

(12) **United States Patent**
Osman et al.

(10) **Patent No.:** **US 12,390,728 B2**
(45) **Date of Patent:** **Aug. 19, 2025**

(54) **SMOOTH SWITCHOVER OF COMPUTER GAME CONTROL**

(71) Applicant: **Sony Interactive Entertainment Inc.,**
Tokyo (JP)

(72) Inventors: **Steven Osman**, San Mateo, CA (US);
Olga Rudi, San Mateo, CA (US);
Mahdi Azmandian, San Mateo, CA (US); **Jorge Arroyo Palacios**, San Mateo, CA (US)

(73) Assignee: **Sony Interactive Entertainment Inc.,**
Tokyo (JP)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 318 days.

(21) Appl. No.: **17/821,764**

(22) Filed: **Aug. 23, 2022**

(65) **Prior Publication Data**

US 2024/0066399 A1 Feb. 29, 2024

(51) **Int. Cl.**

A63F 13/422 (2014.01)

A63F 13/25 (2014.01)

A63F 13/45 (2014.01)

A63F 13/533 (2014.01)

(52) **U.S. Cl.**

CPC **A63F 13/422** (2014.09); **A63F 13/25** (2014.09); **A63F 13/45** (2014.09); **A63F 13/533** (2014.09)

(58) **Field of Classification Search**

USPC 463/31
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

10,814,230 B2 * 10/2020 Payzer A63F 13/25
2003/0054868 A1 3/2003 Paulsen et al.
2016/0317917 A1 11/2016 Rom et al.
2018/0165914 A1 * 6/2018 Sternberg G07F 17/3225
2019/0287343 A1 9/2019 Oberberger
2021/0268377 A1 * 9/2021 Hamaguchi A63F 13/5252
2021/0308567 A1 * 10/2021 Hussman A63F 13/24

OTHER PUBLICATIONS

“International Search Report and Written Opinion”, dated Dec. 20, 2023, from the counterpart PCT application PCT/US23/71881.

* cited by examiner

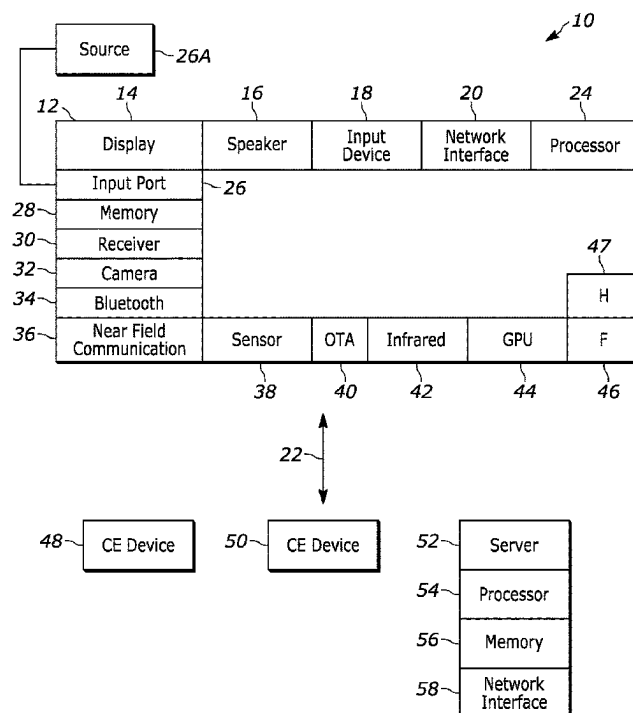
Primary Examiner — Pierre E Elisca

(74) *Attorney, Agent, or Firm* — Fish & Richardson P.C.

(57) **ABSTRACT**

Techniques are described for smooth switchover of computer game control. The current states of game input is communicated to a new player assuming control. The new player is allowed time to catch up to the game. New player control is detected, and any errors are communicated to the new player. If there are differences between the old control scheme and that of the new player, they are reconciled. The outgoing player is adjusted to the transition.

18 Claims, 12 Drawing Sheets



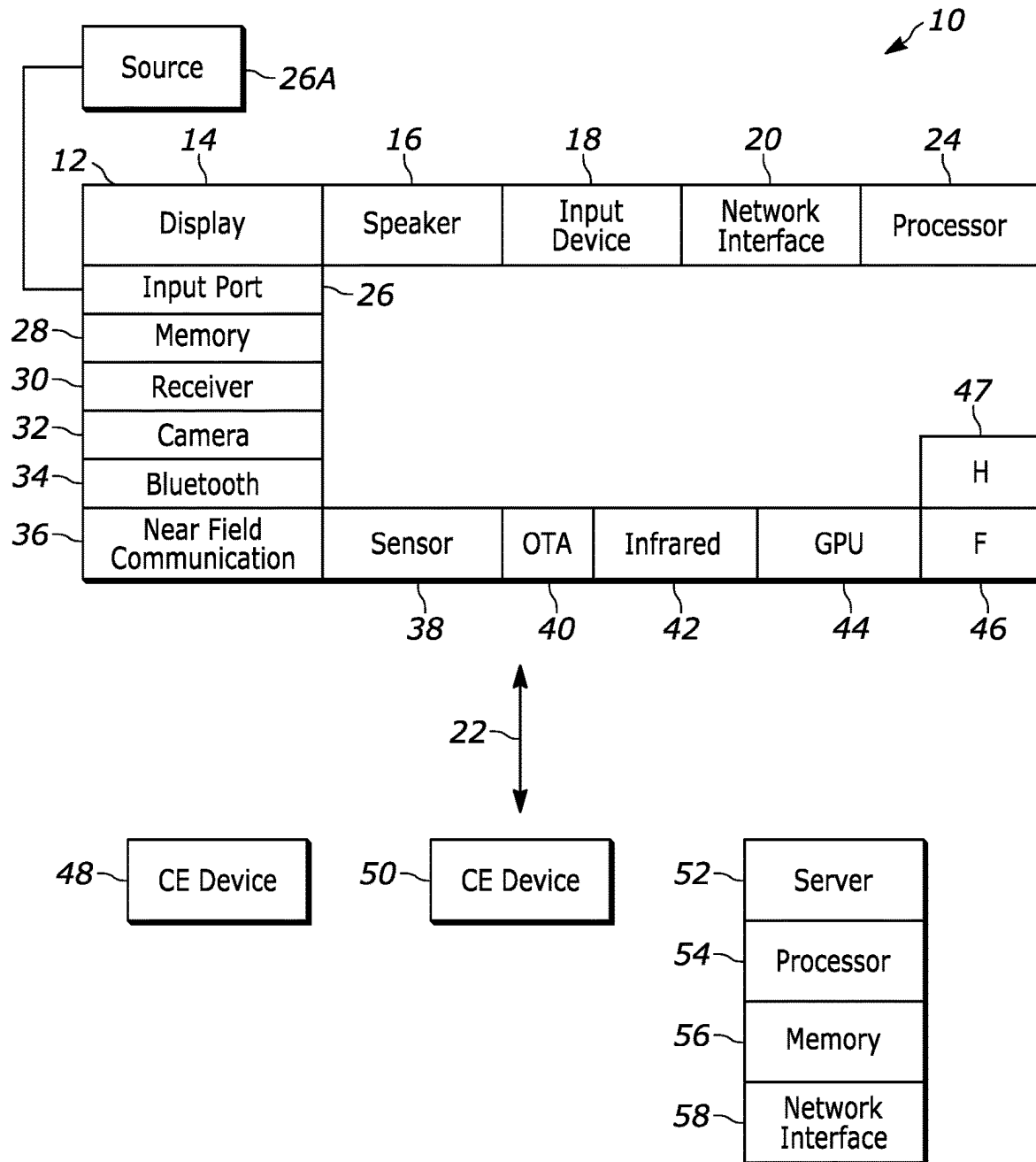


FIG. 1

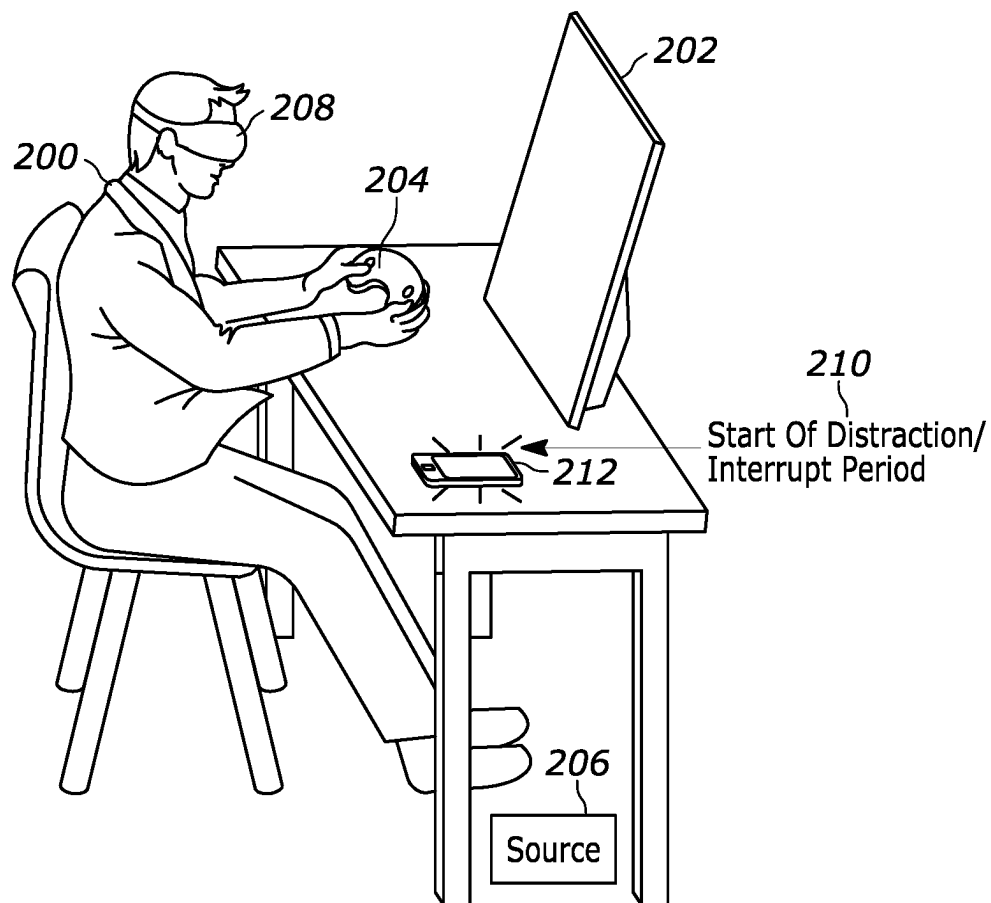


FIG. 2

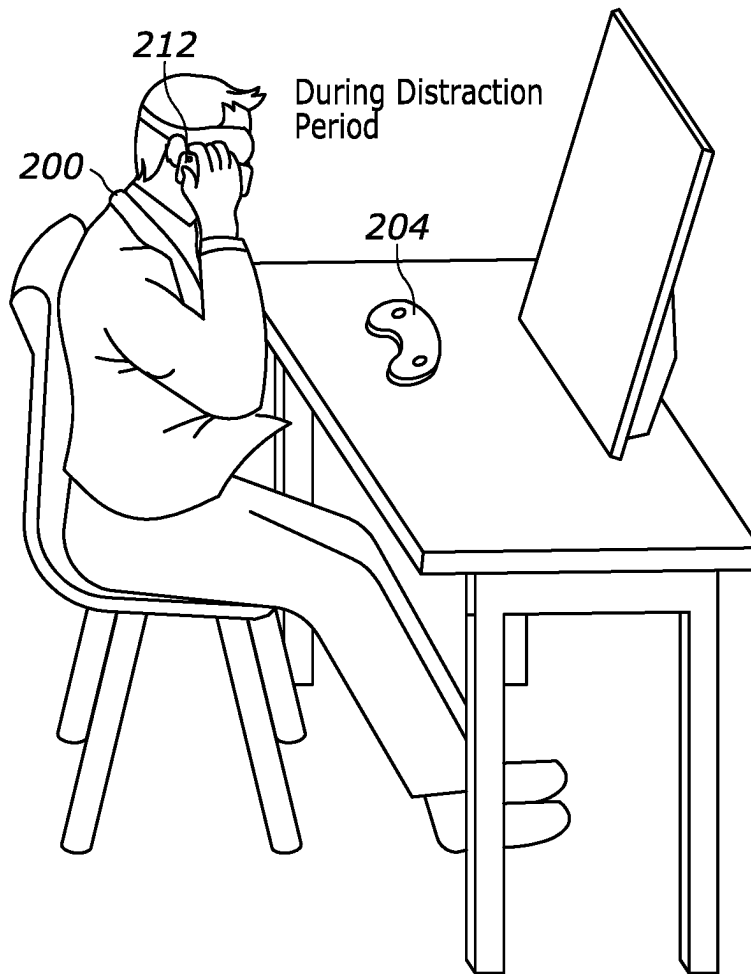


FIG. 3

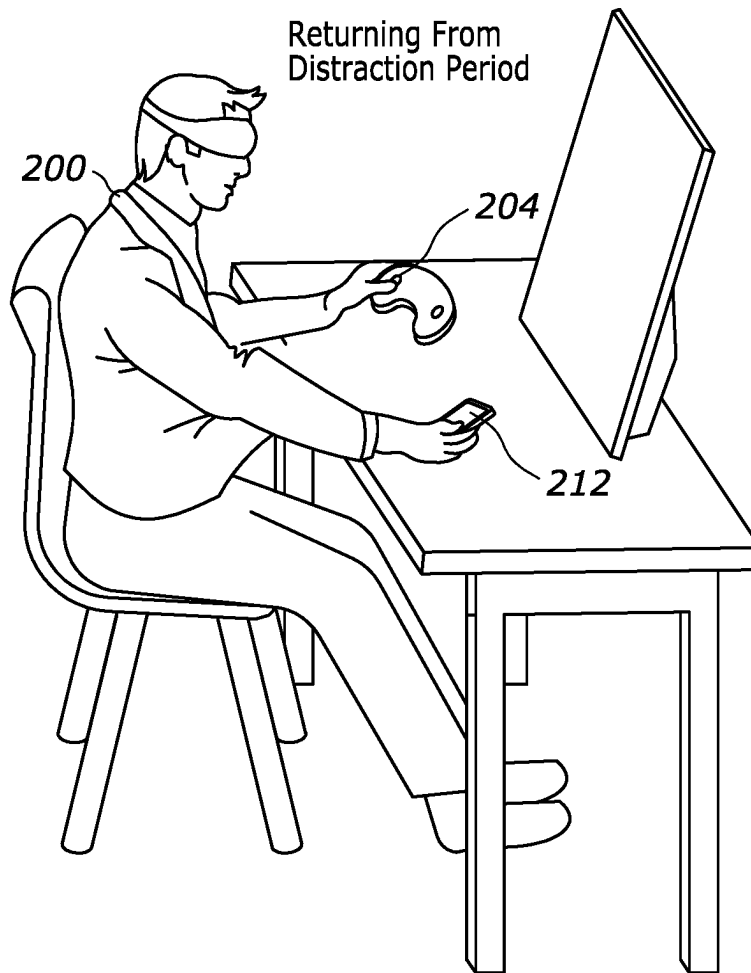


FIG. 4

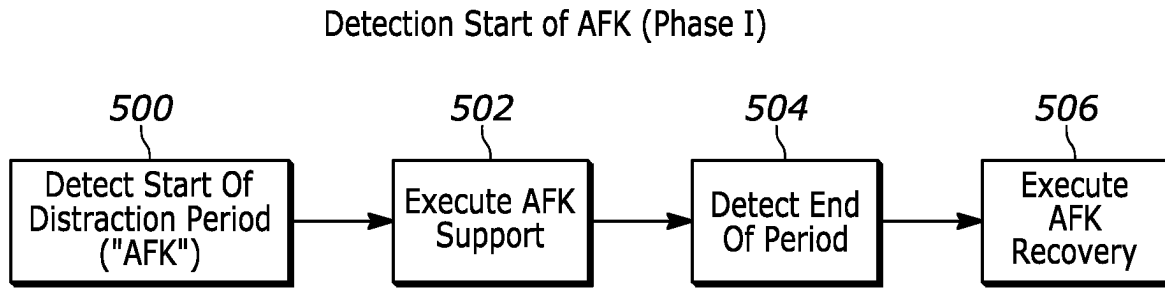


FIG. 5

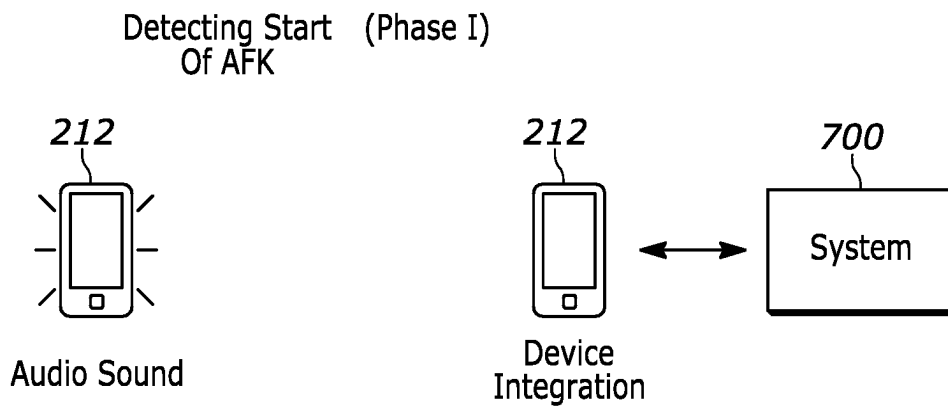
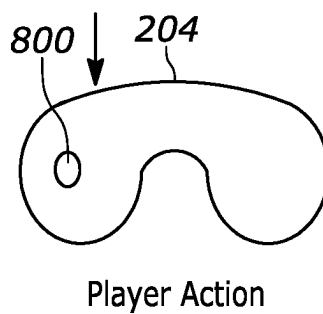


FIG. 6

FIG. 7



Player Action

FIG. 8

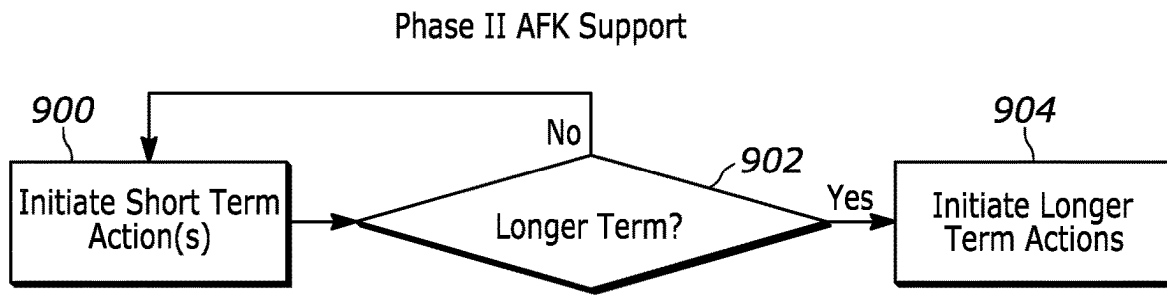


FIG. 9

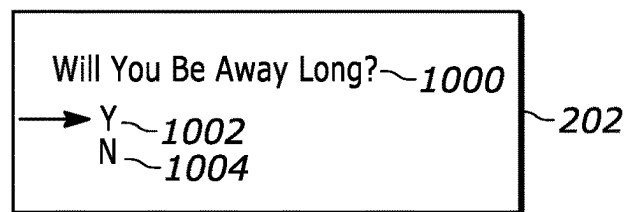


FIG. 10

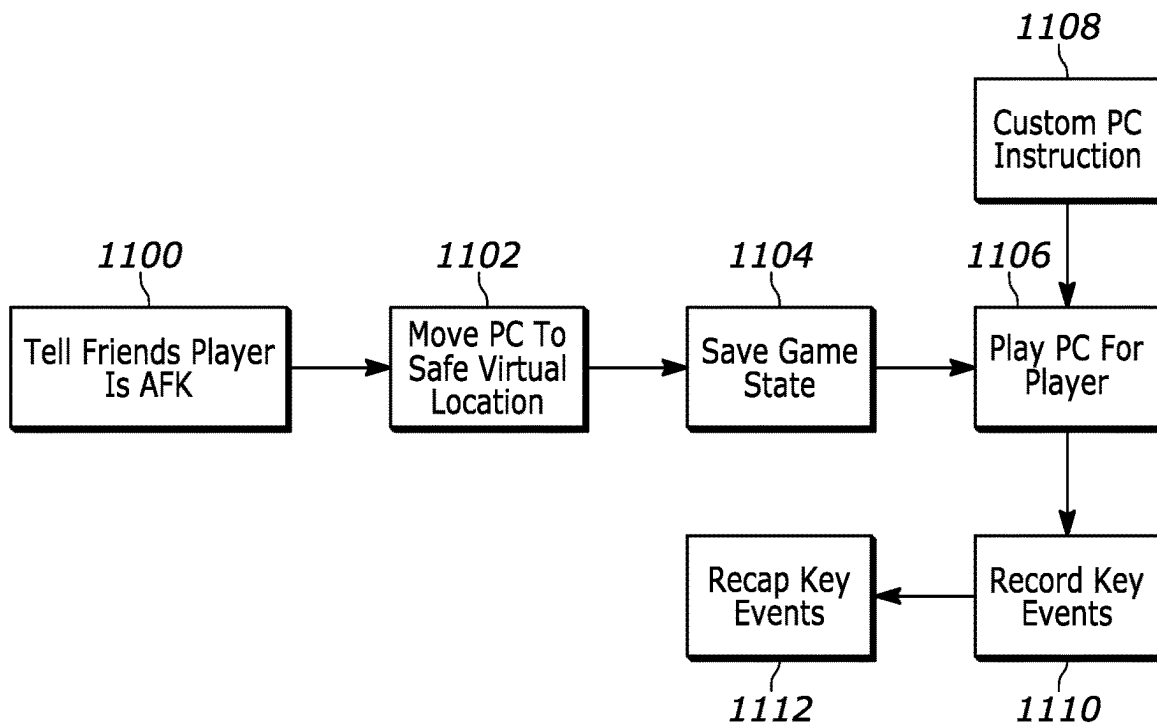


FIG. 11

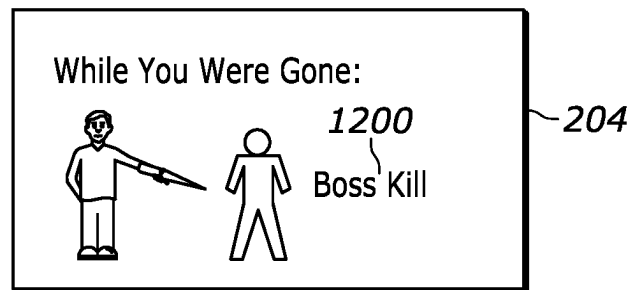


FIG. 12

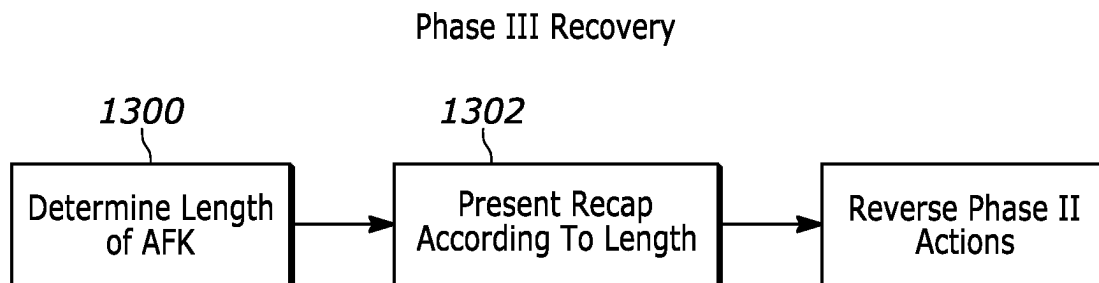


FIG. 13

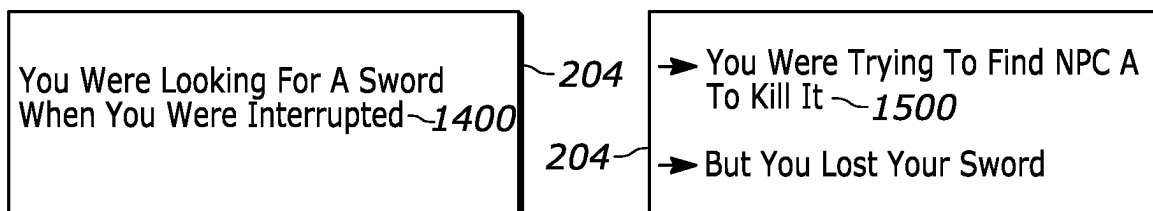


FIG. 14

FIG. 15

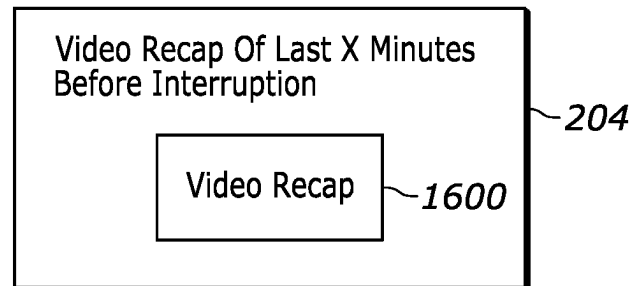


FIG. 16

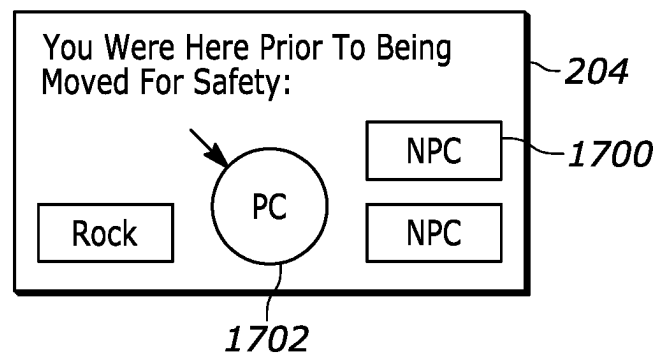


FIG. 17

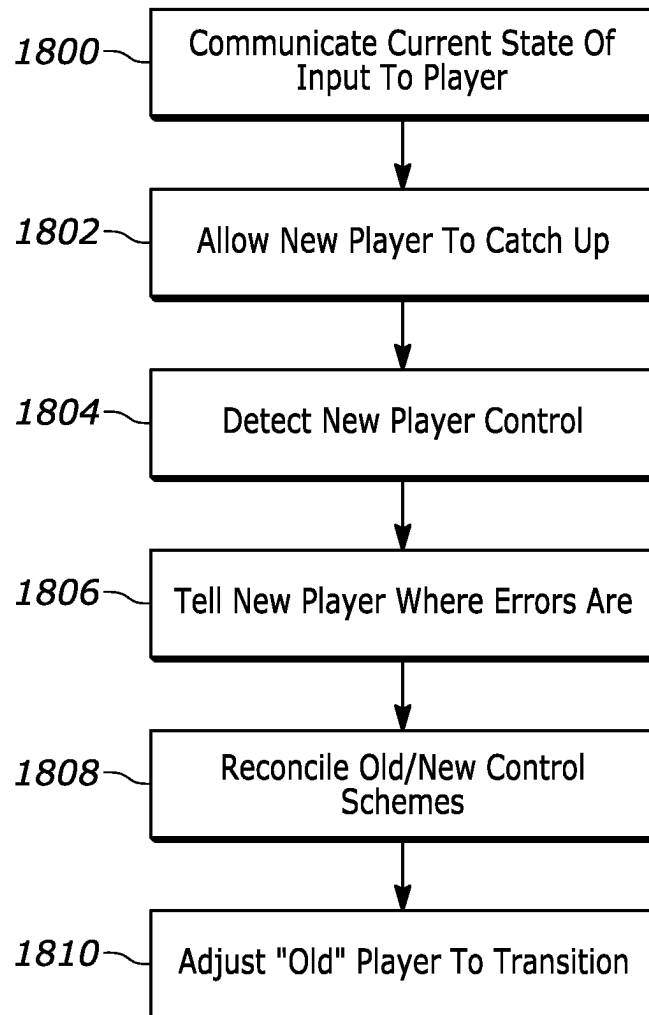


FIG. 18

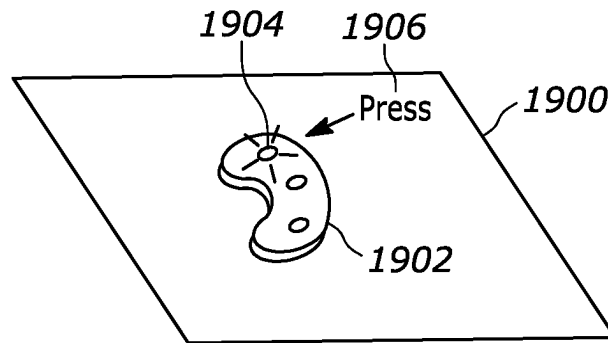


FIG. 19

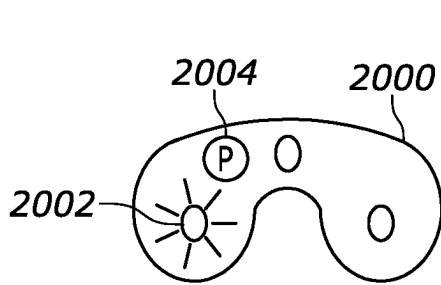


FIG. 20

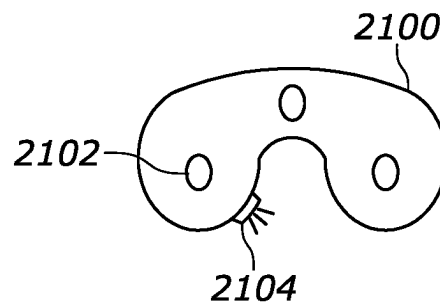


FIG. 21

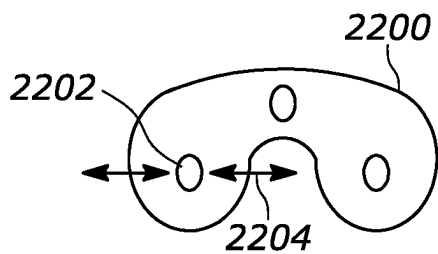


FIG. 22

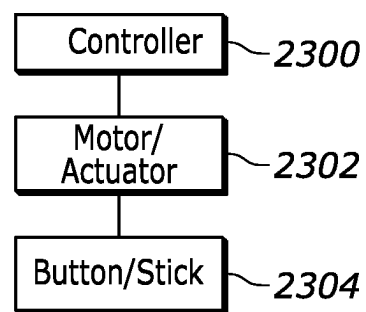


FIG. 23

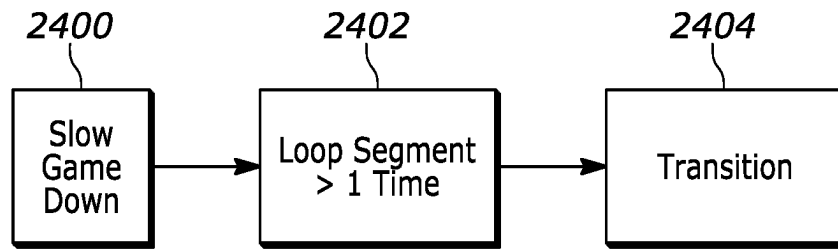


FIG. 24

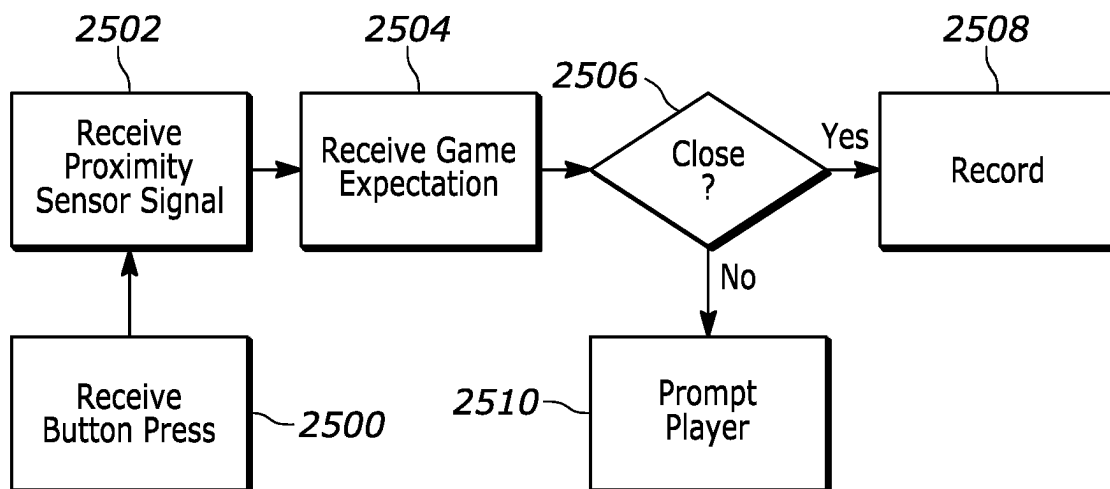


FIG. 25

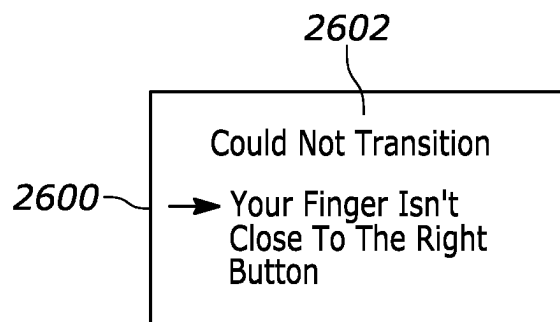


FIG. 26

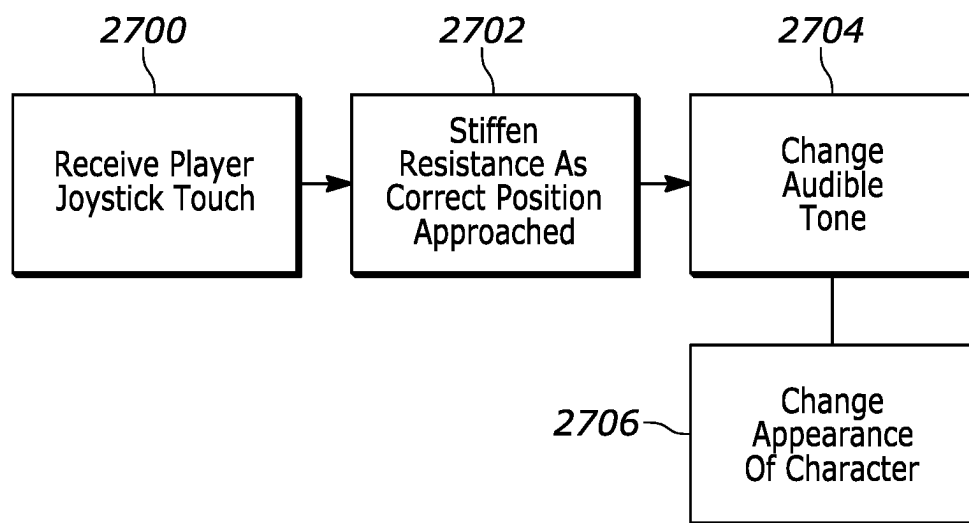


FIG. 27

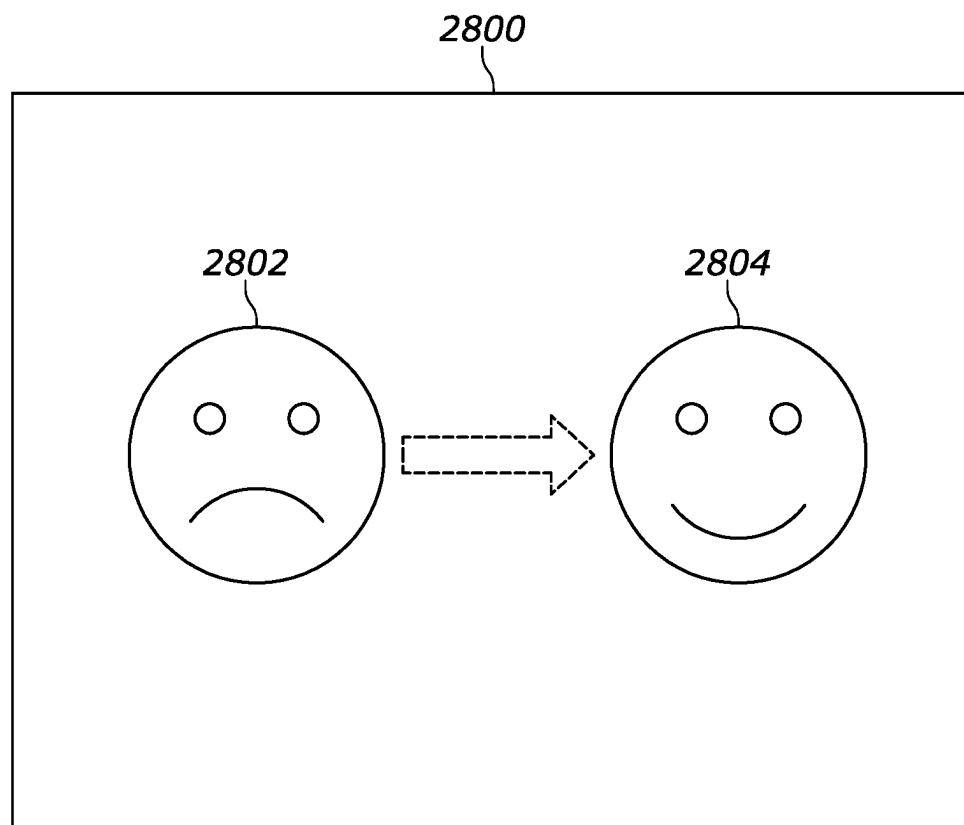


FIG. 28

1

SMOOTH SWITCHOVER OF COMPUTER GAME CONTROL

FIELD

The present application relates generally to smooth switchover of computer game control.

BACKGROUND

Some computer simulations such as computer games require that control of the game, or a player, or a weapon, or other game element be passed from one player to another. For example, in a tag team game, one player may pass control of a character or vehicle to another. As yet another example, to provide for better accessibility, two players may simultaneously be controlling a single character, for instance one controls the motion of the character and the other aims and shoots a gun. The balance of these controls may be adjusted by transitioning some of the capabilities from one player to the other.

Yet again, a video game coach may teach a player how to play a game, and set up a specific scenario that the coach wishes that player to pick up. Yet again, a player may transition control to himself but at a later time, for instance, if pausing a game in the middle of the action and resuming it later. A player may be taking over for an artificial intelligence (AI)-implemented agent, either because the AI agent took control for the player while the player was away from the game or because the play is "possessing" a new computer-controlled character.

Such "passing the baton," as it were, particularly in video games which have inputs with many degrees of freedom can be challenging.

SUMMARY

Accordingly, an assembly includes at least one processor configured with instructions to communicate to a new player assuming control of a computer game current state of game input. The instructions are executable to allow the new player time to catch up to the game, detect control of the game by the new player, and communicate errors in assuming control to the new player. The instructions are further executable to execute the computer game according to control signals received from a device operated by the new player.

In some embodiments the instructions may be executable to reconcile an old control scheme with a control scheme of the device of the new player, and/or to provide signals via at least one computer game element to adjust an old player to transition of control to the new player.

In example implementations, the instructions can be executable to communicate to the new player current state of game input at least in part by presenting on a computer display an image of a game controller having plural control elements, and highlighting at least a first one of the control elements to indicate the first control element is to be pressed during switchover.

In example implementations, the instructions can be executable to communicate to the new player current state of game input at least in part by presenting on a computer display an alpha-numeric prompt to operate at least a first control element of a computer game controller to indicate the first control element is to be pressed during switchover.

In example implementations, the instructions can be executable to communicate to the new player current state of

2

game input at least in part by illuminating at least a first control element on at least one computer game controller to indicate the first control element is to be operated during switchover.

In example implementations, the instructions can be executable to communicate to the new player current state of game input at least in part by activating at least a first speaker adjacent a first control element on at least one computer game controller to indicate the first control element is to be operated during switchover.

In example implementations, the instructions can be executable to communicate to the new player current state of game input at least in part by activating at least a first haptic element adjacent a first control element on at least one computer game controller to indicate the first control element is to be operated during switchover.

In example implementations, the instructions can be executable to communicate to the new player current state of game input at least in part by moving at least a first control element on at least one computer game controller to indicate the first control element is to be operated during switchover.

In example implementations, the instructions can be executable to allow the new player time to catch up to the game at least in part by slowing down play back speed of the game. In example implementations, the instructions can be executable to allow the new player time to catch up to the game at least in part by looping a first game segment repeatedly until the new player achieves a good input match as indicated by correctly operating control elements on the device and/or correctly holding the device in a correct pose. In example implementations, the instructions can be executable to allow the new player time to catch up to the game at least in part by transitioning one function of the device at a time until a full transition is accomplished.

In non-limiting embodiments the instructions can be executable to detect control of the game by the new player at least in part by receiving a signal from at least one proximity sensor on the device, and based at least in part on the signal from the proximity sensor, identifying whether the new player has correctly assumed control.

In non-limiting embodiments the instructions can be executable to communicate errors in assuming control to the new player at least in part by generating at least one haptic signal on the device and/or on a display associated with the game indicating an error in assuming control. In non-limiting embodiments the instructions can be executable to communicate errors in assuming control to the new player at least in part by generating at least one audible signal on the device and/or on a display associated with the game indicating an error in assuming control.

In non-limiting embodiments the instructions can be executable to communicate errors in assuming control to the new player at least in part by generating at least one visible signal on the device and/or on a display associated with the game indicating an error in assuming control.

In non-limiting embodiments the instructions can be executable to communicate errors in assuming control to the new player at least in part by generating at least one mechanical signal using at least one control element on the device indicating an error in assuming control.

A method and a computer storage that is not a transitory signal are also disclosed as aspects of the above.

The details of the present application, both as to its structure and operation, can be best understood in reference

to the accompanying drawings, in which like reference numerals refer to like parts, and in which:

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of an example system in accordance with present principles;

FIG. 2 illustrates a player of a computer simulation such as a computer game during the first phase of an interruption;

FIG. 3 illustrates the player in an ensuing second phase of the interruption;

FIG. 4 illustrates the player in the final (third) phase of the interruption;

FIG. 5 illustrates example overall logic in example flow chart format;

FIG. 6 illustrates a first technique for detecting an interruption;

FIG. 7 illustrates a second technique for detecting an interruption;

FIG. 8 illustrates a third technique for detecting an interruption;

FIG. 9 illustrates example logic in example flow chart format for determining whether the interruption is relative short term or long term;

FIG. 10 illustrates an example screen shot related to FIG. 9 that may be presented on any display herein;

FIG. 11 illustrates example logic in example flow chart format for second phase assistance;

FIG. 12 illustrates an example screen shot related to FIG. 11 that may be presented on any display herein;

FIG. 13 illustrates example logic in example flow chart format for third phase assistance;

FIG. 14 illustrates a first example screen shot related to FIG. 13 that may be presented on any display herein;

FIG. 15 illustrates a second example screen shot related to FIG. 13 that may be presented on any display herein;

FIG. 16 illustrates a third example screen shot related to FIG. 13 that may be presented on any display herein;

FIG. 17 illustrates a fourth example screen shot related to FIG. 13 that may be presented on any display herein;

FIG. 18 illustrates example overall logic in example flow chart format for transitioning control of a computer simulation;

FIG. 19 illustrates an example screen shot of an example UI to communicate a state of input to a new player;

FIG. 20 illustrates a first example use of a computer simulation controller to communicate a state of input to a new player;

FIG. 21 illustrates a second example use of a computer simulation controller to communicate a state of input to a new player;

FIG. 22 illustrates a third example use of a computer simulation controller to communicate a state of input to a new player;

FIG. 23 illustrates an example block diagram of a mechanical system to communicate a state of input to a new player;

FIG. 24 illustrates example logic in example flow chart format for allowing a new player to catch up;

FIG. 25 illustrates example logic in example flow chart format for detecting new player control;

FIG. 26 illustrates an example screen shot of an example UI related to FIG. 25;

FIG. 27 illustrates example logic in example flow chart format for advising a new player of any errors during change of control; and

FIG. 28 illustrates an example screen shot of an example UI related to FIG. 27.

DETAILED DESCRIPTION

This disclosure relates generally to computer ecosystems including aspects of consumer electronics (CE) device networks such as but not limited to computer game networks. A system herein may include server and client components which may be connected over a network such that data may be exchanged between the client and server components. The client components may include one or more computing devices including game consoles such as Sony PlayStation® or a game console made by Microsoft or Nintendo or other manufacturer, extended reality (XR) headsets such as virtual reality (VR) headsets, augmented reality (AR) headsets, portable televisions (e.g., smart TVs, Internet-enabled TVs), portable computers such as laptops and tablet computers, and other mobile devices including smart phones and additional examples discussed below. These client devices may operate with a variety of operating environments. For example, some of the client computers may employ, as examples, Linux operating systems, operating systems from Microsoft, or a Unix operating system, or operating systems produced by Apple, Inc., or Google, or a Berkeley Software Distribution or Berkeley Standard Distribution (BSD) OS including descendants of BSD. These operating environments may be used to execute one or more browsing programs, such as a browser made by Microsoft or Google or Mozilla or other browser program that can access websites hosted by the Internet servers discussed below. Also, an operating environment according to present principles may be used to execute one or more computer game programs.

Servers and/or gateways may be used that may include one or more processors executing instructions that configure the servers to receive and transmit data over a network such as the Internet. Or a client and server can be connected over a local intranet or a virtual private network. A server or controller may be instantiated by a game console such as a Sony PlayStation®, a personal computer, etc.

Information may be exchanged over a network between the clients and servers. To this end and for security, servers and/or clients can include firewalls, load balancers, temporary storages, and proxies, and other network infrastructure for reliability and security. One or more servers may form an apparatus that implement methods of providing a secure community such as an online social website or gamer network to network members.

A processor may be a single- or multi-chip processor that can execute logic by means of various lines such as address lines, data lines, and control lines and registers and shift registers.

Components included in one embodiment can be used in other embodiments in any appropriate combination. For example, any of the various components described herein and/or depicted in the Figures may be combined, interchanged, or excluded from other embodiments.

“A system having at least one of A, B, and C” (likewise “a system having at least one of A, B, or C” and “a system having at least one of A, B, C”) includes systems that have A alone, B alone, C alone, A and B together, A and C together, B and C together, and/or A, B, and C together.

Referring now to FIG. 1, an example system 10 is shown, which may include one or more of the example devices mentioned above and described further below in accordance with present principles. The first of the example devices included in the system 10 is a consumer electronics (CE)

device such as an audio video device (AVD) 12 such as but not limited to a theater display system which may be projector-based, or an Internet-enabled TV with a TV tuner (equivalently, set top box controlling a TV). The AVD 12 alternatively may also be a computerized Internet enabled (“smart”) telephone, a tablet computer, a notebook computer, a head-mounted device (HMD) and/or headset such as smart glasses or a VR headset, another wearable computerized device, a computerized Internet-enabled music player, computerized Internet-enabled headphones, a computerized Internet-enabled implantable device such as an implantable skin device, etc. Regardless, it is to be understood that the AVD 12 is configured to undertake present principles (e.g., communicate with other CE devices to undertake present principles, execute the logic described herein, and perform any other functions and/or operations described herein).

Accordingly, to undertake such principles the AVD 12 can be established by some, or all of the components shown. For example, the AVD 12 can include one or more touch-enabled displays 14 that may be implemented by a high definition or ultra-high definition “4K” or higher flat screen. The touch-enabled display(s) 14 may include, for example, a capacitive or resistive touch sensing layer with a grid of electrodes for touch sensing consistent with present principles.

The AVD 12 may also include one or more speakers 16 for outputting audio in accordance with present principles, and at least one additional input device 18 such as an audio receiver/microphone for entering audible commands to the AVD 12 to control the AVD 12. The example AVD 12 may also include one or more network interfaces 20 for communication over at least one network 22 such as the Internet, an WAN, an LAN, etc. under control of one or more processors 24. Thus, the interface 20 may be, without limitation, a Wi-Fi transceiver, which is an example of a wireless computer network interface, such as but not limited to a mesh network transceiver. It is to be understood that the processor 24 controls the AVD 12 to undertake present principles, including the other elements of the AVD 12 described herein such as controlling the display 14 to present images thereon and receiving input therefrom. Furthermore, note the network interface 20 may be a wired or wireless modem or router, or other appropriate interface such as a wireless telephony transceiver, or Wi-Fi transceiver as mentioned above, etc.

In addition to the foregoing, the AVD 12 may also include one or more input and/or output ports 26 such as a high-definition multimedia interface (HDMI) port or a universal serial bus (USB) port to physically connect to another CE device and/or a headphone port to connect headphones to the AVD 12 for presentation of audio from the AVD 12 to a user through the headphones. For example, the input port 26 may be connected via wire or wirelessly to a cable or satellite source 26a of audio video content. Thus, the source 26a may be a separate or integrated set top box, or a satellite receiver. Or the source 26a may be a game console or disk player containing content. The source 26a when implemented as a game console may include some or all of the components described below in relation to the CE device 48.

The AVD 12 may further include one or more computer memories/computer-readable storage media 28 such as disk-based or solid-state storage that are not transitory signals, in some cases embodied in the chassis of the AVD as stand-alone devices or as a personal video recording device (PVR) or video disk player either internal or external to the chassis of the AVD for playing back AV programs or as removable memory media or the below-described server. Also, in some embodiments, the AVD 12 can include a position or location

receiver such as but not limited to a cellphone receiver, GPS receiver and/or altimeter 30 that is configured to receive geographic position information from a satellite or cellphone base station and provide the information to the processor 24 and/or determine an altitude at which the AVD 12 is disposed in conjunction with the processor 24.

Continuing the description of the AVD 12, in some embodiments the AVD 12 may include one or more cameras 32 that may be a thermal imaging camera, a digital camera such as a webcam, an IR sensor, an event-based sensor, and/or a camera integrated into the AVD 12 and controllable by the processor 24 to gather pictures/images and/or video in accordance with present principles. Also included on the AVD 12 may be a Bluetooth® transceiver 34 and other Near Field Communication (NFC) element 36 for communication with other devices using Bluetooth and/or NFC technology, respectively. An example NFC element can be a radio frequency identification (RFID) element.

Further still, the AVD 12 may include one or more auxiliary sensors 38 that provide input to the processor 24. For example, one or more of the auxiliary sensors 38 may include one or more pressure sensors forming a layer of the touch-enabled display 14 itself and may be, without limitation, piezoelectric pressure sensors, capacitive pressure sensors, piezoresistive strain gauges, optical pressure sensors, electromagnetic pressure sensors, etc. Other sensor examples include a pressure sensor, a motion sensor such as an accelerometer, gyroscope, cyclometer, or a magnetic sensor, an infrared (IR) sensor, an optical sensor, a speed and/or cadence sensor, an event-based sensor, a gesture sensor (e.g., for sensing gesture command). The sensor 38 thus may be implemented by one or more motion sensors, such as individual accelerometers, gyroscopes, and magnetometers and/or an inertial measurement unit (IMU) that typically includes a combination of accelerometers, gyroscopes, and magnetometers to determine the location and orientation of the AVD 12 in three dimension or by an event-based sensors such as event detection sensors (EDS). An EDS consistent with the present disclosure provides an output that indicates a change in light intensity sensed by at least one pixel of a light sensing array. For example, if the light sensed by a pixel is decreasing, the output of the EDS may be -1; if it is increasing, the output of the EDS may be +1. No change in light intensity below a certain threshold may be indicated by an output binary signal of 0.

The AVD 12 may also include an over-the-air TV broadcast port 40 for receiving OTA TV broadcasts providing input to the processor 24. In addition to the foregoing, it is noted that the AVD 12 may also include an infrared (IR) transmitter and/or IR receiver and/or IR transceiver 42 such as an IR data association (IRDA) device. A battery (not shown) may be provided for powering the AVD 12, as may be a kinetic energy harvester that may turn kinetic energy into power to charge the battery and/or power the AVD 12. A graphics processing unit (GPU) 44 and field programmable gated array 46 also may be included. One or more haptics/vibration generators 47 may be provided for generating tactile signals that can be sensed by a person holding or in contact with the device. The haptics generators 47 may thus vibrate all or part of the AVD 12 using an electric motor connected to an off-center and/or off-balanced weight via the motor's rotatable shaft so that the shaft may rotate under control of the motor (which in turn may be controlled by a processor such as the processor 24) to create vibration of various frequencies and/or amplitudes as well as force simulations in various directions.

A light source such as a projector such as an infrared (IR) projector also may be included.

In addition to the AVD 12, the system 10 may include one or more other CE device types. In one example, a first CE device 48 may be a computer game console that can be used to send computer game audio and video to the AVD 12 via commands sent directly to the AVD 12 and/or through the below-described server while a second CE device 50 may include similar components as the first CE device 48. In the example shown, the second CE device 50 may be configured as a computer game controller manipulated by a player or a head-mounted display (HMD) worn by a player. The HMD may include a heads-up transparent or non-transparent display for respectively presenting AR/MR content or VR content (more generally, extended reality (XR) content). The HMD may be configured as a glasses-type display or as a bulkier VR-type display vended by computer game equipment manufacturers.

In the example shown, only two CE devices are shown, it being understood that fewer or greater devices may be used. A device herein may implement some or all of the components shown for the AVD 12. Any of the components shown in the following figures may incorporate some or all of the components shown in the case of the AVD 12.

Now in reference to the afore-mentioned at least one server 52, it includes at least one server processor 54, at least one tangible computer readable storage medium 56 such as disk-based or solid-state storage, and at least one network interface 58 that, under control of the server processor 54, allows for communication with the other illustrated devices over the network 22, and indeed may facilitate communication between servers and client devices in accordance with present principles. Note that the network interface 58 may be, e.g., a wired or wireless modem or router, Wi-Fi transceiver, or other appropriate interface such as, e.g., a wireless telephony transceiver.

Accordingly, in some embodiments the server 52 may be an Internet server or an entire server “farm” and may include and perform “cloud” functions such that the devices of the system 10 may access a “cloud” environment via the server 52 in example embodiments for, e.g., network gaming applications. Or the server 52 may be implemented by one or more game consoles or other computers in the same room as the other devices shown or nearby.

The components shown in the following figures may include some or all components shown in herein. Any user interfaces (UI) described herein may be consolidated and/or expanded, and UI elements may be mixed and matched between UIs.

Present principles may employ various machine learning models, including deep learning models. Machine learning models consistent with present principles may use various algorithms trained in ways that include supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, feature learning, self-learning, and other forms of learning. Examples of such algorithms, which can be implemented by computer circuitry, include one or more neural networks, such as a convolutional neural network (CNN), a recurrent neural network (RNN), and a type of RNN known as a long short-term memory (LSTM) network. Support vector machines (SVM) and Bayesian networks also may be considered to be examples of machine learning models. In addition to the types of networks set forth above, models herein may be implemented by classifiers.

As understood herein, performing machine learning may therefore involve accessing and then training a model on training data to enable the model to process further data to

make inferences. An artificial neural network/artificial intelligence model trained through machine learning may thus include an input layer, an output layer, and multiple hidden layers in between that are configured and weighted to make inferences about an appropriate output.

FIG. 2 illustrates a player 200 of a computer simulation such as a computer game being presented on a display 202 under control of a computer simulation controller 204 operated by the player 200. The computer simulation may be provided by a source 206 of computer simulations such as a computer game console or streaming game server consistent with commands input by the player 200 on the controller 204. In the non-limiting example shown, the player 200 also is wearing an XR headset 208 which may present the computer simulation in addition to or in lieu of the display 202.

As indicated at 210, FIG. 2 illustrates the player 200 at the start of a distraction period that has been initiated by one or more interruptions external to the computer simulation. By way of illustration, the interruption is the ringing of a mobile phone 212. Other example interruptions are discussed further below.

FIG. 3 illustrates the player 200 during the interruption. As shown, the player 200 has stopped playing the computer simulation, laying down the controller 204 and picking up the phone 212 to answer the call.

FIG. 4 illustrates the player 200 returning from the interruption to resume play of the computer simulation. The player 200 has laid down the phone 212 and grasped the controller 204 to resume playing the simulation.

FIG. 5 illustrates the overall logic of any one or more processors described herein executing a computer program, referred to herein as an away-from-keyboard (AFK) helper, consistent with the three phases of interruption illustrated in FIGS. 2-4. While the term “AFK” is used for shorthand, it is to be understood that in example embodiments the AFK helper provides assistance to the player during interruptions in playing computer simulations which may not use keyboards per se in the three interrupt phases illustrated in FIGS. 2-4.

Commencing at block 500, the start of a distraction period (“AFK”) is detected. AFK support is executed at block 502 during the interruption. The end of the distraction period is detected at block 504, and AFK recovery support is executed at block 506. These blocks are described in greater detail below.

The AFK helper can be integrated into the system software with hooks into games customized by game developers, offering a consistent, platform-wide experience that makes sense for any given game.

FIGS. 6-8 illustrate example techniques for detecting the start of a distraction period characterized by an interruption during play of a computer simulation. The AFK helper can detect “entering” AFK through a number of means. For instance, audio signals received through a microphone on the controller 204 or other device may trigger the start of a distraction period. Without limitation, such triggering audio signals may include the sound of the cell phone 212 ringing (FIG. 6), a baby crying, a doorbell ringing, sounds of an accident such as glass being shattered, etc. Such sounds may be classified by one more ML models trained on data including audio signals along with ground truth tags of the audio signals.

Depending on the type of sound classified, the duration of the distraction period may be initially estimated. The ringing of a doorbell may be interpreted to mean a short distraction period. The sound of a car starting may be interpreted to

mean a medium length period, whereas the sound of an emergency such as glass shattering may be interpreted to mean a long distraction period. Support may be rendered for a short distraction period merely by pausing game play, for example, at the detection of a phone ringing or doorbell sounding.

Or, a short distraction period always may be assumed, but if the player does not resume manipulating the controller within a short period, e.g., thirty seconds, a longer distraction period may be assumed.

FIG. 7 illustrates that another technique for detecting a distraction period caused by an interruption to simulation play may be by device integration, such as integration of a system 700 such as the computer simulation system with a cell phone 212 or other and video chat system, integration with an Internet of Things (IoT) device such as a doorbell, integration with an alarm/reminder triggered by a home assistant, etc.

FIG. 8 illustrates that another technique for detecting a distraction period caused by an interruption to simulation play may be by player action, for instance, pressing a button 800 such as a special purpose “AFK” button on the controller 204. Another example may be a triple press of a conventional button if no dedicated button exists.

FIG. 9 illustrates that support rendered to the player during the distraction period may change the longer the distraction period is. Commencing at block 900, at or near the start of the distraction period short term actions may be initiated. Such short-term actions may be to pause the simulation for the player 200 only, or for all players in a multi-player game, or transmission of a message to other players in the game that the player 200 will be absent only a short time, as but a few examples.

In the example above in which the distraction period is initially assumed to be short until a certain short term period such as thirty second has elapsed, as the thirty second mark (in one example) approaches, FIG. 10 illustrates at 1000 that the AFK helper can prompt the player, “will you be gone long?” using an audible or visual message, e.g., on the display 202. Responding “yes” 1002 or failure to respond within a threshold period can be interpreted that the distraction period will be longer, and that additional AFK support will be required. If the answer in the example above is NO 1004, the AFK helper continues with short-term AFK actions.

As mentioned above, certain triggers may automatically cause the AFK helper to assume one of the distraction length modes. For instance, glass being shattered might automatically be assumed to be a long AFK so as not to cause additional stress to the player through additional prompts.

FIG. 11 illustrates example assistance that may be rendered to the player 200 by the AFK helper. One or more of the examples shown in FIG. 11 may be rendered at or just following initial detection of the interruption (short term), and one or more of the examples shown in FIG. 11 may be rendered during a longer-term interruption.

Block 1100 indicates that the AFK helper may send messages to friends or fellow competitors of the player 200 that the player 200 is temporarily away, with an indication of whether it is a short or long time. Block 1102 indicates that the character in the computer simulation that is associated with the player 200 (player character or “PC”) may be moved within the virtual space of the simulation to a safe location in the virtual space, such as a cave or another planet. Or, the PC may remain in its virtual location but be made invisible to other game objects, or be made immortal to

blows from weapons by, e.g., establishing an impenetrable force field around the PC, such that the PC cannot be killed while the player is away.

Block 1104 indicates that a game save may be initiated particularly if a long distraction period is anticipated. Also, block 1106 indicates that for longer term distraction periods a ML-based agent may input play commands on the player’s behalf. This may be done according to customization from state 1108, for instance, with the player 200 creating a macro for what behaviors to perform under certain AFK triggers. Or, a ML model trained on player preferences may execute the play at block 1106 for the player 200. If desired, during the distraction period the controller 202 may be disabled to prevent inadvertent signals being input by a pet or child while the player is distracted.

Moving to block 1110, key events in the computer simulation may be recorded. In the context of computer games these may include crux events of game play such as boss kills. Crux events also may include what other players of the game unlocked while the player 200 was interrupted, how the PC died while the player 200 was interrupted, etc. The key events also may include time-based attention (where a competitor was focused in the simulation) and sentiment (what the emotions were of competitors during play in the distraction period). The recorded game events may be used at block 1112 to generate a recap of the simulation for presentation to the player 200 when he or she returns.

Upon returning from the distraction period caused by an interruption, the AFK helper can offer some services to help bring the player 200 back online. Based on the duration of the distraction period, the helper may know how much the player 200 needs to catch up on.

FIG. 12 illustrates that for a short-term absence, a reminder 1200 may be presented audibly or visually, for example on the display 200 shown in FIGS. 2-4, of the short-term goals of the player 200. In the example shown, the short-term goal of the player 200 was to kill a boss character in the simulation as evinced by the conduct of the PC just prior to the interruption.

FIG. 13 illustrates that more generally, when the distraction period has ended the AFK helper may determine at block 1300 a length of the distraction period, and based on that more to block 1302 to present a recap of the events of the computer simulation that transpired prior to and/or during the distraction period based on the length of the period. The simulation may be restarted at the point the recap video ends, perhaps with a merging effect such as slowing down the speed of a race car in a race simulation for a few seconds until the player 200 has had a chance to collect himself. Such slowing down of simulation play overall or only of certain objects in the simulation may be implemented during the distraction period as well.

Phase II actions such as one or more of those illustrated in FIG. 11 may be reversed during phase II at block 1304, for example, notifying friends of the player 200 that the player 200 has returned to play, indicating on a minimap where the PC was located before being moved to safety, etc.

For a mid-term absence a reminder of where the PC was along a quest line/quest chain and additional context may be presented. For a long-term absence (e.g., returning the next day) a recap video may be played automatically upon the player 200 taking up the controller or otherwise indicating a return to play of what the PC was doing shortly before becoming distracted by an interruption as well as what happened in the computer simulation while the player 200 was away.

11

FIGS. 14-17 illustrate the principles of FIG. 13 in further examples. In FIG. 14 a message 1400 is presented that the PC was looking for a weapon prior to being interrupted. In FIG. 15 message 1500 is presented that the PC was looking for a non-player character (NPC) to kill but that the PC had lost its weapon prior to being interrupted. In FIG. 16 a video recap 1600 is presented of the last "X" minutes of the simulation prior to being interrupted. Or, the video recap 1600 may be a recap of simulation events generated at block 1112 in FIG. 11 and presented at block 1302 in FIG. 13.

In presenting a recap of events just prior to the interruption, a brief rewind of the recap video may be effected to account for, e.g., the PC about to be killed, and an immediate continuation of the simulation springing this unfortunate event on the player before the player has had a chance to "get up to speed" in play. Additionally, further actions to seamlessly transition the player 200 back to player control may be taken. As an example, the AFK helper may insist, though onscreen instructions or audible instructions, for example, that the player 200 return key poses and controller pose to the pose these elements were in at the start of the interruption.

FIG. 17 illustrates, as alluded to above, that upon return from AFK a minimap 1700 may be presented indicating where the PC 1702 was located before being moved to safety.

Now referring to FIG. 18 et seq., whether in the context of AFK or other context, present principles provide for smoothly passing control from one player to another, or from nothing to a player. To this end, techniques are described for smooth switchover of computer game control. At block 1800 the current states of game input is communicated to a new player assuming control or the same player at a later time. The new player is allowed time to catch up to the game at block 1802. New player control is detected at block 1804 and any errors are communicated to the new player at block 1806. If there are differences between the old control scheme and that of the new player, they are reconciled at block 1808. The outgoing player is adjusted to the transition at block 1810. Each of these blocks is described in greater detail below.

FIGS. 19-23 illustrate techniques related to block 1800 in FIG. 18. In FIG. 19, a display 1900 such as any display described herein may present an image 1902 of a game controller in which a button 1904 to be pressed during switchover is highlighted and other buttons are not. Highlighting may include changing color of the button 1904, causing the button 1904 to blink, more brightly illuminating the button 1904, etc. An alpha-numeric prompt 1906 to press the button 1904 also may be presented.

FIG. 20 illustrates game controller 2000 with physical buttons, one of which buttons 2002 may be illuminated by, e.g., an internal light emitting diode (LED) to indicate it is to be pressed during switchover. One or more proximity sensors 2004 also may be provided on the controller 2000 for purposes to be shortly disclosed.

FIG. 21 illustrates game controller 2100 with physical buttons, one of which buttons 2102 may be adjacent a speaker 2104 that can be activated to beep or make some other audible sound indicating that the button 2102 is to be pressed during switchover.

FIG. 22 illustrates game controller 2200 with physical buttons, one of which buttons 2202 may be adjacent a haptic element that can be activated as indicated at 2204 to "buzz" the button 2202 or make some other haptic indication indicating that the button 2102 is to be pressed during switchover.

12

FIG. 23 indicates that mechanical controllers that automatically transition buttons, sticks and triggers to the correct position may be used. A control circuit 2300 may control an actuator 2302 to move a button or joystick on a game controller to the correct position.

FIG. 24 relates to block 1802 in FIG. 18. The new player needs a chance to catch up. This could be achieved, for instance, by slowing down the game at block 2400, looping the same game segment over and over at block 2402 until the new player achieves a good input match as indicated by correctly pressing buttons and/or correctly holding the controller in a correct pose, or transitioning one function at a time at block 2404 until a full transition is accomplished (e.g., left-right steering, up-down steering, throttle, then camera controls). These transitions can be accomplished in a discrete or continuous fashion. For instance, the throttle value for a driving game implemented by one or more elements on a game controller can be immediately transitioned from the first player to the second player, or the throttle value can be interpolated from the first to the second for instance, by using a linear interpolation function which begins by using 100% of the input value of the first player and 0% of the second, smoothly moving to a 75%-25% split, then 50%-50%, 25%-75% and finally a 0%-100% split at which point the second (new) player will be in full control over that input.

FIGS. 25 and 26 relate to block 1804 in FIG. 18. The system needs to be aware of the new player's controls. While the controller can immediately be aware of a player's button presses when they happen at block 2500, controllers can be augmented with capacitive and proximity sensors such as the sensor 2004 shown in FIG. 20 to predict an error the new player will make. For instance, a proximity signal may be received at block 2502 and a game expectation received at block 2504. If the player is expected to press a game within $\frac{1}{4}$ of a second for example, and if it is determined at state 2506 that the capacitive sensor detects that the finger was not even resting on the button at the time, feedback can be provided to the player at block 2510 that the reason the player was unable to achieve a match is because the player wasn't even prepared for it. If the expectation is met at state 2506, the controller pose and button presses may be recorded at state 2508.

Example feedback that the player was not prepared for switchover is shown in FIG. 26, illustrating a display 2600 such as any display herein that presents an alpha-numeric or other type of indication 2602 that the player was not ready to transition to control and the reason therefor.

FIGS. 27 and 28 relate to block 1806 in FIG. 18. The new player needs to be informed about where his or her errors are. This can be through haptic, audible, visual, and mechanical means. For example, an analog joystick could be very loose when a new or returning player initially touches it at block 2700 and immediately stiffen at block 2702 as the player reaches the right position. Also, or alternatively the intensity of lights on the screen or on the controller can change (e.g., dim or increase) for buttons that are being incorrectly pressed. In addition, or alternatively, an audible tone can change pitch or volume at block 2704 as the correct position of an analog control is neared.

Visual indications need not be limited to images of controllers on screen. For instance, at block 2706 the appearance of a character on screen can change as the inputs are being matched. As an example, the character can change color under these circumstances. Or, as illustrated by the display 2800 in FIG. 28, the facial expression on a character

13

2802 can transition from annoyance when there is no match to happiness (2804) when there is a good match.

With further respect to block 1808 in FIG. 18, two players (old being transitioned away from and new being transitioned to) may have different control schemes, either in the form of different layouts or different input devices. Examples of layout include button reassignment, joystick or mouse sensitivity settings, and inversion of the horizontal or vertical axes. There can be also differences in calibration between various players, for instance, when using an eye tracking system. This may be solved by matching intent instead of absolute control data. For instance, an analog trigger that controls vehicle throttle within a computer game with an input range of 0-1023 can be depressed to a level of 713 by the first player. This will be very difficult to match by the second player since it is difficult to match an analog press to that precision. Additionally, the second player may not even have that analog trigger but instead a pedal since the second (new) player may be using a steering wheel controller instead of a dual sense. In that case, the intention is to achieve a throttle of 69-70%. So rather than try to match the exact configuration of the controller (analog button depressed with a value of 713 out of 1023) the system matches the intention as defined by the action of the new player (throttle is 70%).

With further respect to block 1810 in FIG. 18, the old player may need to adjust to the transition. This can be done by providing numerous feedback to the first (old) player, including visual, audible, haptic, and even stiffening portions of the controls of the old player to make the old player adjust to the capabilities of the second (new) player.

While the particular embodiments are herein shown and described in detail, it is to be understood that the subject matter which is encompassed by the present invention is limited only by the claims.

What is claimed is:

1. An assembly, comprising:
at least one processor configured with instructions to:
communicate to a new player assuming control of a computer game current state of game input;
allow the new player time to catch up to the game;
detect control of the game by the new player;
communicate errors in assuming control to the new player; and
execute the computer game according to control signals received from a device operated by the new player, wherein the instructions are executable to communicate errors in assuming control to the new player at least in part by:
generating at least one haptic signal on the device and/or on a display associated with the game indicating an error in assuming control; and/or
generating at least one audible signal on the device and/or on a display associated with the game indicating an error in assuming control; and/or
generating at least one visible signal on the device and/or on a display associated with the game indicating an error in assuming control; and/or
generating at least one mechanical signal using at least one control element on the device indicating an error in assuming control.
2. The assembly of claim 1, wherein the instructions are executable to:
reconcile an old control scheme with a control scheme of the device of the new player.
3. The assembly of claim 1, wherein the instructions are executable to:

14

provide signals via at least one computer game element to adjust an old player to transition of control to the new player.

4. The assembly of claim 1, wherein the instructions are executable to communicate to the new player current state of game input at least in part by:

presenting on a computer display an image of a game controller having plural control elements; and
highlighting at least a first one of the control elements to indicate the first control element is to be pressed during switchover.

5. The assembly of claim 1, wherein the instructions are executable to communicate to the new player current state of game input at least in part by:

presenting on a computer display an alpha-numeric prompt to operate at least a first control element of a computer game controller to indicate the first control element is to be pressed during switchover.

6. The assembly of claim 1, wherein the instructions are executable to communicate to the new player current state of game input at least in part by:

illuminating at least a first control element on at least one computer game controller to indicate the first control element is to be operated during switchover.

7. The assembly of claim 1, wherein the instructions are executable to communicate to the new player current state of game input at least in part by:

activating at least a first speaker adjacent a first control element on at least one computer game controller to indicate the first control element is to be operated during switchover.

8. The assembly of claim 1, wherein the instructions are executable to communicate to the new player current state of game input at least in part by:

activating at least a first haptic element adjacent a first control element on at least one computer game controller to indicate the first control element is to be operated during switchover.

9. The assembly of claim 1, wherein the instructions are executable to communicate to the new player current state of game input at least in part by:

moving at least a first control element on at least one computer game controller to indicate the first control element is to be operated during switchover.

10. The assembly of claim 1, wherein the instructions are executable to allow the new player time to catch up to the game at least in part by:

slowing downplay back speed of the game.

11. The assembly of claim 1, wherein the instructions are executable to allow the new player time to catch up to the game at least in part by:

looping a first game segment repeatedly until the new player achieves a good input match as indicated by correctly operating control elements on the device and/or correctly holding the device in a correct pose.

12. The assembly of claim 1, wherein the instructions are executable to allow the new player time to catch up to the game at least in part by:

transitioning one function of the device at a time until a full transition is accomplished.

13. The assembly of claim 1, wherein the instructions are executable to detect control of the game by the new player at least in part by:

receiving a signal from at least one proximity sensor on the device; and

15

based at least in part on the signal from the proximity sensor, identifying whether the new player has correctly assumed control.

14. The assembly of claim 1, wherein the instructions are executable to communicate errors in assuming control to the new player at least in part by:

generating at least one haptic signal on the device and/or on a display associated with the game indicating an error in assuming control.

15. The assembly of claim 1, wherein the instructions are executable to communicate errors in assuming control to the new player at least in part by:

generating at least one audible signal on the device and/or on a display associated with the game indicating an error in assuming control.

16. The assembly of claim 1, wherein the instructions are executable to communicate errors in assuming control to the new player at least in part by:

generating at least one visible signal on the device and/or on a display associated with the game indicating an error in assuming control.

17. The assembly of claim 1, wherein the instructions are executable to communicate errors in assuming control to the new player at least in part by:

generating at least one mechanical signal using at least one control element on the device indicating an error in assuming control.

16

18. A device comprising:

at least one computer storage that is not a transitory signal and that comprises instructions executable by at least one processor to:

communicate to a new player assuming control of a computer game current state of game input;
allow the new player time to catch up to the game;
detect control of the game by the new player;
communicate errors in assuming control to the new player; and

execute the computer game according to control signals received from a device operated by the new player, wherein the instructions are executable also to:

communicate errors in assuming control to the new player at least in part by:

generating at least one haptic signal on the device and/or on a display associated with the game indicating an error in assuming control, and/or

generating at least one audible signal on the device and/or on a display associated with the game indicating an error in assuming control; and/or

generating at least one visible signal on the device and/or on a display associated with the game indicating an error in assuming control; and/or

generating at least one mechanical signal using at least one control element on the device indicating an error in assuming control.

* * * * *