord with a certain harmonic range. For example, touch zone 162 can include a C triad comprising C-E-G where the G can be 3.5 whole steps from the root C.
- 10 The next touch zone 162 may correspond to a chord voicing (e.g., inversion) with a wider harmonic range than the base chord of touch zone 162. For example, touch zone 162 can include a C triad comprising C-E-G with E in the bass and a C as the top note, where the top note C is 4 steps from the root E. The third and fourth touch zones 166 and 168 may follow in kind.
- 15 [0047] FIG. 16 illustrates a chord touch region 1600 including four touch zones with chord configurations having successively wider harmonic ranges, according to an embodiment of the invention. The chord touch region 1600 is assigned base chord C major and includes touch zones 1610, 1620, 1630, and 1640. In some embodiments, each touch zone has a different chord voicing of the base chord. As described above, the term “chord voicing”
- 20 relates to a harmonic arrangement of notes. For example, a C major chord has the notes C-E-G. One chord configuration can have a C note as the bass note, while a second configuration may have the G note as the bass note. Each of these configurations can be referred to as a chord voicing. The distance between notes can be measured by half steps and whole steps, which can be thought of as musical units of measurement. Touch zone 1610 has a first
- 25 voicing (i.e., harmonic arrangement) 1615 of a C major chord. The harmonic range of the first voicing is 3.5 whole steps. Touch zone 1620 has a second voicing 1625 of the C major chord with a wider harmonic range than the first voicing spanning 4 whole steps. Touch zone 1630 has a third voicing 1635 of the C major chord with a wider harmonic range than the second voicing spanning 8 whole steps. Touch zone 1640 has a fourth voicing 1645 of the C
- 30 major chord with a wider harmonic range than the third voicing spanning 10 whole steps. Different chords, chord configurations, inversions, key signatures, and the like, can be customized to preference.

[0048] FIG. 17 illustrates a chord touch region 1700 including four touch zones with chord configurations having harmonic ranges successively higher in pitch, according to an embodiment of the invention. The chord touch region 1700 is assigned base chord C major and includes touch zones 1710, 1720, 1730, and 1740. In some embodiments, each touch zone has a different chord voicing of the base chord. As described above, the term “chord voicing” relates to a harmonic arrangement of notes. Touch zone 1710 has a first voicing (i.e., harmonic arrangement) 1715 of a C major chord. Touch zone 1720 has a second voicing 1725 of the C major chord with a higher pitch than the first voicing. In other words, both the E and C notes in chord voicing (“chord”) 1725 are higher in pitch than the highest note (G) of chord 1715. In some cases, the audible effect of only one note in a first chord being higher in pitch than all notes of a second chord will make the first chord sound higher in pitch overall. Referring back to FIG. 17, touch zone 1730 has a third voicing 1735 of the C major chord with a higher pitch than the first (1715) and second voicings (1725). For example, the G note is higher in pitch than any of the notes of chord 1715 and 1725. Touch zone 1740 has a fourth voicing 1745 of the C major chord with a higher pitch than the first (1715), second voicings (1725), and third voicings (1735). For example, the C note is higher in pitch than any of the notes of chords (i.e., voicings) 1715, 1725, and 1735. It should be noted that other chords, chord configurations, inversions, key signatures, and the like, can be customized to preference.

[0049] In some embodiments, touching or articulating gestures up or down through the touch zones 160 may cause the chord voicing to change by the minimum number of notes needed to switch to the nearest inversion from the chord voicing that was being played prior to the touch or gesture articulation. For example, if touch zone 162 includes a C major triad C-E-G with C in the bass, and touch zone 164 includes a C major triad E-G-C with an E in the bass, the E and G notes will not change between chords. Changing the minimum amount on notes in a chord change can result in a smoother or more harmonically pleasing sound, as further illustrated in FIG. 18.

[0050] FIG. 18 illustrates a chord touch region including four touch zones with chord configurations that change by a minimum number of notes between chord inversions, according to an embodiment of the invention. The chord touch region 1800 is assigned base chord C major and includes touch zones 1810, 1820, 1830, and 1840. Touch zone 1810 includes a C major triad C-E-G with C in the bass, and touch zone 1820 includes a C major

triad E-G-C with an E in the bass. It should be noted that the E and G notes do not change during the chord change 1815. As described above, changes between chord inversions (e.g., 1810 to 1820) result in a change in the minimum amount of notes needed for the chord change. In some cases, this can be a dynamic process such that different chord voicings may be assigned to the same touch zone depending on the arrangement of the previous chord voicing. For example, the C chord inversion assigned to 1810 may produce (e.g., play back) a chord voicing of a different arrangement when approached from touch zone 1820 rather than from touch zone 1830 because the arrangement of the notes and the minimum changes required for the chord change is different. A first inversion C chord with an E in the bass may only require one note to shift positions when switching to a C chord with a C root note (e.g., C note moves to the bass position and other notes do not change). However, a second inversion chord with a G in the bass may require multiple notes to shift positions when switching to a C chord with a C root note. It should be noted that different chords, chord configurations, inversions, key signatures, and the like, can be customized to preference.

[0051] In other embodiments, touching or articulating gestures between chord regions may cause chord voicings to change by the minimum number of notes needed to switch between chords. For example, if one touch region includes a C major triad C-E-G as a base chord, and a second touch region includes an E minor triad E-G-B, the E and G notes does not change between the chord change. Changing the minimum number of notes between chord changes can result in more harmonically pleasing chord progressions.

[0052] FIG. 19 illustrates a chord touch region including four touch zones with chord configurations that change by a minimum number of notes between chords, according to an embodiment of the invention. The chord touch region 1900 is assigned base chord C major and includes touch zones 1910, 1920, 1930, and 1940. The chord touch region 1950 is assigned base chord E minor and includes touch zones 1960, 1970, 1980, and 1990. Touch zone 1910 includes a C major triad C-E-G with C in the bass, and touch zone 1960 includes a E minor triad E-G-B with an E in the bass. As described above, changes between chords result in a change in the minimum amount of notes needed for the chord change. In this case, the E and G notes do not change position during the chord change 1935. In some cases, this can be a dynamic process such that different chord voicings may be assigned to the same touch zone depending on the arrangement of the previous chord voicing. For example, the E minor chord inversion assigned to 1970 may produce (e.g., play back) a chord voicing of a

different when approached from touch zone 1910 rather than from touch zone 1920 because the arrangement of the notes and the minimum changes required for the chord change may be different. As described above, changing from a C chord with a C note in the bass to an E minor chord with an E in the bass results in only one note change (i.e., C changes to B), while both the E and G notes do not change. However, changing from a second inversion C chord (1930) with a G in the bass to an Em chord with E in the bass position may require multiple notes to shift position. It should be noted that different chords, chord configurations, inversions, key signatures, and the like, can be customized to preference.

[0053] Touch Gesture Articulations in Chord View

[0054] Each touch zone 160 in each touch region 150 is configured to detect one or more of a number of different touch gestural articulations, according to certain embodiments. The touch gesture articulations can include a legato articulation, a pizzicato articulation, and a staccato articulation. Each of the gesture articulations triggers playback (i.e., output) of a corresponding audio signal that simulates the selected articulation. In some cases, the audio signal can be a MIDI-controlled audio sample or a DSP generated audio stream. It should be noted that the term “audio signal” can also mean any suitable means of playing an audio signal. Playing an audio signal can include playback on an electro-acoustic transducer (e.g., a speaker), decoding the audio signal, saving in memory, coupling to other systems or devices, or other non-audible output scenarios that would be appreciated by one of ordinary skill in the art with the benefit of this disclosure. It should be noted that although three different gesture articulations are described, any number and/or type of gesture articulations (e.g., double taps, dual finger swipes, palm, circular motions, polygonal outline motions, multi-finger gestures, etc.) may be used in any combination or set as required.

[0055] FIG. 2 illustrates a plurality of legato articulations over a plurality of touch regions 250, according to an embodiment of the invention. The touch regions 250 and touch zones 260 are similar to those which are described in FIG. 1. The legato articulation is initiated by a tap-to-swipe gesture on one of the touch zones 260. In certain cases, the initial tap should be longer than a predetermined period of time before initiating the swipe gesture to differentiate the legato articulation from the staccato (i.e., tap gesture shorter than the predetermined time) or pizzicato articulation (i.e., swipe gesture with no initial tap period) further described below. A continuous up-and-down swiping motion of the legato articulation mimics the motion of a bow played across a set of strings on a stringed

instrument (e.g., violin, cello, bass). Similarly, the corresponding audio signal (e.g., MIDI-controlled audio sample) plays the sound of a bowing action across a set of strings on a stringed instrument. In FIG. 2, gesture 270 illustrates a touch-and-swipe legato gesture on touch zone 264A. The initial touch point on the touch zone 264A determines the inversion of the base chord that is played. In this example, the legato articulation 270 plays a first inversion legato bowing sound in A minor. Gesture 280 illustrates a touch-and-continuous swipe legato gesture at touch zone 264B. In this example, the legato articulation 280 plays a first inversion legato bowing sound in C major and continues to do so even if the swiping motion moves outside of the associated touch region. In some embodiments, as long as the finger is not removed from the touch surface of the interface 100, the bowing action will continue to play the same inversion of the initial touch point regardless of the subsequent position of the finger on the interface. In some cases, changing base chords or chord inversions occurs by starting a new legato gesture articulation in a different zone by lifting the finger and placing it in the desired chord/inversion region. Alternatively, introducing a second finger in a different touch zone can begin a new legato gesture articulation. In other words, the interface recognizes the second touch input as a new legato gestural articulation and ignores the first finger. Furthermore, the speed or velocity of the legato articulation can determine various expression values of the corresponding audio signal, such as a volume of playback. Other expression values can include pitch, duration, equalization, filtering (e.g., high pass, low pass, band pass, notch filters, etc.), pan, effects (e.g., reverb, chorus, delay, distortion, presence, etc.), or any other parameters that can be modulated as described. In some embodiments, the direction and/or acceleration of the gesture articulation can also be used to modulate one or more expression values (e.g., variable of playback).

[0056] In certain embodiments, multi-finger gestures can allow the user to quickly transition from one chord zone to another. For example, if a second touch point is detected within the chord interface while a bowing gesture is active in another chord touch zone, the chord can quickly change to the new chord touch zone. Furthermore, if the second touch is a bow swipe (e.g., legato articulation), the expression (e.g., a volume) value generated from the first touch transfers to the new expression value for the second touch using the same swipe gesture ballistics that were present immediately prior to the change. To illustrate, if two consecutive swipes are occurring at substantially the same swipe gesture speed, then there should be substantially no change in expression value (volume) when the chord transition occurs. In contrast, if the second gesture occurs at a different speed than the first gesture,

then the expression value (volume) transitions from the first expression value to the second expression value in a smooth and musical way, as further described below.

[0057] FIG. 3 illustrates a two-finger auto-legato gesture articulation, according to an embodiment of the invention. FIG. 3 depicts a first touch 370 (tap-and-hold) and
5 simultaneous second touch 372 (tap-and-hold) at touch zone 364A of the touch region 350A. Tapping and holding two fingers in a touch zone auto-plays a legato chord sound. In this example, the auto legato gesture articulation plays the inversion associated with touch zone 364A of touch region 350A (A minor). Two finger auto-legato gestures do not require a swipe gesture and can be configured to play instantly. In some cases, the volume of the auto-
10 legato articulation is fixed and can be based on the initial velocity of the two finger-tap gesture. The initial velocity can be determined, for example, by an accelerometer. In certain cases where multiple touch zones are simultaneously touched, a resultant touch location can be selected as the initial touch zone. For example, a first touch 380 at touch zone 364B and a simultaneous second touch 382 at touch zone 368B results in an effective two finger-tap
15 gesture 384 at touch zone 366B of touch region 350B. The note characteristics of the two-finger auto-legato touch gestures typically have relatively short attack times to support rapid transitions to different base chords or inversions.

[0058] FIG. 4 illustrates a pizzicato gesture articulation in a touch region 450 of a touch-sensitive interface, according to an embodiment of the invention. The pizzicato gesture
20 articulation can be initiated by a quick tap gesture 480 including an initial tap in one of the plurality of separate touch zones 460 and a release prior to a predetermined period. The pizzicato gestural articulation triggers the playback of a second corresponding MIDI-controlled audio sample that simulates the sound of a plucking action on a stringed instrument. In one embodiment, the predetermined period is any suitably short period of time
25 (e.g., less than 200 ms) that would be appreciated by one of ordinary skill in the art with the benefit of this disclosure. In some embodiments, if the touch is held longer than the predetermined period, the removal of the finger will not trigger the audio sample. Furthermore, the speed or velocity of the pizzicato gestural articulation (e.g., the speed at which the initial tap on the interface is received) can determine various expression values of
30 the corresponding audio sample, such as a volume of playback. Other expression values can include pitch, duration, equalization, filtering (e.g., high pass, low pass, band pass, notch filters, etc.), pan, effects (e.g., reverb, chorus, delay, distortion, presence, etc.), or any other

parameters that can be modulated as described. The initial velocity of the initial tap may be determined by an accelerometer.

[0059] FIG. 5 illustrates a staccato gesture articulation 580 in a touch region 550 of a touch-sensitive interface, according to an embodiment of the invention. The staccato gestural articulation is initiated by an in-motion swipe gesture beginning upon contact with a touch zone 560. In other words, a quick swipe motion is initiated with no pause (e.g., tap) before beginning the swipe, where the swipe begins upon contact with the given touch zone 560. The staccato gesture articulation triggers playback of a corresponding MIDI controlled audio signal that simulates the sound of a short burst of a bowing action on a stringed instrument.

The speed or velocity of the staccato articulation can determine various expression values of the corresponding audio sample, such as a volume of playback. Other expression values can include pitch, duration, equalization, filtering (e.g., high pass, low pass, band pass, notch filters, etc.), pan, effects (e.g., reverb, chorus, delay, distortion, presence, etc.), or any other parameters that can be modulated as described.

[0060] FIG. 6 is a simplified flow diagram illustrating aspects of a method 600 of determining a characteristic of a note played on a virtual instrument, according to an embodiment of the invention. The method 600 is performed by processing logic that may comprise hardware (e.g., circuitry, dedicate logic, etc.), software (which as is run on a general purpose computing system or a dedicated machine), firmware (embedded software), or any combination thereof. In one embodiment, the method 600 is performed by the processor unit 1410 of FIG. 14.

[0061] Referring to FIG. 6, the method 600 begins with the receiving a user input on a touch zone of a touch sensitive interface (610). The user input can include one of a number of touch gesture articulations. At 620, the processor determines if the touch gesture is a legato articulation. A legato articulation includes a touch-to-swipe gesture that imitates the motion of a bow played across a set of strings on a stringed instrument (e.g., violin, cello, bass). In certain cases, the initial touch should be longer than a predetermined period of time before initiating the swipe gesture to differentiate the legato articulation from the staccato (i.e., tap gesture shorter than the predetermined time) or pizzicato articulation (i.e., swipe gesture with no initial tap period). If the processor determines that the user input is a legato articulation, the processor plays (or outputs) one of a number of preselected audio signals (e.g., MIDI controlled audio samples, DSP generated audio streams, etc.) on an output device

(e.g., speaker, headphones, or the like) corresponding to both the touch region selected (e.g., chord inversions) and the legato articulation (650). If the processor determines that the user input is not a legato articulation, the method proceeds to 630.

[0062] At 630, the processor determines if the touch gesture is a pizzicato articulation. A pizzicato gesture articulation can be initiated by a quick tap gesture including an initial tap in a touch zone and a release prior to a predetermined period. The pizzicato articulation imitates the sound of a plucking action on a stringed instrument. If the processor determines that the user input is a pizzicato articulation, the processor plays one of a number of preselected audio signals on an output device corresponding to both the touch region selected (e.g., chord inversions) and the pizzicato articulation (650). If the processor determines that the user input is not a pizzicato articulation, the method proceeds to 640.

[0063] At 640, the processor determines if the touch gesture is a staccato articulation. The staccato articulation can be initiated by an in-motion swipe gesture (i.e., no pause or tap) beginning upon contact with a touch zone. The staccato gesture articulation imitates the sound of a short burst of a bowing action on a stringed instrument. If the processor determines that the user input is a staccato articulation, the processor plays one of a number of preselected audio signals (e.g., MIDI-controlled audio samples) on an output device corresponding to both the touch region selected (e.g., chord inversions) and the staccato articulation (650). If the processor determines that the user input is not a staccato articulation, the method ends.

[0064] Differentiation between each of the legato and staccato gestural articulations is based on detecting a time interval associated with the initial touch which is followed by a consecutive (i.e., without a lift off of the touch) swiping gesture having a particular initial speed. A legato articulation is detected based on the detecting of an initial touch lasting a given time interval, which is followed by a swiping motion of any speed. The staccato gestural articulation is based on detecting that the initial touch is less than the given time interval.

[0065] It should be appreciated that the specific steps illustrated in FIG. 6 provides a particular method of determining a characteristic of a note played on a virtual instrument, according to an embodiment of the present invention. Other sequences of steps may also be performed according to alternative embodiments. In certain embodiments, the method 600

may perform the individual steps in a different order, at the same time, or any other sequence for a particular application. For example, the pizzicato gesture may be determined before the legato gesture. Moreover, the individual steps illustrated in FIG. 6 may include multiple sub-steps that may be performed in various sequences as appropriate to the individual step.

- 5 Furthermore, additional steps may be added or removed depending on the particular applications. One of ordinary skill in the art would recognize and appreciate many variation, modification, and alternatives of the method.

[0066] Notes View Interface

- [0067]** FIG. 7 shows an illustration of a user interface 700 for a virtual musical instrument, according to an embodiment of the invention. FIG. 7 shows the user interface displayed on a
10 tablet computer such as the Apple iPad®; however the interface could be used on any touch screen or touch-sensitive computing device (e.g., smart phone, personal digital assistant, and the like). The interface 700 includes CST sound browser 710, a virtual instrument selector (VIS) 720, a violin neck and headstock 730, an Arco/Pizz badge 740, and a chords/notes
15 switch 770. The violin neck 730 includes a plurality of virtual strings 750, where each string corresponds to a base note (open string) and is divided into a plurality of touch zones 760 with notes corresponding to a predetermined scale (e.g., chromatic, pentatonic, harmonic minor, etc.). Each of the notes can correspond to one or more MIDI-controlled audio samples associated with the selected virtual instrument, the type of articulation performed, expression
20 value, or any combination thereof, as further described below.

- [0068]** The chords/notes switch 770 functions as a virtual toggle switch configured to switch between the chords view (e.g., FIG. 1) and the notes view (e.g., FIG. 7). The CST sound browser 710 can be configured to select a suite of virtual instruments and their associated audio signals (e.g., MIDI-controlled audio sample, DSP generated audio stream,
25 etc.), where each suite or ensemble is suited for a different musical preference. Selecting a given suite (e.g., Symphonic Strings, String Quartet, Baroque Strings, etc.) loads the appropriate CST files specifically suited to the selected ensemble. For example, the Symphonic CST can include a first violin, a second violin, a viola, a cello, and a bass. In contrast, the String Quartet CST may include a first a second violin, a viola, and cello. Any
30 number of virtual instruments can be grouped together in any preferred configuration.

[0069] Virtual Instrument Selection in Notes View

[0070] The VIS 720 is used to select the virtual instruments desired by the user. In some cases, the virtual instruments can include one or more violins, violas, cellos, basses, or the like. When a user selects one or more instruments with the VIS 720, the system will load the appropriate audio samples for the selected instruments. In some embodiments, one
5 instrument can be played at a time in notes view. For example, a user may select a cello, thereby deactivating any basses, violins, and/or violas in that particular suite of virtual instruments. As shown in FIG. 7, only one virtual instrument is fully shown at one time to clearly identify the active virtual instrument. In alternative embodiments, multiple instruments can be selected for simultaneous playback in notes view. For example, the VIS
10 720 ("part selector") can display up to 5 instruments or more, which when selected (e.g., by tapping) either activates or deactivates a note being played by the selected instruments.

[0071] In certain embodiments, the VIS 720 displays a number of virtual instruments (e.g., violins, violas, cellos, and basses) that are arranged on a virtual scrollable area on the user interface 700. A swipe gesture on the virtual scrollable area can cause the area to slide in a
15 direction relative to the swipe gesture, thereby changing an orientation of the virtual scrollable area. For example, as a user performs a swipe gesture on the VIS 720 from right-to-left, the virtual instruments cycle through or moves across the VIS 720 display area from right-to-left at a speed proportional to the speed of the swipe gesture. Selecting a virtual instrument can be based on the orientation of the virtual scrollable area. In addition, the
20 virtual scrollable area can have a selection location 790 to select the virtual instrument located in the selection location 790. In some cases, the selection location is located in the center position of the VIS 720. Referring to FIG. 7, the cello is currently selected because it is positioned in the selection location 790. It should be noted that the configuration of the VIS 720 can be customized to a user preference. For example, all virtual instruments may be
25 visible on the VIS 720 or only a subset at any given time. The virtual scrollable area may slide linearly, or on a virtual axis of a suitable radius such that the virtual instruments appear to "rotate" into position at the selection location 790. In some embodiments, a selection may be made automatically as a virtual instrument slides into the selection location 790, or a selection may require an additional manual tap on the virtual instrument to confirm selection.

[0072] FIG. 20 is a simplified flow diagram illustrating aspects of a method 2000 of selecting a virtual instrument on a user interface, according to an embodiment of the invention. The method 2000 is performed by processing logic that may comprise hardware

(e.g., circuitry, dedicate logic, etc.), software (which as is run on a general purpose computing system or a dedicated machine), firmware (embedded software), or any combination thereof. In one embodiment, the method 2000 is performed by the processor unit 1410 of FIG. 14.

5 [0073] Referring to FIG. 20, the method 2000 begins with displaying a plurality of virtual instruments on a scrollable area of a user interface (2010). In some cases, the user interface can be touch-sensitive, where the scrollable area is directly accessible by user touch (e.g., finger, stylus, etc.). In non-touch sensitive embodiments, the scrollable area may be accessible via input device (e.g., mouse, keyboard, etc.). At 2020, the user interface (by way
10 of a processor device) detects a swipe gesture on the virtual scrollable area. In response to the swipe gesture, the user interface scrolls the scrollable area (e.g., VIS 720) and virtual instruments (e.g., virtual violin, viola, cello, bass, etc.) thereon in a direction relative to the swipe gesture (2030). In certain embodiments, as a user performs a swipe gesture on the scrollable area (e.g., VIS 720) from right-to-left, the virtual instruments cycle through or
15 moves across the VIS 720 display area from right-to-left at a speed proportional to the speed of the swipe gesture. The user interface then selects a virtual instrument based on an orientation of the virtual scrollable area (2040). The virtual scrollable area can have a selection location configured within the scrollable area to identify the virtual instrument to be selected. For example, if the user interface scrolls through the virtual instruments in response
20 to a swipe gesture and a violin stops in the selection location, then the violin will be selected. In some cases, the selection location is located in the center position of the scrollable area. The selection location can optionally be configured in any suitable position on the scrollable area. At 2050, the user interface assigns the selected virtual instrument to a first touch zone on the user interface. The user interface further assign one or more MIDI-controlled audio
25 samples to the first touch zone where the audio samples correspond to the selected virtual instrument (2060)

[0074] It should be appreciated that the specific steps illustrated in FIG. 20 provides a particular method of selecting a virtual instrument on a user interface, according to an embodiment of the present invention. Other sequences of steps may also be performed
30 according to alternative embodiments. In certain embodiments, the method 2000 may perform the individual steps in a different order, at the same time, or any other sequence for a particular application. Moreover, the individual steps illustrated in FIG. 20 may include

multiple sub-steps that may be performed in various sequences as appropriate to the individual step. Furthermore, additional steps may be added or removed depending on the particular applications. One of ordinary skill in the art would recognize and appreciate many variation, modification, and alternatives of the method.

5 **[0075]** Touch Gesture Articulation in Notes View

[0076] Each of the virtual strings 750 of a user interface 700 is configured to detect one or more of a plurality of different touch gesture articulations, according to certain embodiments. The touch gesture articulations can include a legato articulation, a pizzicato articulation, and a staccato articulation. Each of the gesture articulations triggers playback of a corresponding
10 MIDI-controlled audio sample (or DSP generated audio stream) that simulates the selected articulation. As described above, the legato articulation mimics a bowing action across a set of strings, the pizzicato articulation mimics a sound of plucking a stringed instrument, and a staccato articulation mimics a sound of a short burst of a bowing action on a stringed instrument.

15 **[0077]** In some embodiments, the Arco | Pizz (i.e., bow/pizzicato) badge 740 is displayed at the far left of the notes view inside the headstock graphic. The arco/pizz badge (“button badge”) 740 is a toggle switch configured to select between a first mode (auto-bow mode) and second mode (arco | pizz mode) of operation. In some embodiments, the button badge 740 is a temporary switch where the badge remains in the second mode of operation as long
20 as a user is pressing the badge 740. In other embodiments, the badge 740 toggles between the two modes of operation and remains in the selected mode of operation until the badge 740 is depressed again.

[0078] In the auto-bow mode of operation (i.e., first mode), touching any point on one of the strings 750 can produce a corresponding legato note with a short attack. The legato note
25 plays continuously as long as the touch object (e.g., finger, stylus, or any input device) remains on the strings. In certain embodiments, the first mode of operation interprets any gesture (e.g., tap, swipe, etc.) as a touch and plays the corresponding legato note, regardless of the initiating gesture.

[0079] FIG. 8 illustrates a legato articulation on the notes view interface 800, according to
30 an embodiment of the invention. When a user presses the button badge 840 and holds a point 880 on a string 864, a graphic indicator 885 is shown prompting the user to perform an up

and down swiping (bowing) motion. In some cases, the graphic indicator 885 is a note highlight radiating from the initial touch point 880 to indicate the bow motion gesture to the user. The interface 800 typically displays the graphic indicator 885 when the button badge 840 is active, a touch is detected on a string, and the touch is determined not to be a pizzicato tap (i.e., the tap is longer than a predetermined period). As the user swipes up and down, the instrument plays a legato note based on the start position of the bowing action. While the badge button 840 is held, the user can tap any string position for a corresponding pizzicato note. In certain embodiments, only one instrument can be played at a time (violin, viola, cello, bass) in the notes view. In alternative embodiments, multiple instruments can be selected in the notes view for simultaneous play.

[0080] With the button badge 840 pressed, both legato and staccato articulations can be performed in a similar way as in the chords view. The speed or velocity of the staccato articulation (i.e., how hard the note is tapped) can determine various expression values of the corresponding MIDI-controlled audio sample, such as a playback volume. A faster legato articulation can create a louder volume and slower legato articulations may create a softer volume. The notes view can be further configured to support multi-finger detection (e.g., double stop chords) with staccato, legato, and pizzicato articulations. In certain aspects, the instrument strings in the notes view are tuned to the standard open strings for the selected virtual instrument. In some embodiments, note sliding or pitch bending can be supported when the button badge 840 is not active. An accelerometer can optionally be configured to determine the velocity (i.e., how hard the note is touched) of any articulations in notes view.

[0081] Velocity Sample Cross Fades

[0082] According to certain embodiments, the virtual instruments described herein utilize a cross-fade of two channels comprising a loud sample and a soft sample to create notes of varying velocities (i.e., amplitudes, volume, etc.). Lower velocity audio signals created by the virtual instruments may have a higher weighted value for the soft sample, and higher velocity audio signals may have a higher weighted value for the loud sample. The resultant composite sample creates a more realistic sounding virtual instrument that more accurately reproduces the various aural characteristics and harmonic structures of notes played at varying degrees of amplitude.

[0083] FIG. 9 depicts a graph 900 illustrating the cross-fading of two samples of a virtual instrument, according to an embodiment of the invention. Graph 900 includes an envelope level axis 905, a sample output level axis 910, a soft sample 920, and a loud sample 930. As described above, a legato swipe gesture can trigger playback of two preselected MIDI-controlled audio samples for a virtual instrument, where the velocity of the swipe can control characteristics of the cross-fade and the amplitude of the composite signal. In some aspects, the soft sample 920 can be a first preselected audio sample that includes an audio sample of a bowed stringed instrument having a soft attack and a light timbre. The loud sample 930 can be a second preselected audio sample including an audio sample of a bowed stringed instrument having a strong attack and a heavy timbre. As the velocity of the swipe gesture increases, the playback (i.e., envelope level) of the two preselected audio samples cross-fades from the soft sample 920 (i.e., first preselected MIDI file) to the loud sample 930 (i.e., second preselected MIDI file). In contrast, as the velocity of the swipe gesture decreases, the playback of the two preselected audio samples cross-fades from the loud sample 930 to the soft sample 920. In FIG. 9, at an envelope level (e.g., swipe velocity) of 75 out of 128 levels, the loud sample 930 cross-fades at a higher output level than the soft signal 920. In some embodiments, the playback volume of the cross-faded composite of the two preselected audio samples increases in response to an increased velocity of a legato swipe gesture. In contrast, the playback volume of the cross-faded composite of the two preselected audio samples decreases in response to a decreased velocity of a legato swipe gesture.

[0084] FIG. 10 illustrates a stereo field mapping configuration 1000 of a cross-faded composite sample, according to an embodiment of the invention. In each MIDI channel (e.g., virtual instrument), the resulting composite mix of the soft sample and loud sample is summed to a mono signal and panned in a mixer to present a particular stereo spread of the five MIDI channel instruments (e.g., the violins, violas, cellos, and basses) in a stereo field. In other cases, a stereo signal is placed within the stereo field for the final output. In this particular example, the 1st violin, 2nd violin, violas, and basses are enabled and the cellos are disabled. The resulting mix of soft samples and loud samples for each instrument (i.e., channel) is summed, panned, and added to a master stereo output for playback. In alternative embodiments, additional expression data can be modulated with the cross-faded signals. For example, as low frequency notes are played, haptic feedback (i.e., vibration) can be cross-faded into the composite stereo signal to simulate the rumble of bass frequencies. Similarly,

other expression data (e.g., equalization data) can be cross-faded into the output signal as would be appreciated by one of ordinary skill in the art with the benefit of this disclosure.

[0085] Dual Fly Wheel Aspect for Modeling a Physical System Response to a User Input

[0086] The legato gesture articulation (i.e., tap-and-swipe) provides a realistic sounding bowing action on a stringed instrument due, in part, to a virtual mass modeling system. The mass modeling system can include two series connected signal processors that can be analogized to a virtual dual flywheel system having mass and momentum to create smooth transitions between notes and chords by modeling a virtual momentum created by a user swiping gesture on the interface 100 and maintaining that momentum through subsequent harmonic or melodic progressions.

[0087] FIG. 11 depicts a mass modeling system (“flywheel system”) 1100, according to an embodiment of the invention. The modeling system 1100 includes a first signal processing stage 1110 with a first input and a first output, and a second signal processing stage 1120 receiving the first output and having a second output. The first user input can include the legato gesture articulation (e.g., up-and-down swiping gesture) for a virtual instrument on the interface 100. The first signal processing stage 1110 is coupled to the second signal processing stage 1120 such that the second input of the second signal processing stage 1120 receives the output of the first signal processing stage 1110. The output of the second signal processing stage 1120 can determine an output expression level for the system 100 (e.g., a volume) in response to the first user input. In some embodiments, the output of the flywheel system 1100 is configured to modulate the lifecycle of a note 1130 played by a virtual instrument. The output of both the first and second signal processing stages 1110, 1120 can be modeled by equation (1):

$$L = (((i*(1-h)) + (L'*h)) * F_s) + C \quad (1)$$

[0088] The various components of equation 1 are defined as follows: L is current level, speed, or momentum of the given flywheel, and L' is the previous level, speed, or momentum of the given flywheel. The external input ‘i’ can be a physical user input (e.g., a legato swiping motion on a touch zone) for the first signal processing stage 1110. Alternatively, the external input ‘i’ can be the output L of the first signal processing stage 1110 as the input to the second signal processing stage 1120. The ‘h’ variable is a mass coefficient associated with each flywheel and can vary due to a state of the second flywheel (e.g., momentum). ‘F’

can be a scaling factor and 'C' can be an offset constant. The first term ($i * (1-h)$) corresponds to an influence of a new input momentum in the system. The second term ($L' * h$) corresponds to the existing momentum of the system. As can be seen in equation 1, a very high mass coefficient (e.g., 0.9) will require very high input levels to make any appreciable change to the existing momentum. Conversely, very small mass coefficients (e.g., 0.1) will only require small input levels to appreciably change the existing momentum of the particular flywheel. By coupling two virtual flywheels together, the output of the second signal processing stage 1120 depends on a combination of the input from the first signal processing stage 1110 and the current momentum of the second signal processing stage 1120. In an alternative aspect, the momentum can be described as a rectification and low-pass filter of the input signal (e.g., up-and-down swipe gesture over time).

[0089] An aspect of the flywheel system 1100 is to determine a resulting expression level (e.g., volume) that produces a realistic sounding transition from one note or chord to the next by maintaining and modulating a virtual momentum developed in the flywheel system 1100.

To illustrate this concept, a user may initiate a very soft legato articulation on a touch zone by slowing swiping in an up-and-down motion (e.g., a slow first swipe speed) and gradually increasing speed until a relatively fast and constant rate is achieved (e.g., a fast second swipe speed). This can create the sound of a musical crescendo with the relative volume (i.e., expression data) of the virtual instrument increasing until the volume is commensurate with the momentum created by the constant rate of the input signal. Over a period of time, both signal processing stages 1110, 1120 (i.e., virtual flywheels) will be virtually rotating with the same momentum (e.g., the fast second swipe speed). As the user moves to a different touch zone (e.g., a different chord) and begins a swipe gesture at a slower constant swipe speed (e.g., a third medium swipe speed), the flywheel system 1100 would begin playback of the new chord with a strong volume corresponding to the second fast swipe speed and slowly reduce to a lower volume corresponding to the third medium swipe speed. The rate of change can be dictated by the various terms and parameters of equation 1 that define the virtual mass effect (i.e., momentum) of the flywheel system 1100. It should be noted that the flywheel system 1100 can be applied to many applications other than modeling a lifecycle of a note or chord. For example, the flywheel system 1100 could control drawing/painting applications where the intensity (e.g., color, width of lines, straightness of lines, line characteristics) is modulated based on the output of the flywheel system 1100. Alternatively, the flywheel system 1100 can be configured to control a smart drum machine that alters beat

parameters (e.g., beat complexity, fill complexity, sample selection, volume, etc.) based on the output of the flywheel system 1100 to provide a continuously evolving drum track. It should be known that the application of the flywheel system 1100 is not limited to musical applications and may be adapted to many other systems well-suited for continuous modulation and control by a virtual momentum based system.

[0090] FIG. 12 illustrates an embodiment of a note lifecycle 1200, according to an embodiment of the invention. The note lifecycle can be measured in a relative level 1210 (e.g., volume) versus time 1220. The note life cycle 1200 can include an attack phase 1230, a sustain phase 1240, and a release phase 1250. In some embodiments, the note lifecycle can include additional phases including a decay phase or any other characteristic that can be used to model the life cycle of a musical note. The foregoing note lifecycle phases would be readily understood by those of ordinary skill in the art.

[0091] FIG. 13 is a simplified flow diagram illustrating aspects of a method 1300 of modeling a lifecycle of a note played on a virtual instrument, according to an embodiment of the invention. The method 1300 is performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, etc.), software (which as is run on a general purpose computing system or a dedicated machine), firmware (embedded software), or any combination thereof. The method 1300 is discussed in terms of the dual flywheel physical system model 1100 (FIG. 11) as applied to the exemplary system 700 (FIG. 7), which is configured to execute the method 1300.

[0092] Referring to FIG. 13, the method 1300 includes receiving a touch down and motion touch gesture (e.g., a legato gesture) on a touch zone 760 of an interface 700 (1310). At 1320, the attack phase of the note or chord begins. In some embodiments, the level (e.g., volume) and rate of attack ($\Delta L/\Delta t$) is based, in part, on a mass coefficient (h_a) of the dual flywheel physical system model ("flywheel system") 1100 and the input signal (e.g., legato articulation). After a first predetermined period of time (1330), the attack phase expires. The predetermined period of time for the attack phase 1330 can be a time constant of any suitable length as would be appreciated by one of ordinary skill in the art. At 1340, the note enters the sustain phase. The length and rate of decay ($\Delta L/\Delta t$) of the sustain phase is based, in part, on a mass coefficient (h_s) of the flywheel system 1100 and the input signal (e.g., speed of the legato articulation). If the input (e.g., legato gesture) stops for a second predetermined period of time (1350), the sustain phase 1240 expires and the note enters the release phase 1360.

The level (e.g., volume) of the release phase decays according to a predetermined time constant. The length and rate of decay ($\Delta L/\Delta t$) of the release phase is based, in part, on a mass coefficient (h_g) of the flywheel system 1100. At 1370, a lift-off condition occurs and the method ends.

5 [0093] In certain embodiments, the flywheel system 1100 provides a mechanism to modulate the level and phase of a note lifecycle in real time. As described above, the sustain phase 1340 gradually decreases and eventually enters the release phase 1360 in response to a static input 1350. For example, a level (e.g., output of the second virtual flywheel) of the sustain phase 1340 may decrease below a predetermined value and enter the release phase
10 1360. In some embodiments, the sustain phase 1340 can be maintained indefinitely provided that there is a continuous input signal 1310. Furthermore, increasing the speed of a swipe gesture can proportionally increase a current level of the sustain phase 1340 in accordance with the virtual momentum characteristics of the flywheel system 1100. In some embodiments, the note lifecycle can return (1355) from the release phase 1360 back to the
15 sustain phase 1340 in response to a sufficiently high input signal (e.g., swipe speed). It should be noted that the “level” described herein can refer to a volume. In addition, level can optionally refer to a relative “momentum” created by the flywheel system 1100, where the different phases (e.g., attack, sustain, release) correspond to a level.

[0094] In alternative embodiments, other input signals may be used to modulate the note
20 lifecycle, the flywheel system 1100, or a combination of the two. Other user input signals can include the size of the touch point and/or the acceleration therewith (i.e., detect the pressure or downward force of the input gesture), the angle of the touch signal, multiple touch signals, or any other input signal that would be appreciated by one of ordinary skill in the art with the benefit of this disclosure.

25 [0095] It should be appreciated that the specific steps illustrated in FIG. 13 provides a particular method of modeling a lifecycle of a note played on a virtual instrument, according to an embodiment of the present invention. Other sequences of steps may also be performed according to alternative embodiments. In certain embodiments, the method 1300 may perform the individual steps in a different order, at the same time, or any other sequence for a
30 particular application. Moreover, the individual steps illustrated in FIG. 13 may include multiple sub-steps that may be performed in various sequences as appropriate to the individual step. Furthermore, additional steps may be added or removed depending on the

particular applications. One of ordinary skill in the art would recognize and appreciate many variation, modification, and alternatives of the method.

[0096] FIG. 21 is a simplified flow diagram illustrating aspects of a method 2100 of modulating a lifecycle of a note, according to an embodiment of the invention. Method 2100 is performed by processing logic that may comprise hardware (e.g., circuitry, dedicate logic, etc.), software (which as is run on a general purpose computing system or a dedicated machine), firmware (embedded software), or any combination thereof. In one embodiment, method 2100 is performed by the processor unit 1410 of FIG. 14.

[0097] Referring to FIG. 21, method 2100 begins with generating a user interface on a touch sensitive display, at 2110. The touch sensitive display can be similar to the user interface 100 for a virtual musical instrument, as shown in FIG. 1. The user interface can be implemented on any suitable touch sensitive device configured to detect touch input. At 2120, method 2100 further includes generating a first virtual flywheel system (VFS) 1110. The first virtual flywheel system 1110 can include an input and an output, where the input may include a user touch input on the user interface, or any suitable data input (e.g., input signal) from any desired source, as would be appreciated by one of ordinary skill in the art. The output of the first virtual flywheel system can include a data value that may be an expression value, virtual momentum, or other suitable data output that is based, at least in part, on the data input and a mass coefficient associated with the first virtual flywheel. The mass coefficient of the VFS 1110 may be predetermined, a fixed value, a dynamic value, or any suitable value as required. In some cases, the output of the first virtual flywheel 1110 is the product of a magnitude of the data input and the mass coefficient.

[0098] At 2130, method 2100 further includes generating a second virtual flywheel system 1120. The second virtual flywheel system (VFS) 1120 can include an input and an output. In some embodiments, the second virtual flywheel system 1120 is series connected with the first virtual flywheel system 1100, such that the output of the first virtual flywheel system 1110 is coupled to the input of the second virtual flywheel system 1120. In some cases, the input to the second virtual flywheel system 1120 may include additional sources (e.g., multiple inputs). The output of the second VFS 1120 can be based wholly, or in part, on the input to the VFS 1120 and a mass coefficient associated with the second VFS 1120. The mass coefficient of the VFS 1120 may be predetermined, a fixed value, a dynamic value, or any suitable value as required. In some cases, the output of the first virtual flywheel 1120 is the

product of a magnitude of the data input of the second VFS 1120 and its associated mass coefficient.

[0099] At 2140, the user interface receives a user input. The user input can include a touch gesture, swipe gesture, or any suitable input from a user, stylus, or the like, on a touch-sensitive display. Alternatively, the input may be generated by systems other than the touch sensitive variety. At 2150, method 2100 further includes determining a momentum of the first VFS 1110. As described above, the momentum may be based on the input to the FVS 1110 and a mass coefficient associated with the VFS 1110. At 2160, method 2100 further includes determining a momentum of the second VFS 1110. As described above, the momentum of the second VFS 1120 may be based on an input signal to the second VFS 1120 and its associated mass coefficient. The output of the VFS 1120 can be utilized as an expression level, an output level, a dynamic control signal, or the like, as would be appreciated by one of ordinary skill in the art.

[0100] At 2170, method 2100 further includes applying the virtual momentum of the second VFS 1120 to modulate a lifecycle of a note or chord. In some embodiments, the output of the VFS 1120 (e.g., expression level) can be used to modulate one or more audio properties of an audio signal. For example, a volume of the audio signal can be controlled in real-time based on a magnitude of the expression level. The output of the VFS 1120 can be used to modulate any system including those not related to audio applications. For example, the expression level may be used to control or modulate visual properties (e.g., colors, intensity, contrast, tone) of a system. In broader contexts, the output of the VFS 1120 can be used to modulate in real-time any desired parameter in any suitable application.

[0101] It should be appreciated that the specific steps illustrated in FIG. 21 provides a particular method of modulating a lifecycle of a note, according to an embodiment of the present invention. Other sequences of steps may also be performed according to alternative embodiments. In certain embodiments, method 2100 may perform the individual steps in a different order, at the same time, or any other sequence for a particular application. Moreover, the individual steps illustrated in FIG. 21 may include multiple sub-steps that may be performed in various sequences as appropriate to the individual step. Furthermore, additional steps may be added or removed depending on the particular applications. One of ordinary skill in the art would recognize and appreciate many variation, modification, and alternatives of the method.

[0102] FIG. 15 illustrates a computer system 1500 according to an embodiment of the present invention. The user interfaces described herein (e.g., interface 100 and 700) can be implemented within a computer system such as computer system 1500 shown here.

Computer system 1500 can be implemented as any of various computing devices, including, e.g., a desktop or laptop computer, tablet computer, smart phone, personal data assistant (PDA), or any other type of computing device, not limited to any particular form factor. Computer system 1500 can include processing unit(s) 1505, storage subsystem 1510, input devices 1520, output devices 1525, network interface 1535, and bus 1540.

[0103] Processing unit(s) 1505 can include a single processor, which can have one or more cores, or multiple processors. In some embodiments, processing unit(s) 1505 can include a general purpose primary processor as well as one or more special purpose co-processors such as graphics processors, digital signal processors, or the like. In some embodiments, some or all processing units 1505 can be implemented using customized circuits, such as application specific integrated circuits (ASICs) or field programmable gate arrays (FPGAs). In some embodiments, such integrated circuits execute instructions that are stored on the circuit itself. In other embodiments, processing unit(s) 1505 can execute instructions stored in storage subsystem 1510.

[0104] Storage subsystem 1510 can include various memory units such as a system memory, a read-only memory (ROM), and a permanent storage device. The ROM can store static data and instructions that are needed by processing unit(s) 1505 and other modules of electronic device 1500. The permanent storage device can be a read-and-write memory device. This permanent storage device can be a non-volatile memory unit that stores instructions and data even when computer system 1500 is powered down. Some embodiments of the invention can use a mass-storage device (such as a magnetic or optical disk or flash memory) as a permanent storage device. Other embodiments can use a removable storage device (e.g., a floppy disk, a flash drive) as a permanent storage device. The system memory can be a read-and-write memory device or a volatile read-and-write memory, such as dynamic random access memory. The system memory can store some or all of the instructions and data that the processor needs at runtime.

[0105] Storage subsystem 1510 can include any combination of computer readable storage media including semiconductor memory chips of various types (DRAM, SRAM, SDRAM, flash memory, programmable read-only memory) and so on. Magnetic and/or optical disks

can also be used. In some embodiments, storage subsystem 1510 can include removable storage media that can be readable and/or writeable; examples of such media include compact disc (CD), read-only digital versatile disc (e.g., DVD-ROM, dual-layer DVD-ROM), read-only and recordable Blue-Ray® disks, ultra density optical disks, flash memory cards
5 (e.g., SD cards, mini-SD cards, micro-SD cards, etc.), magnetic “floppy” disks, and so on. The computer readable storage media do not include carrier waves and transitory electronic signals passing wirelessly or over wired connections.

[0106] In some embodiments, storage subsystem 1510 can store one or more software programs to be executed by processing unit(s) 1505, such as a user interface 1515. As
10 mentioned, “software” can refer to sequences of instructions that, when executed by processing unit(s) 1505 cause computer system 1500 to perform various operations, thus defining one or more specific machine implementations that execute and perform the operations of the software programs. The instructions can be stored as firmware residing in read-only memory and/or applications stored in magnetic storage that can be read into
15 memory for processing by a processor. Software can be implemented as a single program or a collection of separate programs or program modules that interact as desired. Programs and/or data can be stored in non-volatile storage and copied in whole or in part to volatile working memory during program execution. From storage subsystem 1510, processing unit(s) 1505 can retrieve program instructions to execute and data to process in order to
20 execute various operations described herein.

[0107] A user interface can be provided by one or more user input devices 1520, display device 1525, and/or one or more other user output devices (not shown). Input devices 1520 can include any device via which a user can provide signals to computing system 1500; computing system 1500 can interpret the signals as indicative of particular user requests or
25 information. In various embodiments, input devices 1520 can include any or all of a keyboard touch pad, touch screen, mouse or other pointing device, scroll wheel, click wheel, dial, button, switch, keypad, microphone, and so on.

[0108] Output devices 1525 can display images generated by electronic device 1500. Output devices 1525 can include various image generation technologies, e.g., a cathode ray
30 tube (CRT), liquid crystal display (LCD), light-emitting diode (LED) including organic light-emitting diodes (OLED), projection system, or the like, together with supporting electronics (e.g., digital-to-analog or analog-to-digital converters, signal processors, or the

like), indicator lights, speakers, tactile “display” devices, headphone jacks, printers, and so on. Some embodiments can include a device such as a touchscreen that function as both input and output device.

[0109] In some embodiments, output device 1525 can provide a graphical user interface, in which visible image elements in certain areas of output device 1525 are defined as active elements or control elements that the user selects using user input devices 1520. For example, the user can manipulate a user input device to position an on-screen cursor or pointer over the control element, then click a button to indicate the selection. Alternatively, the user can touch the control element (e.g., with a finger or stylus) on a touchscreen device.

In some embodiments, the user can speak one or more words associated with the control element (the word can be, e.g., a label on the element or a function associated with the element). In some embodiments, user gestures on a touch-sensitive device can be recognized and interpreted as input commands; these gestures can be but need not be associated with any particular array in output device 1525. Other user interfaces can also be implemented.

[0110] Network interface 1535 can provide voice and/or data communication capability for electronic device 1500. In some embodiments, network interface 1535 can include radio frequency (RF) transceiver components for accessing wireless voice and/or data networks (e.g., using cellular telephone technology, advanced data network technology such as 3G, 4G or EDGE, WiFi (IEEE 802.11 family standards, or other mobile communication technologies, or any combination thereof), GPS receiver components, and/or other components. In some embodiments, network interface 1535 can provide wired network connectivity (e.g., Ethernet) in addition to or instead of a wireless interface. Network interface 1535 can be implemented using a combination of hardware (e.g., antennas, modulators/demodulators, encoders/decoders, and other analog and/or digital signal processing circuits) and software components.

[0111] Bus 1540 can include various system, peripheral, and chipset buses that communicatively connect the numerous internal devices of electronic device 1500. For example, bus 1540 can communicatively couple processing unit(s) 1505 with storage subsystem 1510. Bus 1540 also connects to input devices 1520 and display 1525. Bus 1540 also couples electronic device 1500 to a network through network interface 1535. In this manner, electronic device 1500 can be a part of a network of multiple computer systems (e.g., a local area network (LAN), a wide area network (WAN), an Intranet, or a network of

networks, such as the Internet. Any or all components of electronic device 1500 can be used in conjunction with the invention.

[0112] Some embodiments include electronic components, such as microprocessors, storage and memory that store computer program instructions in a computer readable storage medium. Many of the features described in this specification can be implemented as processes that are specified as a set of program instructions encoded on a computer readable storage medium. When these program instructions are executed by one or more processing units, they cause the processing unit(s) to perform various operation indicated in the program instructions. Examples of program instructions or computer code include machine code, such as is produced by a compiler, and files including higher-level code that are executed by a computer, an electronic component, or a microprocessor using an interpreter.

[0113] It will be appreciated that computer system 1500 is illustrative and that variations and modifications are possible. Computer system 1500 can have other capabilities not specifically described here (e.g., mobile phone, global positioning system (GPS), power management, one or more cameras, various connection ports for connecting external devices or accessories, etc.). Further, while computer system 1500 is described with reference to particular blocks, it is to be understood that these blocks are defined for convenience of description and are not intended to imply a particular physical arrangement of component parts. Further, the blocks need not correspond to physically distinct components. Blocks can be configured to perform various operations, e.g., by programming a processor or providing appropriate control circuitry, and various blocks might or might not be reconfigurable depending on how the initial configuration is obtained. Embodiments of the present invention can be realized in a variety of apparatus including electronic devices implemented using any combination of circuitry and software.

[0114] While the invention has been described with respect to specific embodiments, one skilled in the art will recognize that numerous modifications are possible: displaying the interface 100/700 and the configuration of the various elements therein, such as the position of touch regions and touch zones, the types of virtual instruments used, the types of chord inversions applied to the touch zones, which may be customizable, customizing aspects of the flywheel system, etc. Thus, although the invention has been described with respect to specific embodiments, it will be appreciated that the invention is intended to cover all modifications and equivalents within the scope of the following claims.

[0115] The above disclosure provides examples and aspects relating to various embodiments within the scope of claims, appended hereto or later added in accordance with applicable law. However, these examples are not limiting as to how any disclosed aspect may be implemented,

- 5 [0116] All the features disclosed in this specification (including any accompanying claims, abstract, and drawings) can be replaced by alternative features serving the same, equivalent or similar purpose, unless expressly stated otherwise. Thus, unless expressly stated otherwise, each feature disclosed is one example only of a generic series of equivalent or similar features.
- 10 [0117] Any element in a claim that does not explicitly state “means for” performing a specified function, or “step for” performing a specific function, is not to be interpreted as a “means” or “step” clause as specified in 35 U.S.C. § 112, sixth paragraph. In particular, the use of “step of” in the claims herein is not intended to invoke the provisions of 35 U.S.C. § 112, sixth paragraph.

WHAT IS CLAIMED IS:

1 1. A user interface implemented on a touch-sensitive display for a virtual
2 musical instrument comprising:
3 a plurality of chord touch regions, each chord touch region corresponding to a
4 chord in a musical key and being divided into a plurality of separate touch zones, the plurality
5 of chord touch regions defining a predetermined set of chords,
6 wherein each of the plurality of separate touch zones in each region is
7 associated with one or more preselected audio signals stored in a computer-readable medium,
8 and
9 wherein each of the plurality of touch zones is configured to detect and
10 distinguish between a plurality of touch gesture articulations, wherein a detected touch
11 gesture articulation determines the preselected audio signals associated with each of the
12 plurality of separate touch zones.

1 2. The user interface of claim 1 wherein the one or more of the plurality
2 of touch gesture articulations includes at least one of a legato gesture articulation, a pizzicato
3 gesture articulation, or a staccato gesture articulation.

1 3. The user interface of claim 2 wherein the legato articulation includes a
2 tap to swipe gesture, wherein the legato articulation triggers playback of a first corresponding
3 audio signal that simulates the sound of a bowing action on a stringed instrument.

1 4. The user interface of claim 3 wherein a velocity of the legato
2 articulation controls an expression value of the first corresponding audio signal.

1 5. The user interface of claim 4 wherein the expression value of the first
2 corresponding audio signal is an amplitude.

1 6. The user interface of claim 3 wherein playback of the first
2 corresponding audio signal stops when the legato articulation is no longer detected or a
3 second touch gesture articulation is simultaneously detected on the user interface.

1 7. The user interface of claim 3 wherein a sustained legato articulation
2 includes a tap to continuous up-and-down swipe gesture on the user interface.

1 8. The user interface of claim 2 wherein the pizzicato articulation
2 includes a quick tap gesture comprising an initial tap in one of the plurality of separate touch
3 zones and a release prior to a predetermined period, wherein the pizzicato articulation triggers
4 playback of a second corresponding audio signal that simulates the sound of a plucking action
5 on a stringed instrument.

1 9. The user interface of claim 8 wherein a velocity of the pizzicato
2 articulation controls an expression value of the second corresponding audio signal.

1 10. The user interface of claim 9 wherein the expression value of the
2 second corresponding audio signal is an amplitude.

1 11. The user interface of claim 2 wherein the staccato articulation includes
2 an in-motion swipe gesture upon contact with the touch zone, wherein the staccato
3 articulation triggers playback of a third corresponding audio signal that simulates the sound
4 of a short burst of a bowing action on a stringed instrument.

1 12. The user interface of claim 11 wherein a velocity of the staccato
2 articulation controls an expression value of the third corresponding audio signal.

1 13. The user interface of claim 12 wherein the expression value of the third
2 corresponding audio signal is an amplitude.

1 14. A computer program product stored on a non-transitory computer-
2 readable storage medium comprising computer-executable instructions causing a processor
3 to:

4 in response to input from a user interface implemented on a touch-sensitive
5 display for a virtual musical instrument, comprising a plurality of chord touch regions
6 configured in a predetermined sequence, each chord touch region corresponding to a chord in
7 a musical key and being divided into a number of separate touch zones, the plurality of chord
8 regions defining a predetermined set of chords, cause a preselected audio signal to be played
9 on an output device, wherein said input from a user comprises one of a plurality of touch
10 gesture articulations on a touch zone associated with said preselected audio signal stored in a
11 computer-readable medium.

1 15. The computer program product of claim 14 wherein the one of a
2 plurality of touch gesture articulations includes at least one of a legato articulation, a
3 pizzicato articulation, or a staccato articulation.

1 16. The computer program product of claim 15 wherein the legato
2 articulation includes a tap to swipe gesture, wherein the legato articulation triggers playback
3 of a first preselected audio signal that simulates the sound of a bowing action on a stringed
4 instrument,

5 wherein the pizzicato articulation includes a quick tap gesture comprising an
6 initial tap in one of the plurality of separate touch zones and a release prior to a
7 predetermined period, wherein the pizzicato articulation triggers playback of a second
8 preselected audio signal that simulates the sound of a plucking action on the stringed
9 instrument, and

10 wherein the staccato articulation includes an in-motion swipe gesture upon
11 contact with the touch zone, wherein the staccato articulation triggers playback of a third
12 preselected audio signal that simulates the sound of a short burst of a bowing action on the
13 stringed instrument.

1 17. The computer program product of claim 16 wherein a velocity of the
2 legato articulation controls an expression value of the first preselected audio signal, a velocity
3 of a pizzicato articulation controls an expression value of the second preselected audio signal,
4 and a velocity of the staccato articulation controls an expression value of the third preselected
5 audio signal.

1 18. A method of determining a characteristic of a note played on a virtual
2 stringed instrument, the method comprising:

3 receiving, from a user interface on a touch sensitive display for the virtual
4 instrument, a user input comprising one of a plurality of touch gesture articulations performed
5 on a touch zone of the user interface;

6 determining if the one of a plurality of touch gesture articulations includes one
7 of a legato articulation, a pizzicato articulation, and a staccato articulation; and

8 playing one of a plurality of preselected audio signals on an output device
9 corresponding to both the touch region and the one of a plurality of gesture articulations.

1 19. The method of claim 18 wherein the legato articulation includes a tap
2 to swipe gesture on the touch zone, wherein the legato articulation triggers playback of a first
3 audio signal of the plurality of preselected audio signals that simulates the sound of a bowing
4 action on a stringed instrument,

5 wherein the pizzicato articulation includes a quick tap gesture comprising an
6 initial tap in the touch zone and a release prior to a predetermined period, wherein the
7 pizzicato articulation triggers playback of a second audio signal of the plurality of preselected
8 audio signals that simulates the sound of a plucking action on the stringed instrument, and

9 wherein the staccato articulation includes an in-motion swipe gesture upon
10 contact with the touch zone, wherein the staccato articulation triggers playback of a third
11 audio signal of the plurality of preselected audio signals that simulates the sound of a short
12 burst of a bowing action on the stringed instrument.

1 20. The method of claim 19 wherein a velocity of the legato articulation
2 controls an expression value of the first audio signal, wherein a velocity of the pizzicato
3 articulation controls an expression value of the second audio signal, and wherein a velocity of
4 the staccato articulation controls an expression value of the third audio signal.

1 21. A user interface implemented on a touch-sensitive display for a virtual
2 musical instrument comprising:

3 one or more chord touch regions, each chord touch region corresponding to a
4 base chord in a musical key and being divided into a plurality of separate touch zones, each
5 of the plurality of separate touch zones in each touch region being associated with one or
6 more preselected audio signals stored in a computer-readable medium,

7 each of the plurality of separate touch zones in each touch region being
8 associated with a different harmonic configuration of the base chord of the chord touch
9 region that it belongs to, and

10 each of the plurality of touch zones being configured to detect two or more
11 touch gesture articulations.

1 22. The user interface of claim 21 wherein each of the chord touch regions
2 comprises:

3 a first touch zone corresponding to the base chord assigned to the
4 corresponding chord touch region; and

5 a second touch zone corresponding to a first inversion of the base chord
6 assigned to the corresponding chord touch region.

1 23. The user interface of claim 22 wherein the first inversion chord is
2 higher in relative pitch than the base chord.

1 24. The user interface of claim 22 wherein the first inversion chord is in
2 wider harmonic range relative to the base chord.

1 25. The user interface of claim 21 wherein each of the chord touch regions
2 comprises:

3 a first touch zone corresponding to a first chord voicing of the base chord
4 assigned to the corresponding chord touch region; and

5 a second touch zone corresponding to second chord voicing of the base chord
6 assigned to the corresponding chord touch region, wherein the second chord voicing includes
7 a wider harmonic range than the first chord voicing..

1 26. The user interface of claim 21 wherein changing between a first
2 harmonic configuration of the base chord and a second harmonic configuration of the base
3 chord results in a change by a minimum number of notes required to switch between the first
4 and second harmonic configuration of the base chord.

1 27. The user interface of claim 21 wherein changing between a first base
2 chord and a second base chord results in a change by a minimum number of notes required to
3 switch between the first and second base chord.

1 28. A computer program product stored on a non-transitory computer-
2 readable storage medium comprising computer-executable instructions causing a processor
3 to:

4 in response to input from a user interface implemented on a touch-sensitive
5 display for a virtual musical instrument, comprising a plurality of chord touch regions, each
6 chord touch region corresponding to a base chord in a musical key and being divided into a
7 number of separate touch zones, each of the number of separate touch zones in each touch
8 region corresponding with a different harmonic configuration of the base chord of the chord
9 touch region that it belongs to,

10 cause a preselected audio signal to be played on an output device, the input
11 from the user including one of a plurality of gesture articulations on a touch zone associated
12 with the preselected audio signal stored in a computer-readable medium.

1 29. The computer program product of claim 28 wherein each of the chord
2 touch regions comprises:

3 a first touch zone corresponding to the base chord assigned to the
4 corresponding chord touch region; and

5 a second touch zone corresponding to a first inversion of the base chord
6 assigned to the corresponding chord touch region.

1 30. The computer program product of claim 29 wherein the first inversion
2 chord is higher in relative pitch than the base chord.

1 31. The computer program product of claim 29 wherein the first inversion
2 chord is in wider harmonic range relative to the base chord.

1 32. The computer program product of claim 28 wherein each of the chord
2 touch regions comprises:

3 a first touch zone corresponding to a first chord voicing of the base chord
4 assigned to the corresponding chord touch region; and

5 a second touch zone corresponding to second chord voicing of the base chord
6 assigned to the corresponding chord touch region, wherein the second chord voicing includes
7 a wider harmonic range than the first chord voicing.

1 33. The computer program product of claim 28 wherein changing between
2 a first harmonic configuration of the base chord and a second harmonic configuration of the
3 base chord results in a change by a minimum number of notes required to switch between the
4 first and second harmonic configuration of the base chord.

1 34. The computer program product of claim 28 wherein changing between
2 a first base chord and a second base chord results in a change by a minimum number of notes
3 required to switch between the first and second base chord.

1 35. A computer implemented system, comprising:
2 one or more data processors;

3 one or more non-transitory computer-readable storage media containing
4 instructions configured to cause the one or more processors to perform operations including:
5 associating a base chord with one or more chord touch regions, wherein a
6 chord touch region is divided into one or more touch zones, wherein a touch zone can detect
7 two or more types of gesture articulations, and wherein each touch zone corresponds to a
8 harmonic configuration of the base chord;
9 associating an audio signal with each of the one or more touch zones;
10 generating a graphical interface implemented on a touch-sensitive display,
11 wherein the graphical interface displays a virtual musical instrument including the base
12 chord; and
13 receiving input corresponding to a gesture articulation on a touch zone,
14 wherein the gesture articulation causes the audio signal associated with the touch zone to be
15 played on an output device.

1 36. The system of claim 35 wherein each of the chord touch regions
2 comprises:
3 a first touch zone corresponding to the base chord assigned to the
4 corresponding chord touch region; and
5 a second touch zone corresponding to a first inversion of the base chord
6 assigned to the corresponding chord touch region.

1 37. The system of claim 36 wherein the first inversion chord is higher in
2 relative pitch than the base chord, wherein the second inversion chord is higher in relative
3 pitch than the first inversion chord, and wherein the third inversion chord is higher in relative
4 pitch than the second inversion chord.

1 38. The system of claim 36 wherein the first inversion chord is in wider
2 harmonic range relative to the base chord.

1 39. The system of claim 35 wherein each of the chord touch regions
2 comprises:
3 a first touch zone corresponding to a first chord voicing of the base chord
4 assigned to the corresponding chord touch region;

5 a second touch zone corresponding to second chord voicing of the base chord
6 assigned to the corresponding chord touch region, wherein the second chord voicing includes
7 a wider harmonic range than the first chord voicing.

1 40. The system of claim 35 wherein changing between a first harmonic
2 configuration of the base chord and a second harmonic configuration of the base chord results
3 in a change by a minimum number of notes required to switch between the first and second
4 harmonic configuration of the base chord.

1 41. A computer program product tangibly embodied on a non-transitory
2 computer-readable storage medium comprising computer-executable instructions causing a
3 data processing system to:

4 generate a virtual dual flywheel system including:

5 a first virtual flywheel configured to generate a first virtual momentum
6 based on a user input and a predetermined virtual mass of the first virtual flywheel;

7 and

8 a second virtual flywheel configured to generate a second virtual
9 momentum based on the first virtual momentum and a predetermined second virtual
10 mass of the second virtual flywheel;

11 generate a user interface implemented on a touch-sensitive display for a virtual
12 musical instrument, including a plurality of touch regions, each touch region corresponding to
13 a chord in a musical key and being divided into a number of separate touch zones; and

14 in response to the user input, generate an expression level based on the second
15 virtual momentum.

1 42. The computer program product of claim 41 wherein the first virtual
2 flywheel is series connected to the second virtual flywheel.

1 43. The computer program product of claim 41 further comprising
2 applying the expression level to an audio signal to modulate one or more audio properties of
3 the audio signal.

1 44. The computer program product of claim 43 wherein the one or more
2 audio properties of the audio signal includes a volume of a musical note or chord.

1 45. The computer program product of claim 43 wherein the one or more
2 audio properties of the audio signal includes a lifecycle of a musical note or chord.

1 46. The computer program product of claim 45 wherein the lifecycle of the
2 musical note includes:

3 an attack phase;

4 a sustain phase; and

5 a release phase, the sustain phase operable to enter the release phase when the
6 expression level while in the sustain phase drops below a first predetermined value, and the
7 release phase operable to return to the sustain phase in response to the expression level rising
8 above a second predetermined value.

1 47. The computer program product of claim 46 wherein the attack phase
2 corresponds to a beginning portion of the lifecycle of the musical note or chord.

1 48. The computer program product of claim 46 wherein the sustain phase
2 corresponds to a middle portion of the lifecycle of the musical note or chord.

1 49. The computer program product of claim 46 wherein the release phase
2 corresponds to an end portion of the lifecycle of the musical note or chord.

1 50. A computer-implemented method comprising:
2 generating a user interface implemented on a touch-sensitive display;
3 generating a virtual dual flywheel system including a first virtual flywheel
4 system and a second virtual flywheel system, the first virtual flywheel system series
5 connected to the second virtual flywheel system such that an output of the first virtual
6 flywheel system is coupled to an input of the second virtual flywheel system;
7 receiving a user input on the user interface;
8 determining a virtual momentum for the first virtual flywheel based on the
9 user input and a predetermined mass coefficient of the first virtual flywheel system; and
10 determining a virtual momentum for the second virtual flywheel based on the
11 virtual momentum of the first virtual flywheel system and a predetermined mass coefficient
12 of the second virtual flywheel.

1 51. The method of claim 50 wherein the user input is a swipe gesture, a
2 magnitude of the user input based in part on a speed or duration of the swipe gesture.

1 52. The method of claim 50 wherein the virtual momentum of the second
2 virtual flywheel is an output level operable to modulate a volume or musical characteristic of
3 a musical note or chord.

1 53. The method of claim 50 further comprising applying the virtual
2 momentum for the second virtual flywheel to modulate a lifecycle of a musical note.

1 54. The method of claim 53 wherein the virtual momentum of the second
2 virtual flywheel represents a level, and wherein the lifecycle of the musical note includes:
3 an attack phase,
4 a sustain phase, and
5 a release phase, the sustain phase entering the release phase when the level in
6 the sustain phase drops below a first predetermined value, the release phase returning to the
7 sustain phase in response to the level rising above a second predetermined value.

1 55. A computer-implemented system, comprising:
2 one or more processors;
3 one or more non-transitory computer-readable storage media containing
4 instructions configured to cause the one or more processors to perform operations including:
5 generate a user interface implemented on a touch-sensitive display;
6 generate a virtual dual flywheel system including a first virtual flywheel
7 system and a second virtual flywheel system, the first virtual flywheel system series
8 connected to the second virtual flywheel system such that an output of the first virtual
9 flywheel system is virtually coupled to an input of the second virtual flywheel system;
10 receive a user input on the user interface;
11 determine a virtual momentum for the first virtual flywheel based on the user
12 input and a predetermined mass coefficient of the first virtual flywheel system; and
13 determine a virtual momentum for the second virtual flywheel based on the
14 virtual momentum of the first virtual flywheel system and a predetermined mass coefficient
15 of the second virtual flywheel.

1 56. The system of claim 55 wherein the user input is a swipe gesture, a
2 magnitude of the user input based in part on a speed of the swipe gesture.

1 57. The system of claim 55 wherein the virtual momentum of the second
2 virtual flywheel is an output level operable to modulate a volume or musical characteristic of
3 a musical note or chord.

1 58. The system of claim 55 further comprising applying the virtual
2 momentum for the second virtual flywheel to modulate a lifecycle of a musical note.

1 59. The system of claim 58 wherein the virtual momentum of the second
2 virtual flywheel represents a level, and wherein the lifecycle of the musical note includes:
3 an attack phase,
4 a sustain phase, and
5 a release phase, the sustain phase entering the release phase when the level in
6 the sustain phase drops below a first predetermined value, the release phase returning to the
7 sustain phase in response to the level rising above a second predetermined value.

1 60. The computer program product of claim 59 wherein the attack phase
2 corresponds to a beginning portion of the lifecycle of the musical note or chord, the sustain
3 phase corresponds to a middle portion of the lifecycle of the musical note or chord, and the
4 release phase corresponds to an end portion of the lifecycle of the musical note or chord.

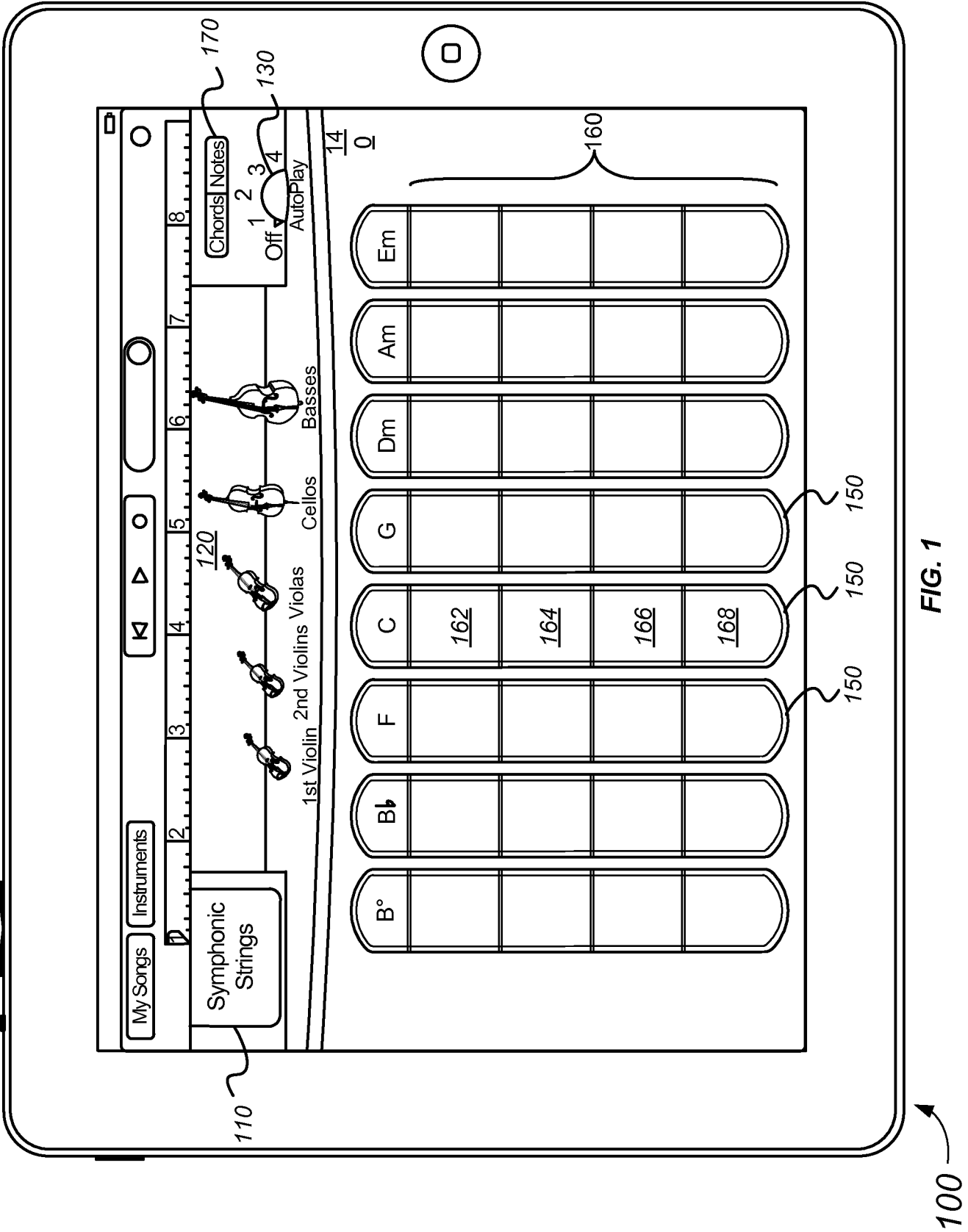


FIG. 1

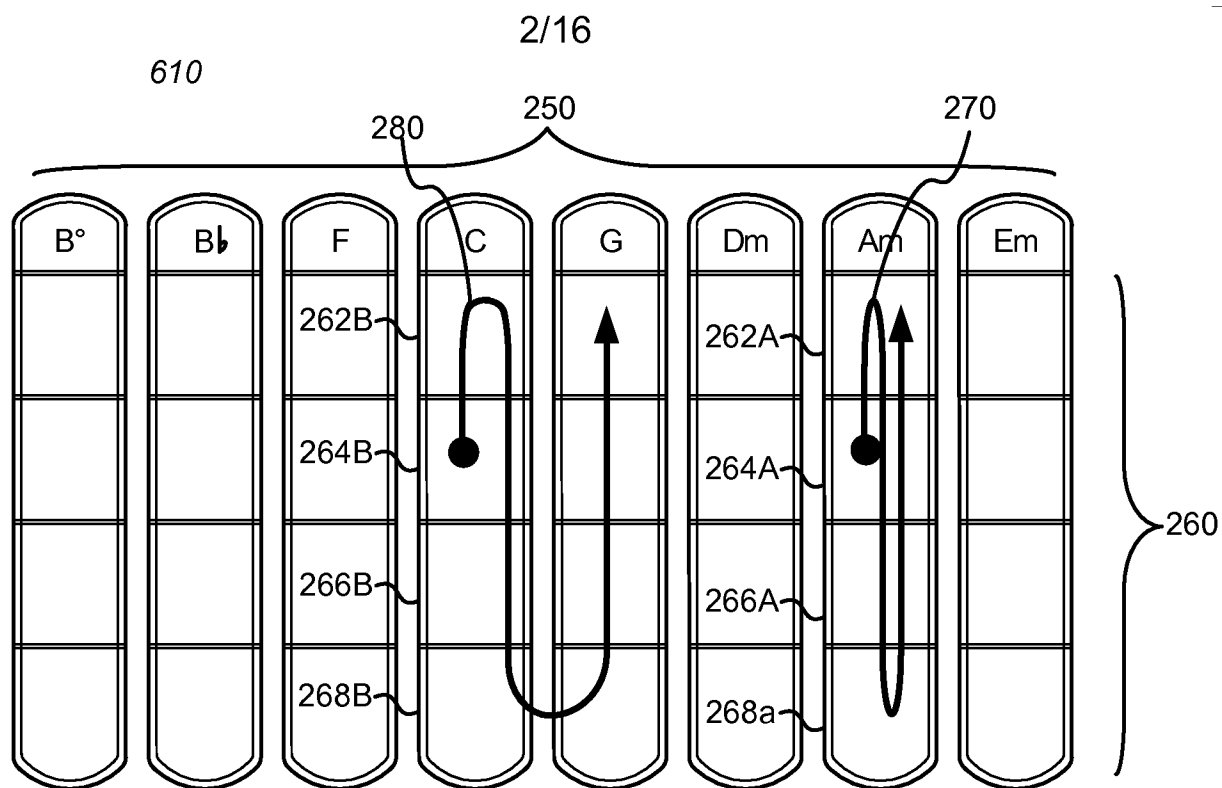


FIG. 2

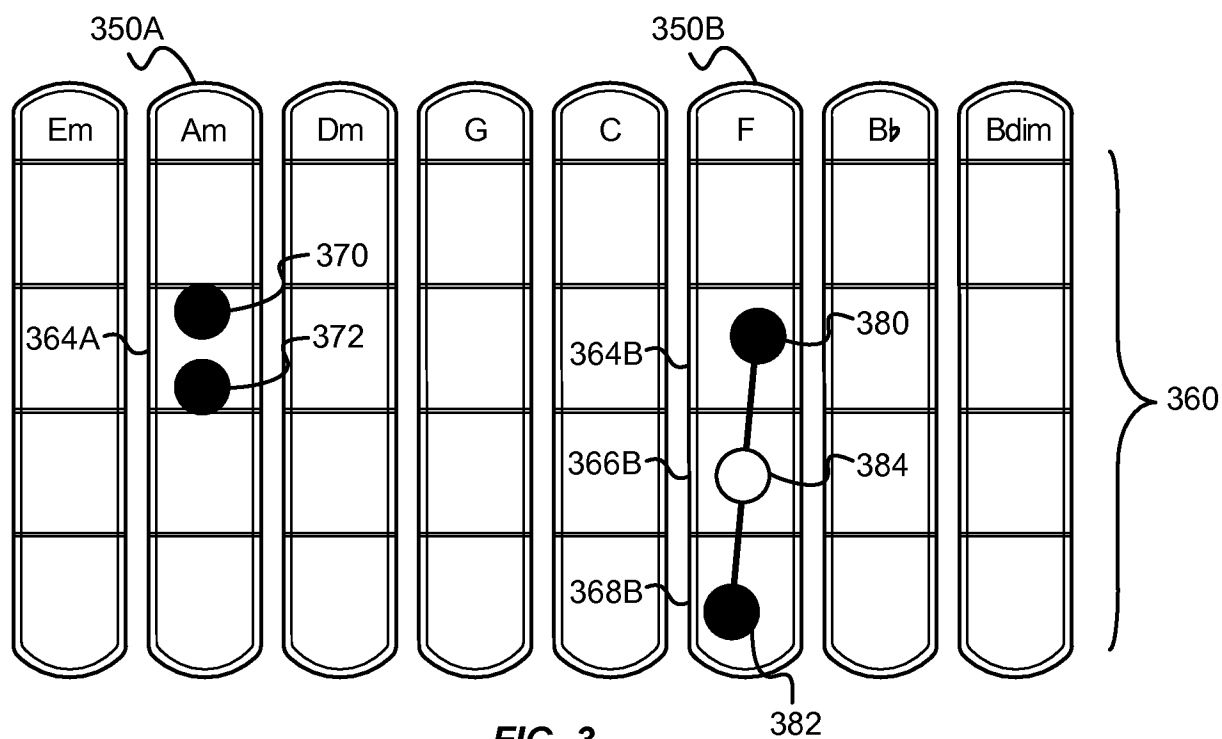


FIG. 3

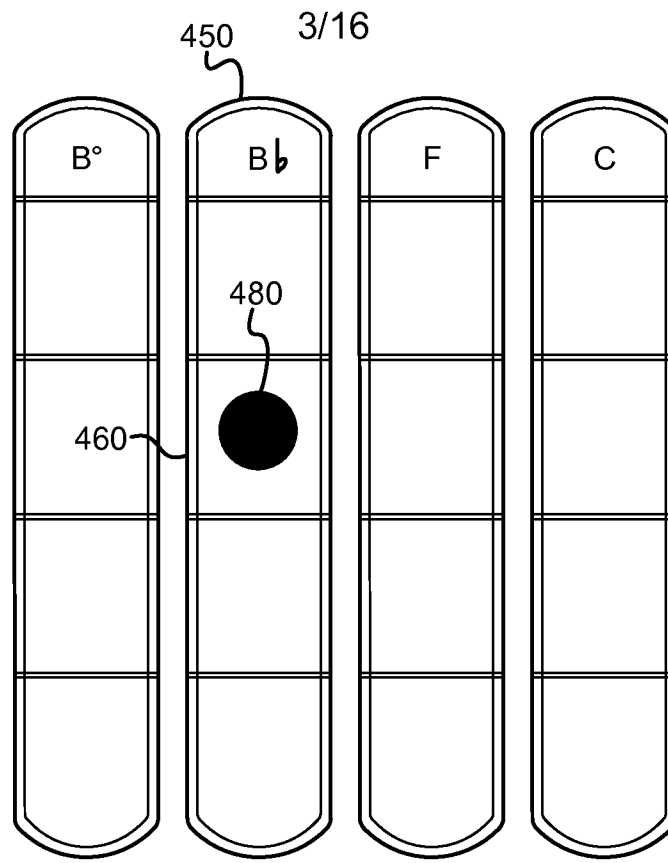


FIG. 4

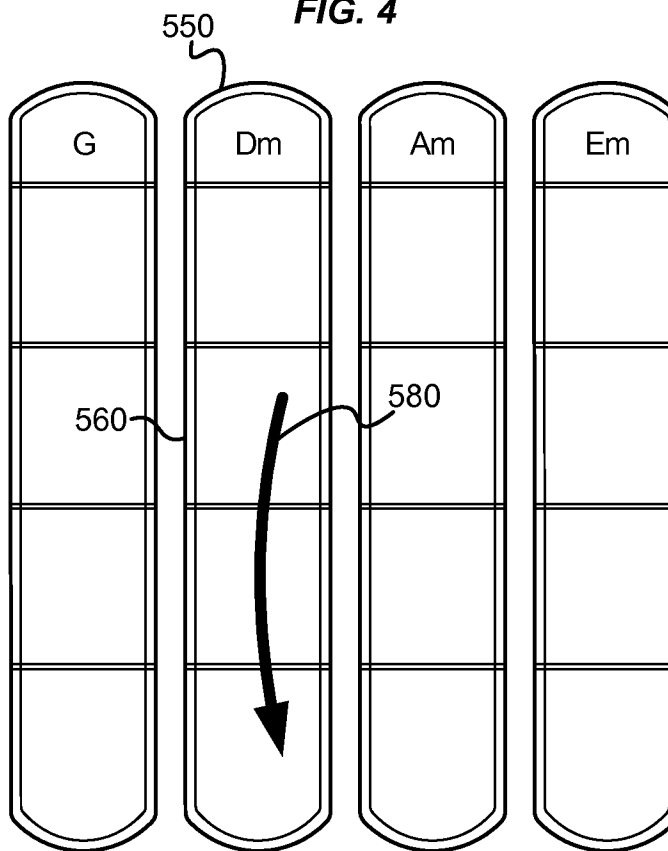
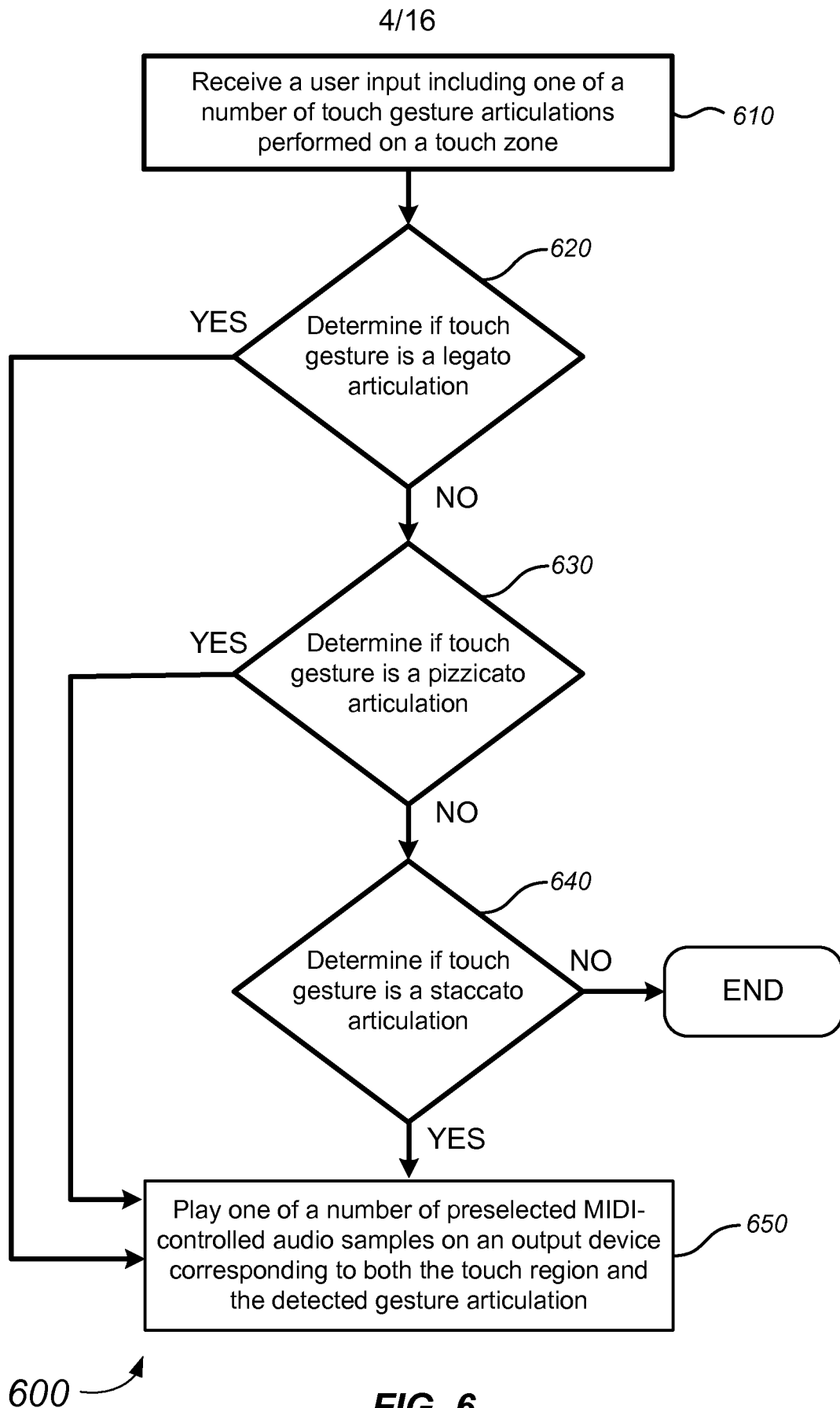


FIG. 5



5/16

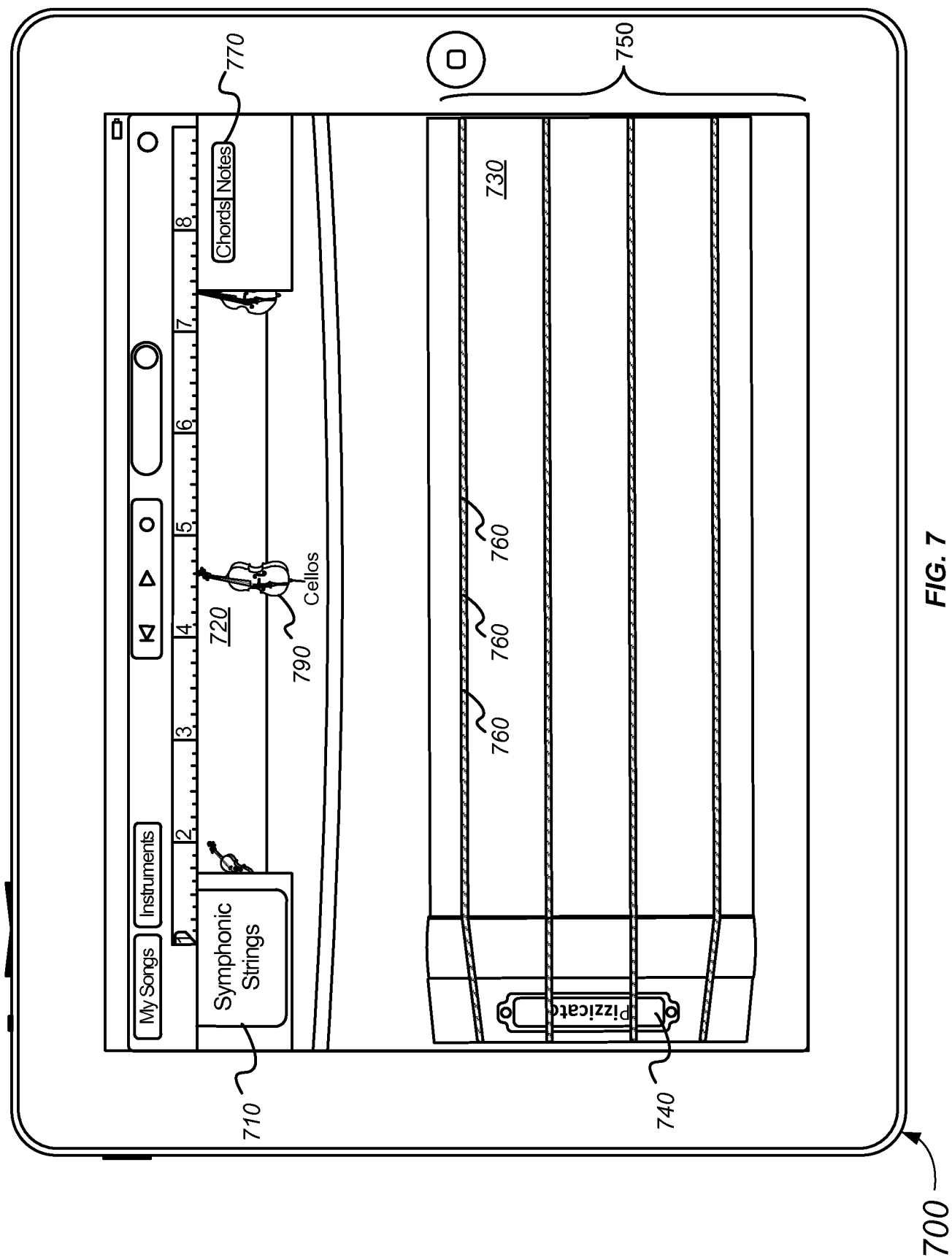
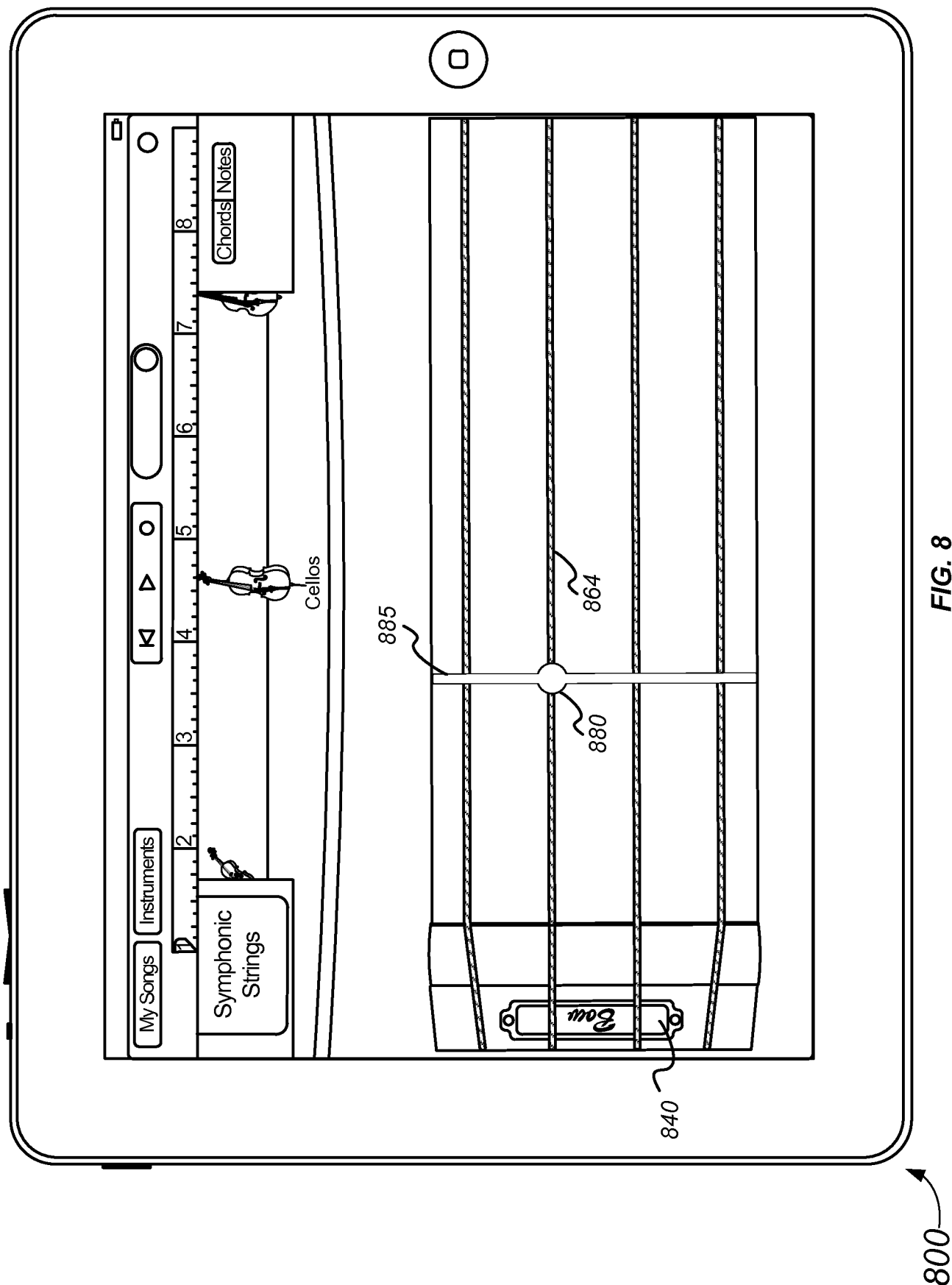
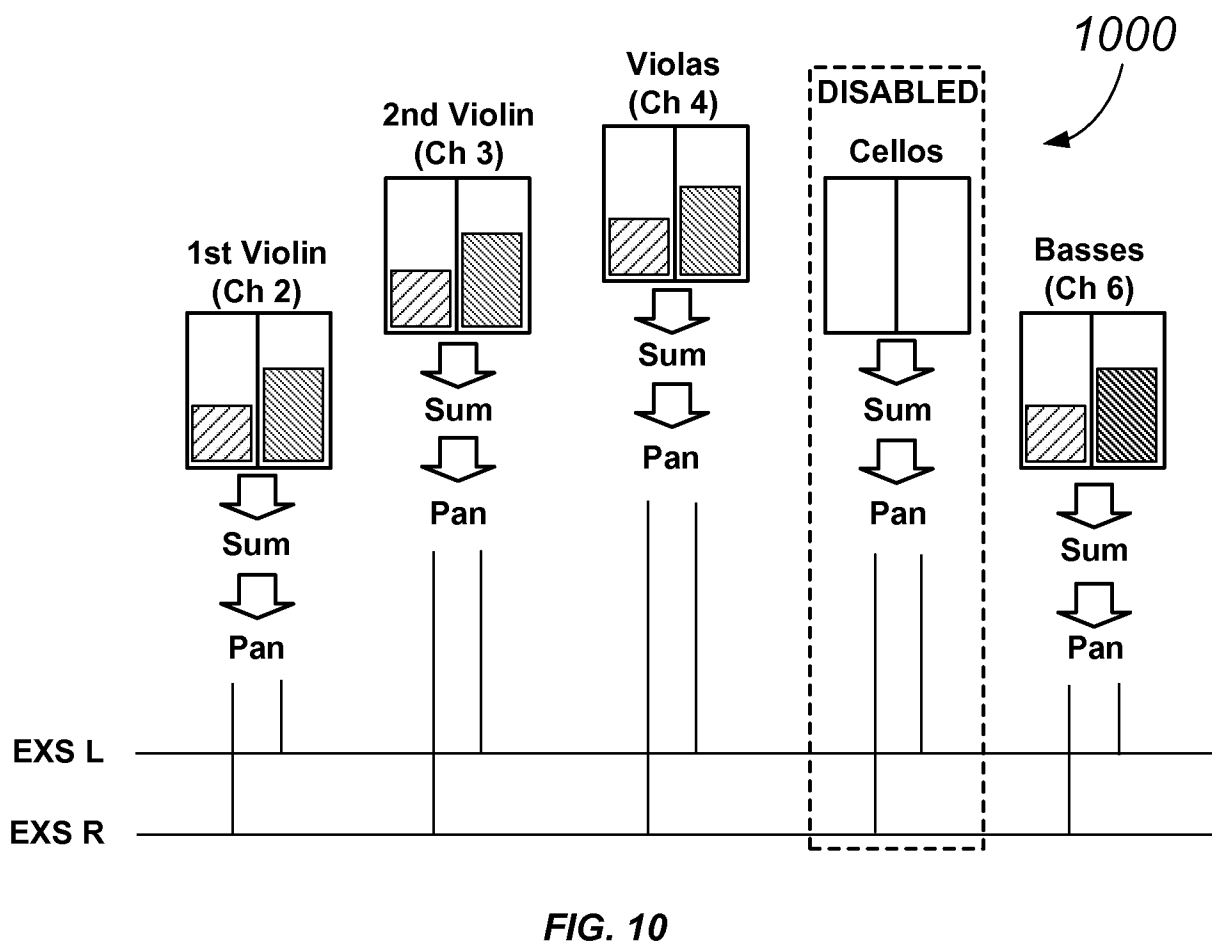
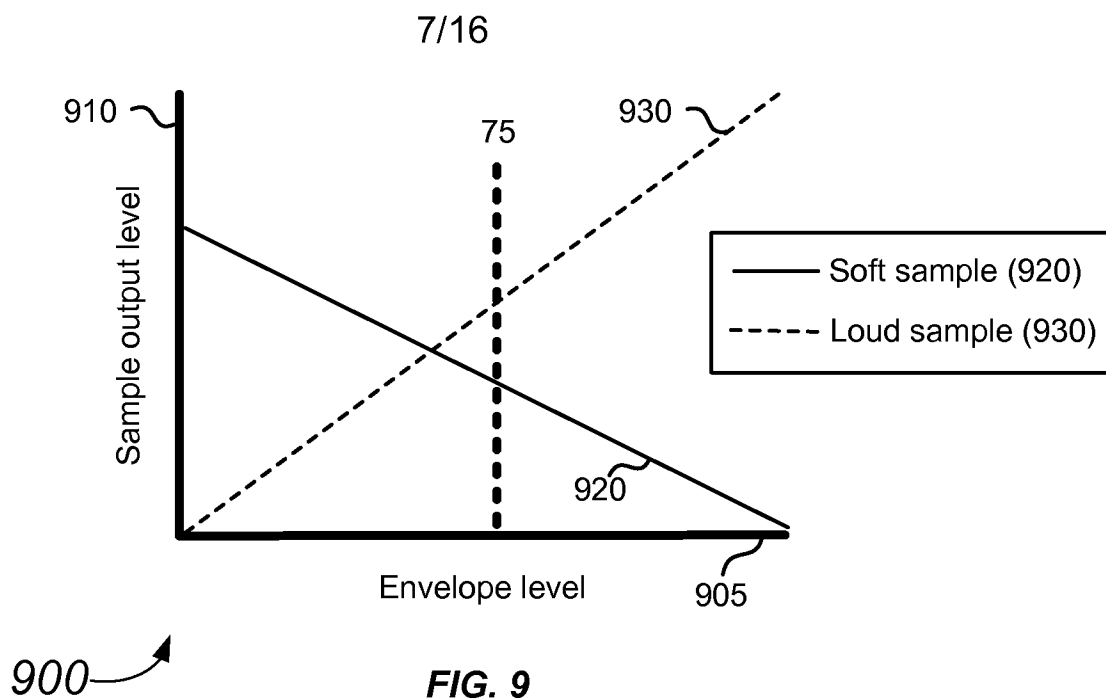
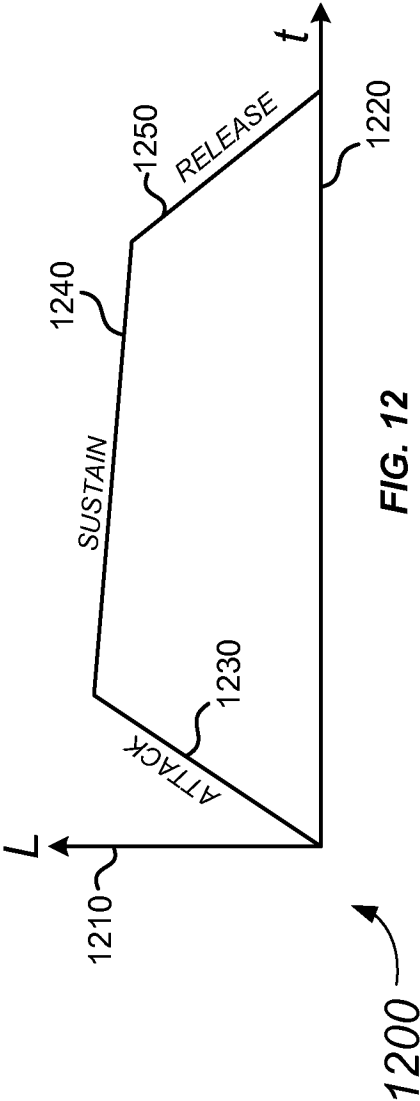
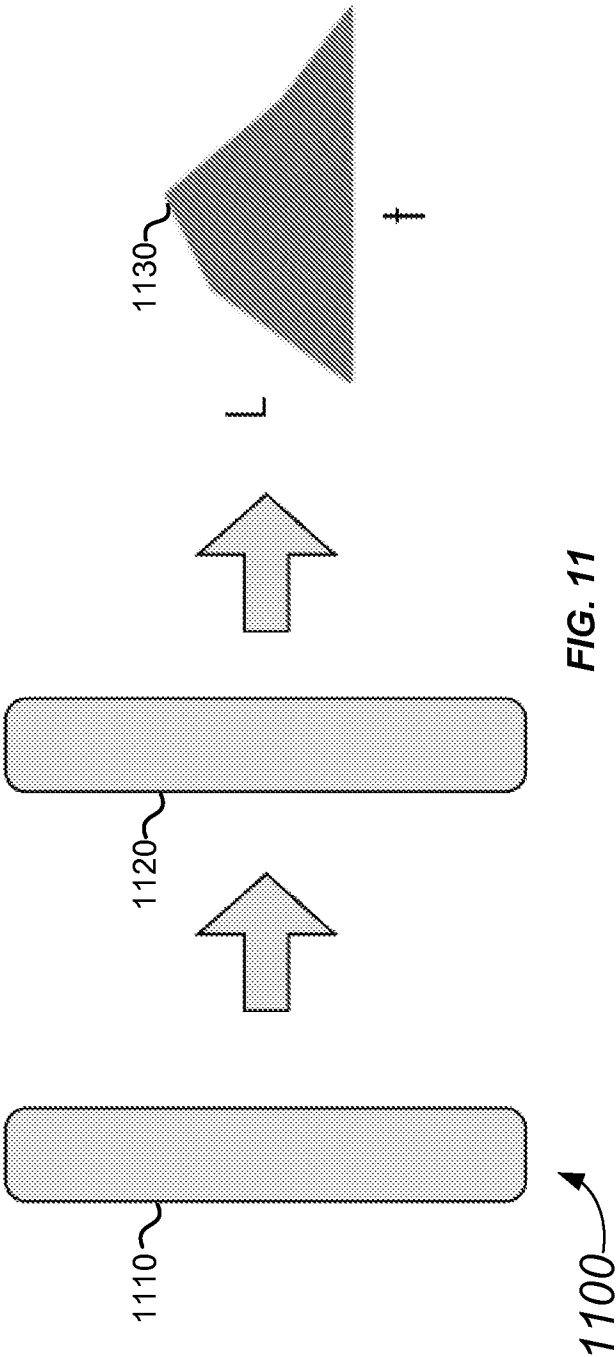


FIG. 7







Note Lifecycle:
State machine for determining mass coefficient of 2nd flywheel based on user input

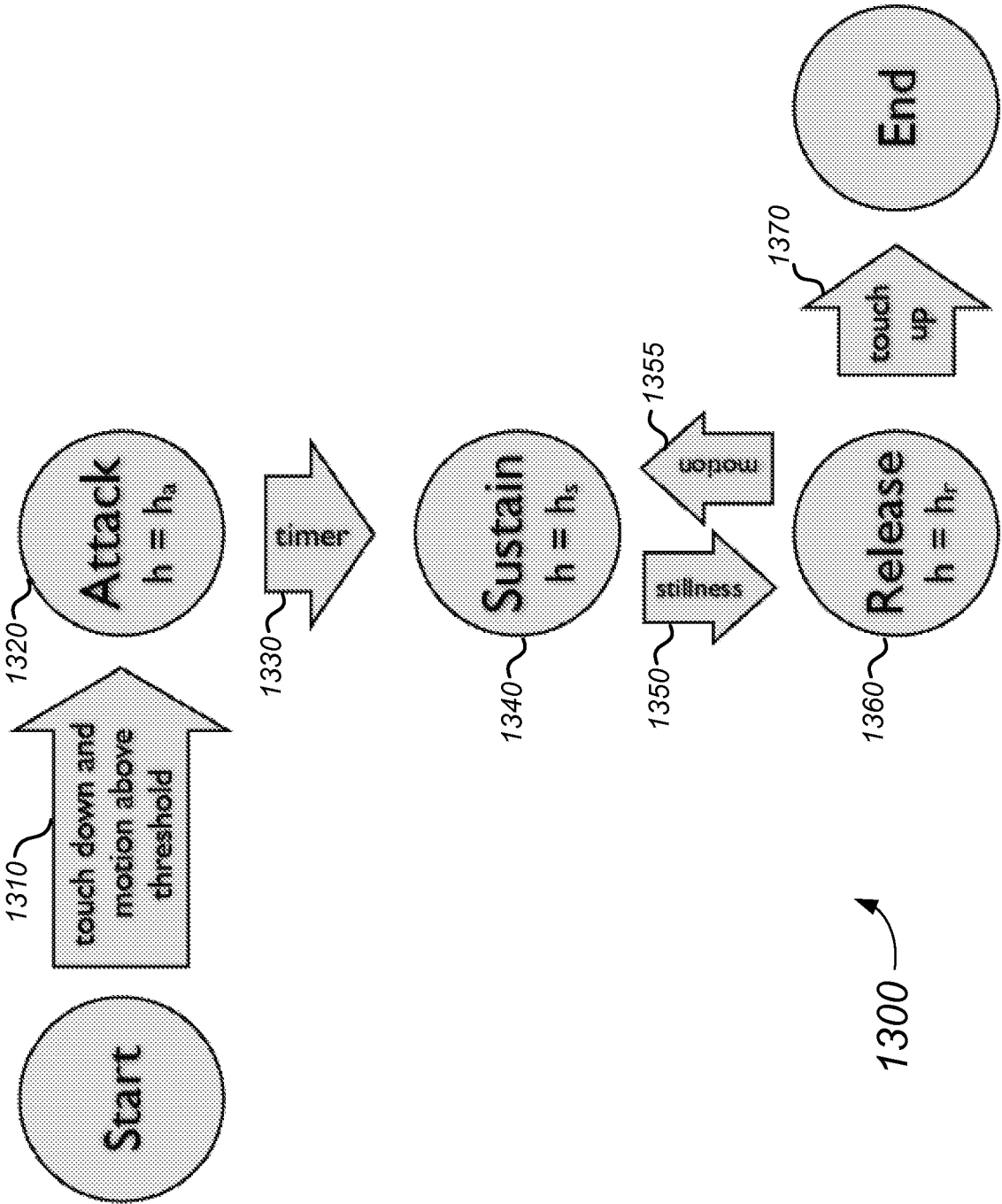
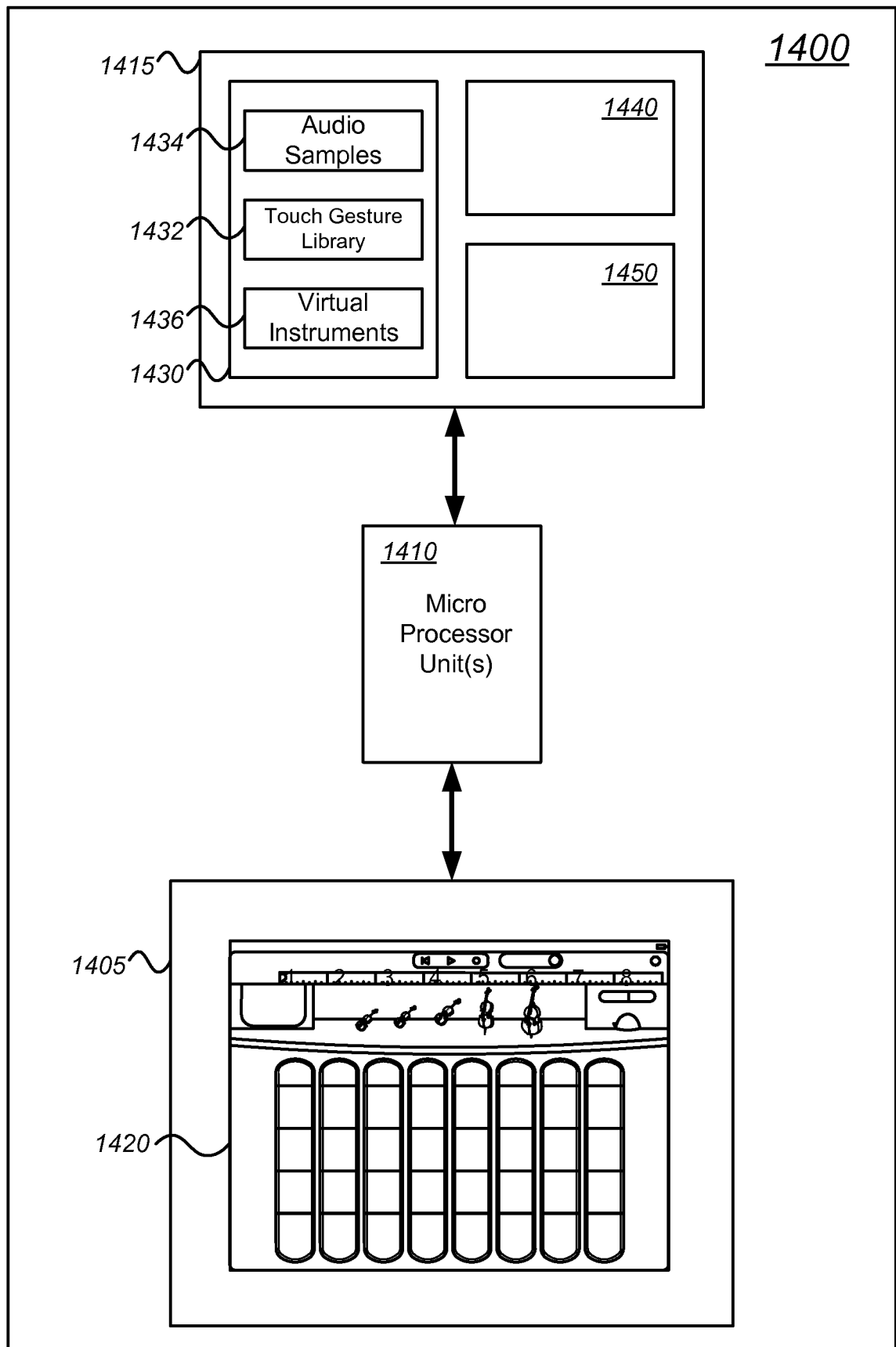


FIG. 13

10/16



11/16

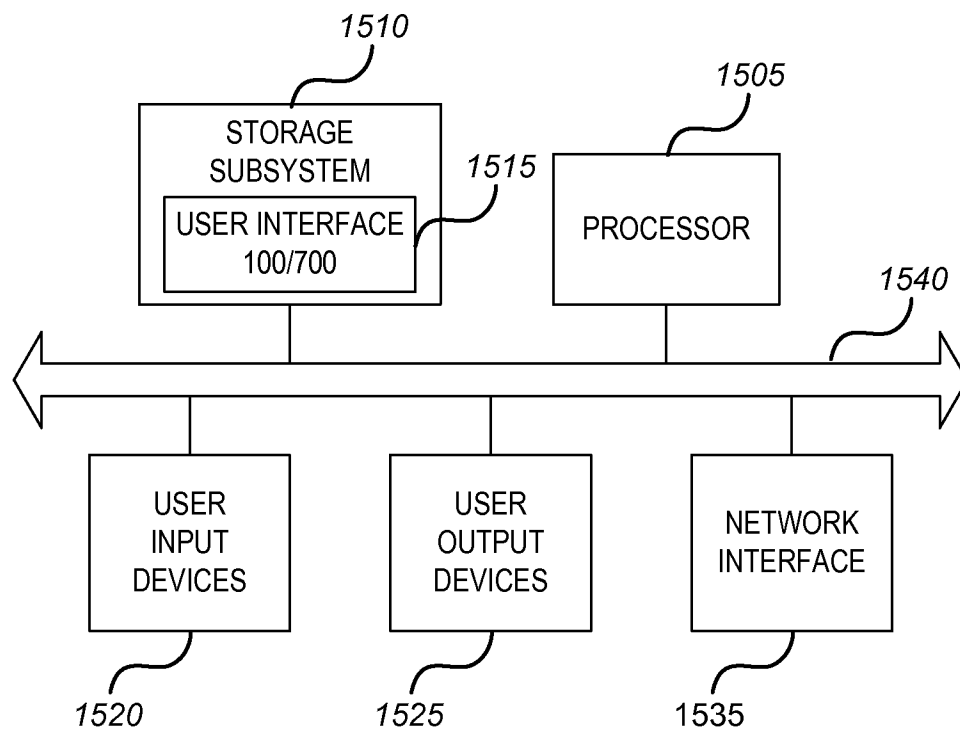
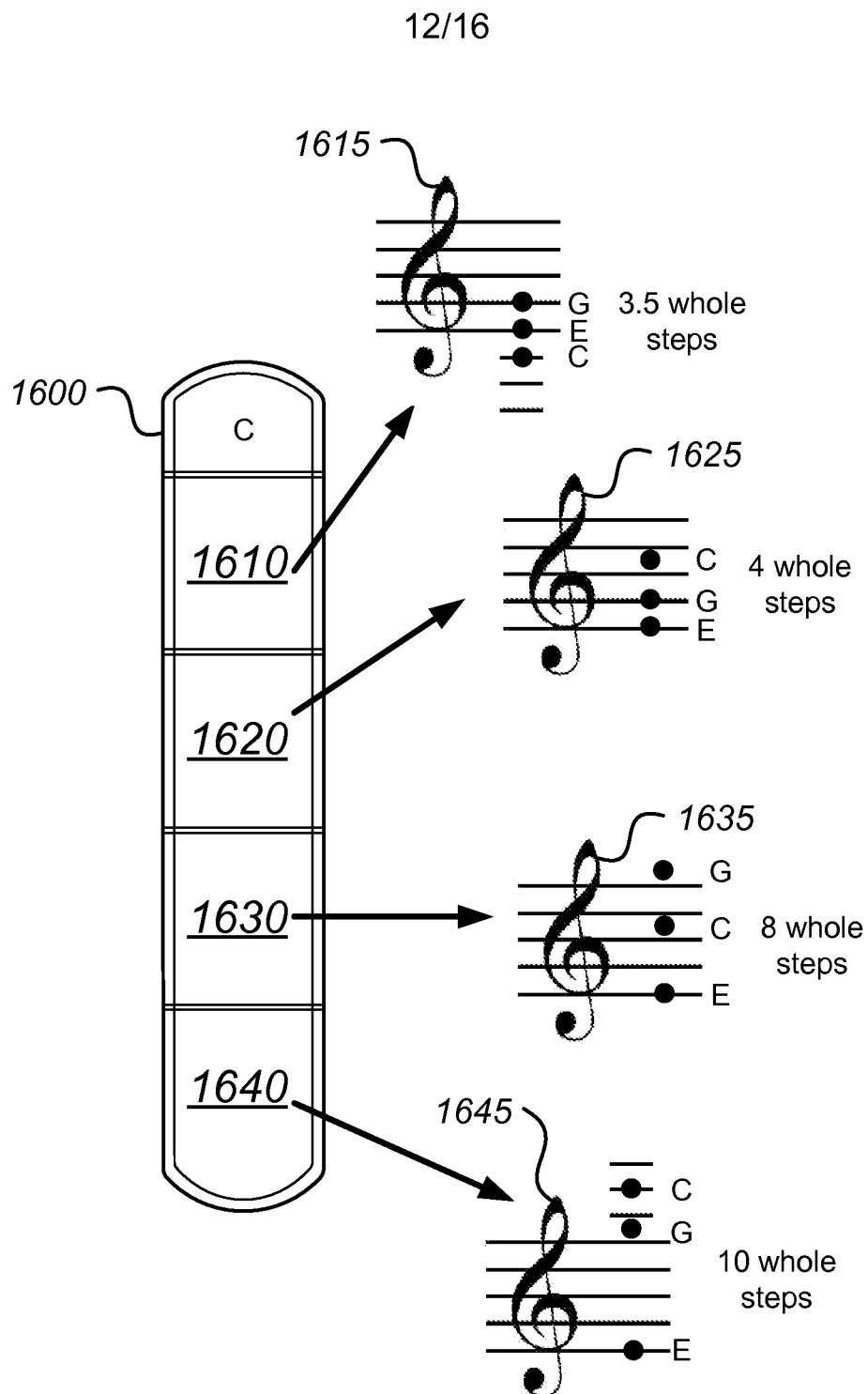


FIG. 15

1500

**FIG. 16**

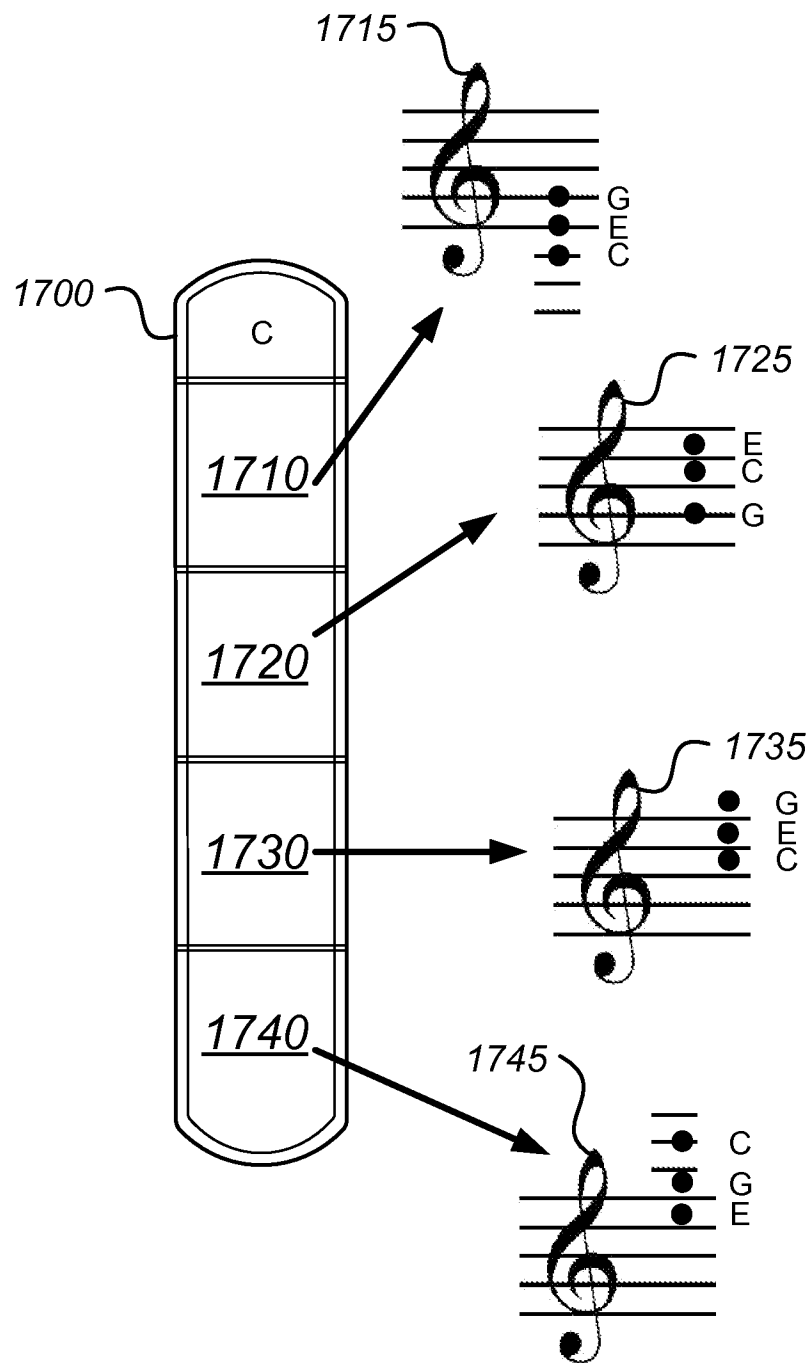


FIG. 17

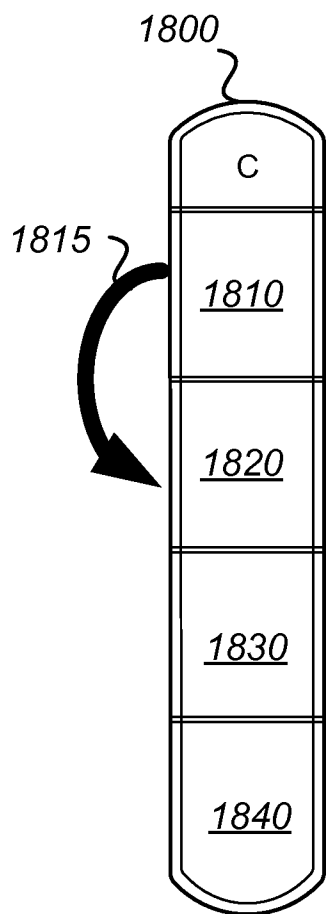
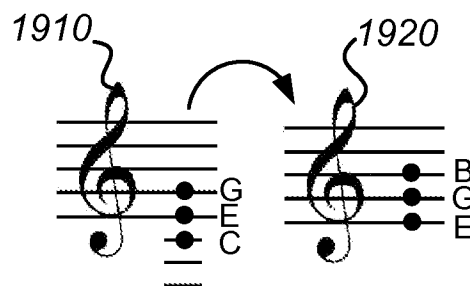
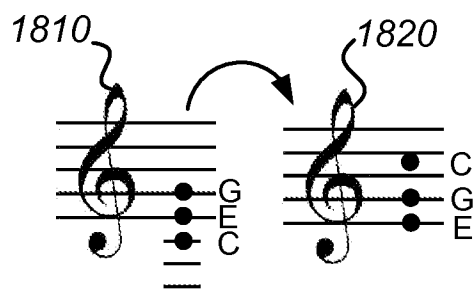


FIG. 18

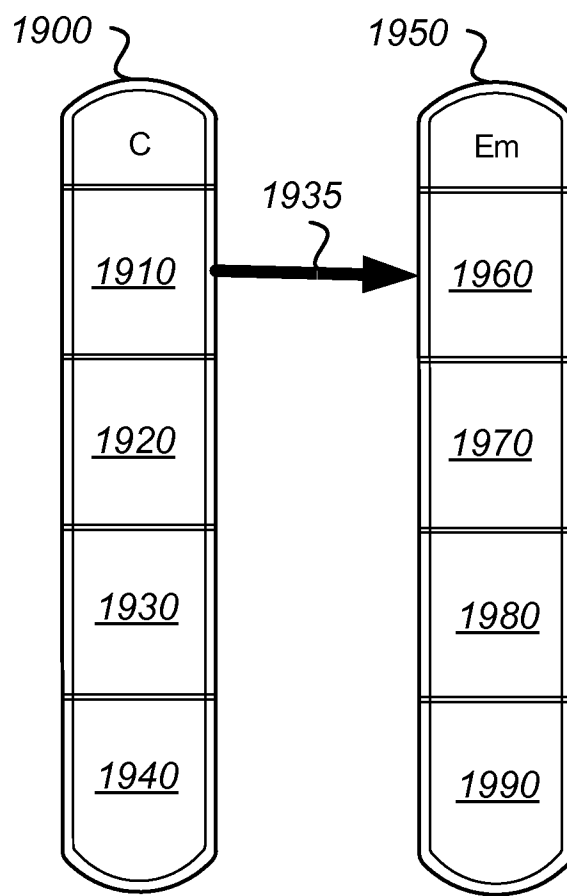
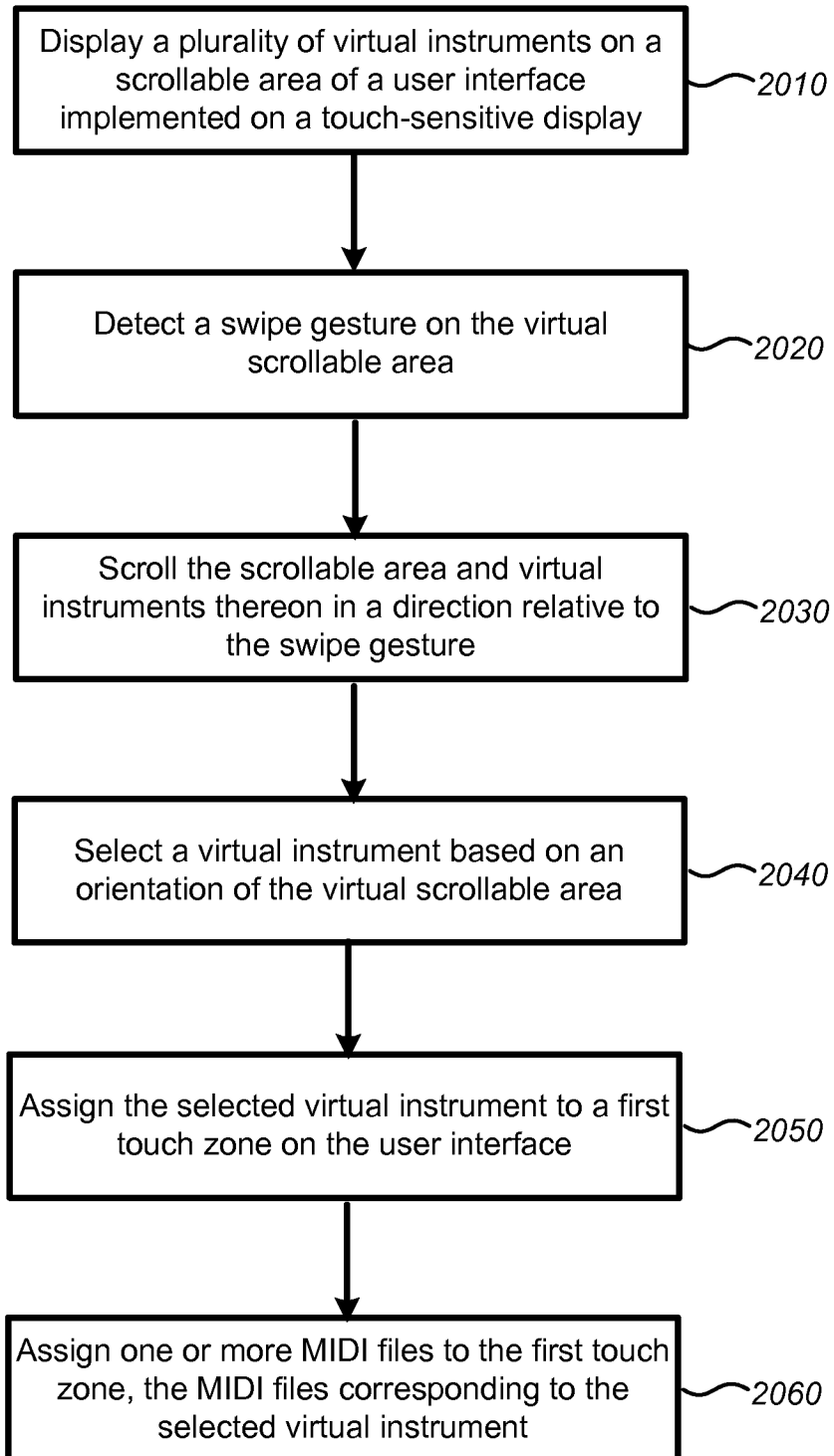


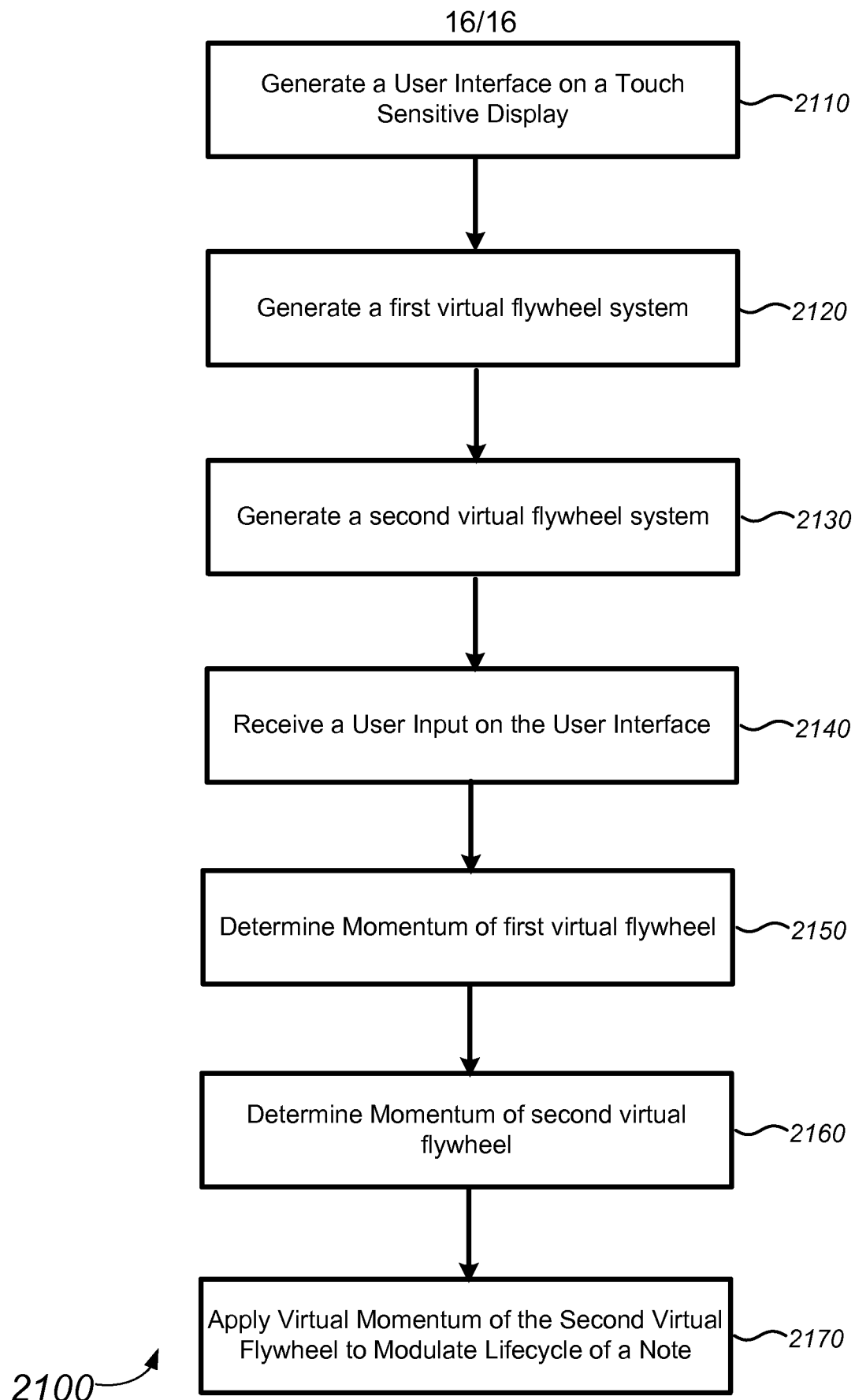
FIG. 19

15/16



2000

FIG. 20

**FIG. 21**