Techniques for using a paper interface to replay time-based media information captured for a simulation and/or restart execution of a simulation program from a user-selectable time point. A static representation is generated for time-based media information captured during a simulation. The static representation may be printed on a paper medium. The paper medium on which the static representation is printed may be used to perform various operations such as replaying previously recorded time-based media information for a simulation from a user-selectable time point. The paper medium on which the static representation is printed may also be used to restart execution of a simulation program from a user-selectable time point from a previous run of the simulation program.
[Back to Previous Page] [Table of Contents] [Next Page]


[Back to Previous Page] [Table of Contents] [Next Page]


Identiteich, “Identiteich announces FY 2.5.2 release with NT object server, SmartForm, and server, SmartForm,” Business Editors, Jul. 1996.


“Spry Inc.’s Internet Front End Picked for PSI Cable Internet Service,” Information & Interactive Services Report, 15(7), (1994).


* cited by examiner
FIG. 2
Hero Rescue, Player: jamey
Level: Level 1, Duration: 01:23:43
Date: Dec 18, 2004
Events: “health packs”

FIG. 3
Hero Rescue, Player: Jamey
Dec. 16, 2003 - Jan. 15, 2004

FIG. 4
HERO RESCUE

Player: Jamey

FIG. 5A
FIG. 5B
START

1. RECEIVE A SIGNAL TO GENERATE A STATIC REPRESENTATION FOR A SIMULATION

2. RECEIVE INFORMATION SPECIFYING THE TIME DURATION(S) FOR WHICH THE STATIC REPRESENTATION IS TO BE GENERATED

3. RECEIVE INFORMATION THAT INFLUENCES THE GENERATION OF THE STATIC REPRESENTATION

4. DETERMINE TIME-BASED MEDIA INFORMATION RECORDED FOR THE SIMULATION TO BE USED FOR GENERATING THE STATIC REPRESENTATION

5. GENERATE A STATIC REPRESENTATION FOR THE TIME-BASED MEDIA INFORMATION

6. OUTPUT THE STATIC REPRESENTATION

7. PRINT THE STATIC REPRESENTATION ON A PAPER MEDIUM

END

FIG. 7
START

RECEIVE INFORMATION IDENTIFYING A TAG READ FROM A STATIC REPRESENTATION PRINTED ON A PAPER MEDIUM

YES

MODE TAG?

Determine present mode

NO

TIME TAG OR ACTION TAG?

TIME

DETERMINE TIME CORRESPONDING TO THE TAG

PERFORM ACTION CORRESPONDING TO THE TAG

TIME

DETERMINE TIME CORRESPONDING TO THE TAG

PERFORM ACTION CORRESPONDING TO THE TAG

ACTION

DETERMINE THE PORTION OF THE TIME-BASED MEDIA INFORMATION TO BE REPLAYED

INITIATE REPLAY OF THE TIME-BASED MEDIA INFORMATION DETERMINED IN 816

REPLAY OR RESTART?

REPLAY

DETERMINE STATE INFORMATION FOR THE TIME DETERMINED IN 812 FOR THE TAG

USE STATE INFORMATION DETERMINED IN 820 TO RESTART SIMULATION

RESTART

END

FIG. 8
FIG. 9 PRIOR ART
PAPER INTERFACE FOR SIMULATION ENVIRONMENTS

CROSS-REFERENCES TO RELATED APPLICATIONS

The present application claims priority from and is a continuation-in-part (CIP) application of the following applications, the entire contents of which are incorporated herein by reference for all purposes:

(1) U.S. application Ser. No. 10/001,895, “PAPER-BASED INTERFACE FOR MULTIMEDIA INFORMATION”, filed Nov. 19, 2001;
(2) U.S. application Ser. No. 10/001,894, “TECHNIQUES FOR RETRIEVING MULTIMEDIA INFORMATION USING A PAPER-BASED INTERFACE”, filed Nov. 19, 2001;
(3) U.S. application Ser. No. 10/001,849, “TECHNIQUES FOR ANNOTATING MULTIMEDIA INFORMATION”, filed Nov. 19, 2001;
(4) U.S. application Ser. No. 10/001,891, “PAPER-BASED INTERFACE FOR MULTIMEDIA INFORMATION STORED BY MULTIPLE MULTIMEDIA DOCUMENTS”, filed Nov. 19, 2001;
(8) U.S. application Ser. No. 11/021,303, “TECHNIQUES FOR GENERATING A STATIC REPRESENTATION FOR TIME-BASED MEDIA INFORMATION” filed Dec. 23, 2004;
(9) U.S. patent application Ser. No. 10/081,129, entitled “Multimedia Visualization and Integration Environment,” filed on Feb. 21, 2002;
(11) U.S. patent application Ser. No. 10/465,022, entitled “Techniques For Displaying Information Stored In Multiple Multimedia Documents,” filed on Jun. 18, 2003 which in turn is a non-provisional of U.S. Provisional Application No. 60/434,314 filed Dec. 17, 2002 (the entire contents of the provisional application are incorporated herein by reference for all purposes);
(12) U.S. patent application Ser. No. 10/465,027 entitled “Interface For Printing Multimedia Information,” filed on Jan. 18, 2003 which in turn is a non-provisional of U.S. Provisional Application No. 60/434,314 filed Dec. 17, 2002 (the entire contents of the provisional application are incorporated herein by reference for all purposes);
(13) U.S. patent application Ser. No. 10/701,966 entitled “Multimedia Visualization and Integration Environment,” filed on Nov. 4, 2003;
(15) U.S. application Ser. No. 10/814,944, “Multimedia Print Driver Dialog Interfaces”, filed Mar. 30, 2004. This application in turn claims priority from the following provisional patent applications, which are incorporated herein by reference in their entirety for all purposes: (i) U.S. Provisional Patent Application No. 60/506,303 entitled “Printer Including One or More Specialized Hardware Devices” filed on Sep. 25, 2003; and (ii) U.S. Provisional Patent Application No. 60/506,206 entitled “Printer Driver, Interface and Method for Selecting and Printing Representations of Audio, Video or Processed Information” filed on Sep. 25, 2003; and
(16) U.S. application Ser. No. 10/814,844, “Printable Representation for Time-based Media”, filed Mar. 30, 2004. This application in turn claims priority from the following provisional patent applications, which are incorporated herein by reference in their entirety for all purposes: (i) U.S. Provisional Patent Application No. 60/506,303 entitled “Printer Including One or More Specialized Hardware Devices” filed on Sep. 25, 2003; (ii) U.S. Provisional Patent Application No. 60/506,411 entitled “Printer For Non-Paper and Multiple Media Types” filed on Sep. 25, 2003; and (iii) U.S. Provisional Patent Application No. 60/506,263 entitled “Systems and Methods for Printing Audio or Video Paper” filed on Sep. 25, 2003.

The present application is also related to the following co-pending patent applications, each of which is hereby incorporated by reference in its entirety for all purposes:

U.S. patent application Ser. No. 10/814,931 entitled, “Printer Having Embedded Functionality for Printing Time-Based Media,” to Hart et al., filed Mar. 30, 2004;
U.S. patent application Ser. No. 10/814,844 entitled “Printable Representations for Time-Based Media,” to Hull, et. al., filed on Mar. 30, 2004; and

BACKGROUND OF THE INVENTION

The present invention relates to simulation environments, and more particularly to techniques that provide a paper interface for replaying time-based media information generated by a simulation and/or restarting execution of a simulation program.

Simulators or simulation applications/programs are used in various fields for a variety of applications. For example in the medical field, simulators are commonly used for virtual surgeries and for training doctors. In the defense field, simulators such as flight simulators, troop movement simulators, scenario simulators, etc. are used to train personnel such as pilots, tank commanders, etc. Flight simulators are also being used in the aviation field for training pilots. Simulators are also used for other applications.
Simulation programs in the form of video games are becoming very popular in the consumer entertainment industry. The video gaming industry has enjoyed great success over the past few years. It is estimated that total video game sales topped $10 billion in 2003. The age group of video game players has also widened significantly over the last few years. In addition to male game players, the last few years have seen a dramatic increase in the number of female participants. The video gaming industry is thus expected to see continued explosive growth.

Due to the ever increasing use of simulators in different application domains, techniques that provide different ways of using simulators and accessing information generated by the simulations are desirable.

BRIEF SUMMARY OF THE INVENTION

Embodiments of the present invention provide techniques for using a paper interface to replay time-based media information generated by a simulation and/or restart execution of a simulation program.

According to an embodiment of the present invention, a static representation is generated for time-based media information captured during execution of a simulation. The static representation may be printed on a paper medium. The paper medium on which the static representation is printed may be used to perform various operations such as replaying previously recorded time-based media information for a simulation from a user-selectable time point. The paper medium on which the static representation is printed may also be used to restart execution of a simulation program from a user-selectable time point during a previous run of the simulation program.

According to an embodiment of the present invention, techniques are provided for initiating execution of a simulation program. Information is received identifying a first tag read from a paper document. A first time point is determined corresponding to the first tag. From state information stored for the simulation program, first state information is determined for the first time point. Execution of the simulation program is initiated from the first time point using the first state information.

According to another embodiment of the present invention, techniques are provided for replaying information. Information is received identifying a first tag read from a paper document. A first time point is determined corresponding to the first tag. Based upon the first time point, a portion of time-based media information recorded for a simulation program is determined. Replay of the portion of the time-based media information is initiated using an output device.

According to another embodiment of the present invention, techniques are provided for performing an operation based upon information read from a paper document. Information is received identifying a first tag read from a paper document. A first time point corresponding to the first tag is determined. A present mode is determined. If the present mode is determined to be a first mode, then from state information stored for the simulation program, first state information is determined for the first time point, and execution of the simulation program is initiated from the first time point using the first state information. If the present mode is determined to be a second mode, then based upon the first time point, a portion of time-based media information recorded for the simulation program is determined and the portion of the time-based media information is replayed.

The foregoing, together with other features, embodiments, and advantages of the present invention, will become more apparent when referring to the following specification, claims, and accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a simplified block diagram of a system that may incorporate an embodiment of the present invention;

FIG. 2 depicts a print dialog box that may be used to generate a static representation for a simulation and print the static representation on a paper medium according to an embodiment of the present invention;

FIG. 3 depicts a page from a paper document generated by printing a static representation for time-based media information recorded for a simulation on a paper medium according to an embodiment of the present invention;

FIG. 4 depicts another paper document generated by printing a static representation for time-based media information recorded for a simulation on a paper medium according to an embodiment of the present invention;

FIGS. 5A and 5B depicts another static representation that may be generated and printed on a paper medium for a simulation according to an embodiment of the present invention;

FIGS. 6A, 6B, and 6C depict various execution paths that may result from restarting a game from different time points according to an embodiment of the present invention;

FIG. 7 is a simplified high-level flowchart depicting a method of generating a static representation for a simulation according to an embodiment of the present invention;

FIG. 8 is a simplified high-level flowchart depicting a method of processing tags read from a static representation printed on a paper medium according to an embodiment of the present invention; and

FIG. 9 is a simplified block diagram of a computer system that may be used to practice an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

In the following description, for the purposes of explanation, specific details are set forth in order to provide a thorough understanding of the invention. However, it will be apparent that the invention may be practiced without these specific details.

Embodiments of the present invention provide techniques for using a paper interface to replay (or playback) time-based media information generated by a simulation and/or restart the execution of a simulation program. According to an embodiment of the present invention, a static representation is generated for time-based media information that is generated and captured during the execution of a simulation program. The static representation may be printed on a paper medium. The paper medium on which the static representation is printed may be used to perform various operations such as replaying (i.e., playing back) previously recorded time-based media information for a simulation from a user-selectable time point. The paper medium on which the static representation is printed may also be used to restart the execution of a simulation program from a user-selectable time point. Various other operations may also be performed using the paper medium.

A simulation is the result of running or executing a simulation program. Examples of simulation programs include flight simulation programs, programs that simulate virtual surgery, video games, and the like.

FIG. 1 is a simplified block diagram of a system that may incorporate an embodiment of the present invention. FIG. 1 is merely illustrative of an embodiment incorporating the
The present invention and does not limit the scope of the invention as recited in the claims. One of ordinary skill in the art would recognize other variations, modifications, and alternatives.

System 100 comprises a simulation engine 102 that is configured to execute a simulation program. A simulation refers to the execution of the simulation program by simulation engine 102. Simulation engine 102 may be embodied in various forms. Simulation engine 102 may be implemented in software, or hardware, or combinations thereof. For example, for video games, simulation engine 102 may be a game console such as those provided by companies like Sony™, Sega™, Nintendo™, Microsoft™, etc. Simulation engine 102 may be configured to receive a game cartridge or an optical disk storing the simulation program and to execute the simulation program. Simulation engine 102 may also be a data processing system such as a personal computer, and the like. For example, a data processing system may be configured to execute a simulation program (e.g., a video game, a flight simulator program, etc.). In some embodiments, simulation engine 102 may also be implemented as software modules (code or instructions) that are executed by a data processing system and facilitate running of a simulation.

The output of executing the simulation program may be displayed to a user via one or more output devices 104. Output device 104 may be a monitor (e.g., computer monitor, TV monitor), an audio output device, etc. One or more input devices 106 may be coupled with simulation engine 102. The connection between an input device 106 and simulation engine 102 may be wired or wireless. Input devices 106 enable a user to interact with a simulation executed by simulation engine 102. For example, during a game, a player may use an input device to move a character in a game. Examples of input devices 106 include keyboards, mouse controllers, joysticks, specialized controllers, motion sensors, audio capture devices, and the like.

The execution of a simulation program (i.e., a simulation) may generate different types of information. The information that is generated may include time-based media information 108 and state information 110. Time-based media information refers to information that has a temporal component associated with it. Time-based media information generally includes information that is captured or recorded over a period of time. Time-based media information is generally characterized by a start time and an end time. The time period between the start time and end time represents the time duration of the time-based media information. Time-based media information may comprise information of one or more types (or streams) such as audio stream information, video stream information, animation information, etc. or combinations thereof. Time-based media information may include one or more channels, where each channel represents information of a particular type. Multimedia information is an example of time-based media information. Multimedia information may refer to any one or a combination of text information, graphics information, animation information, audio information, video information, images (e.g., slides, whiteboard images), etc. Various formats may be used to store the multimedia information such as various MPEG formats (e.g., MPEG 1, MPEG 2, MPEG 3, MPEG 4, MPEG 7, etc.), MP3 format, Synchronized Multimedia Integration Language (SMIL) format, HTML+TIME format, WMF (Windows Media Format), RM (Real Media) format, Quicktime format, Shockwave format, various image formats (e.g., JPEG, etc.), various streaming media formats, formats being developed by the engineering community, proprietary and customized formats, and others. Time-based media information may comprise media objects of different types such as an audio media object, video media object, etc. The media objects may comprise information one or more types.

In addition to time-based media information, state information may also be generated during the execution of a simulation program. The state information may comprise information that references the time-based media information and may be used to recreate the simulation. For example, state information generated during the execution of a car racing video game may include information such as the type of car used in the simulation, the state of the car (e.g., fuel level, damage level) selected by a player for racing, the racetrack used in the simulation, the path taken by the player’s car along the racetrack, penalties earned by the player, etc. State information 110 may be temporally synchronized with time-based media information 108.

According to an embodiment of the present invention, simulation engine 102 is configured to capture and store time-based media information 108 that is generated by the execution of a simulation program. For example, simulation engine 102 may be configured to store the audio and video information generated by the execution of a video game. The video and audio information may be recorded as separate channels of the time-based media information. Different formats may be used for storing the time-based media information. The time-based media information that is generated and stored may include information of one or more types. The types of information included in time-based media information that is generated for a simulation program may depend upon the type of simulation program.

In addition to capturing time-based media information 108 for a simulation, simulation engine 102 may also be configured to capture state information 110 generated during the simulation. Simulation engine 102 may be configured to capture the state information as the simulation progresses. The state information 110 may be temporally synchronized with captured time-based media information 108.

In one embodiment, time-based media information 108 may be subdivided into units and state information 110 is captured for each unit of time-based media information 108. A unit can vary in its representation. For example, a unit may be one video frame captured every 1/30 of a second (which amounts to the standard 30 fps video recording rate). In this embodiment, for each video frame, state information 110 may comprise a unit of information storing simulation-related application state information for the video frame. The state information that is captured for each unit may vary depending upon the nature of the simulation. The format used for storing the state information may also vary. Various formats may be used such as text, XML, etc.

In some embodiments, time-based media information 108 and/or state information 110 may be recorded and stored by capture device 112 instead of (or in conjunction with) simulation engine 102. In such an embodiment, capture device 112 may be configured to tap (or receive) the time-based media information output and the state information output of simulation engine 102 and store time-based media information 108 and state information 110. The time-based media information and the state information may be captured while the simulation program is executing (i.e., as the simulation is progressing). For example, a video signal from a simulation engine 102 executing a video game may be piped to a video capture device while playing the game. Simulation engine 102 may include a video card (e.g., video cards from ATI™) that allows the output of the video card to be sent to both a monitor (e.g., output device 104) and a video capture system such as a VCR. The video input signal may be digitized during game play by recording the video to a storage medium (e.g.,
50 as a VHS tape). As a result, a MPEG2 file may be stored capturing the video information from the simulation.

55 A data processing system 114 may be configured to access time-based media information 108 for an executed simulation and generate a static representation 116 for the time-based media information. Static representation 116 is such that it can be printed on a paper medium. Static representation 116 is intended to provide a visual representation of the time-based media information 108 or a portion thereof captured for a simulation.

60 The contents of static representation 116 may vary depending on the types of information included in the time-based media information 108 for which the static representation is generated. For example, if time-based media information 108 comprises video information, then a static representation 116 for the video information may comprise one or more key-frames extracted from the video information. If time-based media information 108 comprises audio information, then a static representation 116 of the audio information may comprise a transcription (or closed-caption text equivalent) of the audio information. If time-based media information 108 comprises different types of information (e.g., audio, video, etc.), static representation 116 may comprise visual static representations generated for the various types of information. Techniques for generating various types of static representations are described in the various U.S. patent applications incorporated by reference in this application. Static representation 116 may be generated in various forms such as PDF, JPEG, and the like.

65 According to an embodiment of the present invention, static representation 116 may comprise a timeline representing the duration of the simulation. Occurrences of one or more events that occur in the time-based media information may be visually indicated along the timeline. An event (or feature) may be related to some aspect of the contents of the time-based media information or related to some other attribute of the time-based media information. For example, for time-based media information generated by execution of a video game, events may correspond to instances when a user picks up health packs in the game, kills a monster, uses a particular weapon, etc. Information identifying which events are to be included in the static representation may be provided to data processing system 114. Alternatively, data processing system 114 may be configured to automatically analyze the time-based media information captured for a simulation and detect occurrences of one or more predefined events. Description related to generating a static representation 116 comprising a timeline with events mapped to the timeline is provided in U.S. application Ser. No. 11/021,303 filed Dec. 23, 2004, the entire contents of which are incorporated herein by reference for all purposes.

70 Static representation 116 may also comprise various types of tags. The tags may include one or more tags that correspond to or encode specific time points. As described below in further detail, these time encoding tags (or time tags) may be used to initiate replay of the recorded time-based media information from a time corresponding to the time tag. The time tags may also be used to restart execution of the simulation program from a time point corresponding to the tag. Static representation 116 may also comprise tags that may be used to perform other actions. For example, one or more tags (action tags) may be provided to activate a particular mode, e.g., a replay mode or a restart mode, or to toggle between modes.

75 Various different techniques may be implemented to use the tags. In the implementations described below, barcodes are used to implement tags. The barcodes may be read by devices such as barcode readers that are configured to initiate playback or replay or initiate other actions corresponding to the tags.

80 Static representation 116 may be communicated to a printer device 118 (e.g., a printer, a copier, a multi-function device, etc.) that is configured to print the static representation on a paper medium to generate a paper document 120. Paper document 120 may comprise one or more pages with static representation 116 printed on the pages. The term paper or paper medium is intended to refer to any physical medium on which information can be printed, written, drawn, imprinted, embossed, etc.

85 Paper document 120 may be used for a variety of purposes. In one embodiment, paper document 120 may be used to replay a portion of the recorded time-based media information 108 recorded for a previously executed simulation. Replaying a portion of the recorded time-based media information causes playback of the portion of the time-based media information. For example, a user may scan a tag printed on paper document 120 using a reader device 122 to activate "playback" mode (if not already in playback mode). The user may then scan a tag representing or encoding a time point using reader device 122 to cause replay of playback of the time-based media information 108 recorded for the simulation from a time corresponding to the scanned time tag. The replay may be output on output device 104 or some other output device. In one embodiment, a media player 124 (or some other application) may be invoked responsive to selection of the time tag and media player 124 may be used to replay a portion of previously recorded time-based media information 108.

90 Most simulation engines allow a user to replay a previously executed simulation. For example, a user may play a video game (e.g., a car race) and then replay the race. The replay enables a user to see the moves that the user made during the game. The user may then use information gleaned from the replay to improve the user’s skills during a subsequent execution of the game. Replay modes in simulation applications thus allow users to learn from and build upon the user’s previous experiences in the simulation. Embodiments of the present invention enable such replays to be initiated using a paper interface.

95 Paper document 120 may also be used to restart the execution of a simulation program. As a result of the restart, execution of a simulation program is restarted from any time point defined by a previous execution of the simulation program (i.e., from any time point during a previous simulation). The re-executed simulation program then follows a new execution path from the restart time point. The starting state of the restarted simulation (i.e., the state of the restarted simulation at the starting time point) is substantially the same as the state at the corresponding time point (i.e., the time point used for the restart) during a previous simulation.

100 Several previous execution paths (or simulations) may exist for a particular simulation program. For example, several different previously executed paths may exist for a video game. The time point and the specific previous simulation to be used for the restart may be user-selectable. In some embodiments, the most recent execution path may be automatically selected for restart.

105 In one embodiment, in order to restart execution of a simulation program from a particular time point using paper document 120, the user may scan an action tag printed on paper document 120 using a reader device 122 to activate "restart" mode (if not already in restart mode). The user may then scan a tag representing or encoding a time point using reader device 122. Responsive to scanning of the time tag, information identifying the time corresponding to the scanned tag is
communicated to simulation engine 102. Simulation engine 102 then selects the most simulation (or a user-specified simulation) and determines state information 110 for the previously executed simulation up until the time point corresponding to the scanned tag. Simulation engine 102 then restarts execution of the simulation program from the time point using the determined state information.

For example, during a previous execution of a car racing video game, the player may come across a fork in the race track and may have selected the left fork to continue the race. The player may later restart the video game from a time point where the fork was shown, and during the newly restarted execution of the video game the player may select the right fork to continue the race.

As described above, a static representation may be generated for time-based media information recorded for a simulation. The static representation may be printed on a paper document and the paper document used to replay or restart the simulation. FIG. 2 depicts a print dialog box 200 that may be used to generate a static representation for a simulation and print the static representation on a paper medium according to an embodiment of the present invention. Print dialog box 200 depicted in FIG. 2 is merely illustrative of an embodiment of the present invention and is not intended to limit the scope of the present invention. Other variations, modifications, and alternatives are also within the scope of the present invention.

Print dialog box 200 allows a user to select a portion or portions of the recorded time-based media information captured for a simulation for which a static representation is to be generated and printed on a paper medium. For example, as depicted in FIG. 2, print dialog box 200 comprises a thumbnail timeline 202 that provides a visual representation of the recorded time-based media information for a simulation. Timeline 202 may be generated based upon content extracted from the recorded time-based media information and gives the user a context of the recorded information. Timeline 202 in FIG. 2 depicts times along the timeline when certain events occur in the recorded information. The events to be depicted may be selected from menu 206. As depicted in FIG. 2, the "Health Packs" events type (i.e., instances in the recorded information when health packs were picked up) has been selected.

The user may use timeline 202 to identify one or more portions of the time-based media information for which a static representation is to be generated. The user may select the entire time-based media information for generation of the static representation. The user may also select a portion of the time-based media information for generation of the static representation. In one embodiment, the portion may be specified by specifying a segment of time characterized by a start time (which may be different from the start of the time-based media information recording for the simulation) and an end time (which may be different from the end of the time-based media information recording for the simulation) and the static representation is generated for time-based media information corresponding to the specified time segment between the start time and the end time.

A user may also specify multiple possibly non-contiguous time segments, each characterized by a start time and an end time for generation of the static representation. A static representation is generated for the user-specified time segments using the time-based media information recorded for the time segments. Various user interfaces may be provided for selecting portions of the time-based media information for which a static representation is to be generated and printed.

Print dialog box 200 also enables the user to specify various layout and style and print parameters to be used for the generation and printing of the static representation. A print preview of the static representation is also provided based upon the information configured by the user. Selection of "OK" button 204 causes the static representation to be generated for portions of the time-based media information specified by the user and the static representation to be printed on a paper medium to generate a paper document comprising one or more pages on which the static representation is printed.

As previously indicated, in one embodiment, the static representation that is generated may comprise a timeline representing the duration of the simulation. Occurrences of one or more events that occur in the time-based media information may be visually indicated along the timeline. A user may provide information identifying the types of events to be included in the static representation. In FIG. 2, this information may be provided using the "Advanced Options" section of dialog box 200. Under the "Segmentation Type" selection, the user may specify the type of events to be included in the static representation. Sections of the time-based media information when the specified event occurs may be highlighted in thumbnail 202.

Further details related to print dialog box 200 and techniques for printing the static representation on a paper medium are described in the various applications incorporated herein by reference. For example, U.S. application Ser. No. 10/814,944, "Multimedia Print Driver Interfaces", filed Mar. 30, 2004 describes various print dialog interfaces.

FIG. 3 depicts a page from a paper document generated by printing a static representation for time-based media information recorded for a simulation on a paper medium according to an embodiment of the present invention. Page 300 depicted in FIG. 3 is merely illustrative of an embodiment of the present invention and is not intended to limit the scope of the present invention as recited in the claims. Other variations, modifications, and alternatives are also within the scope of the present invention.

Page 300 depicts a static representation generated and printed on a page 300 according to an embodiment of the present invention. The static representation depicted in FIG. 3 is generated using time-based media information captured during the execution of a video game entitled "Hero Rescue". As shown in FIG. 3, information 302 related to the game is printed at the top of page 300. Information 302 includes the name of the game, information identifying a player of the game, the level of the game, the duration of the time-based media information for which the static representation is generated, and the date on which the game was played.

Page 300 comprises a timeline 304 (laid vertically) representing the duration of the time-based media information for which the static representation was generated. The top of the timeline represents the start time (00:00:00) of the game recording and the bottom of the timeline represents the end (01:23:43) of the time period for which the static representation is generated.

Various events are mapped to the timeline 304. A visual representation 306 is provided for each event. In FIG. 3, the visual representation for each event is a keyframe extracted from the recorded video portion of the time-based media information when the event occurs. Other types of visual representations may be used to depict an event in alternative embodiments.

The user may specify the events to be detected in the time-based media information and included in the static representation. For example, a user may specify that events representing instances when "health packs" were picked up dur-
ing the game are to be included in the static representation for the time-based media information. In page 300, occurrences of events when “health packs” were picked up are plotted along timeline 304. Information identifying the events ("health packs") that are depicted in the static representation is included in information 302.

A tag 308 is printed for each event. Tag 308 ("time tag") for an event represents or encodes a time during the simulation when the event approximately occurred. In FIG. 3, a barcode is used to implement each time tag. A time 310 corresponding to each tag associated with an event is also printed. For each event, a line or arrow 312 is printed showing the approximate location along timeline 304 when the event occurs in the recorded time-based media information for the simulation. Depending upon the mode of operation, a time tag 308 may be scanned by a user using a barcode reader to either restart the simulation from a time corresponding to the scanned tag or to initiate replay of previously recorded time-based media information from a time corresponding to the scanned tag. In this manner, page 300 provides a paper-based interface for replaying or restarting the game.

A tag 314 ("mode tag") is also provided for activating a particular mode. If two modes are provided, namely replay mode and restart mode, mode barcode 314 may be scanned to switch between the two modes. In this scenario, scanning mode tag 314 toggles the present mode of operation. For example, if the present mode of operation is "replay" mode, then scanning mode tag 314 changes the mode to a "restart" mode, and vice versa. In alternative embodiments, more than two modes of operation may be provided. Further, multiple mode tags may be provided in alternative embodiments. A set of tags 316 is also provided for controlling the playback of the information. The events may be detected while the game is being played. For example, simulation engine 102 may be configured to analyze the time-based media information as it is being captured during game play to determine occurrences of events. The time-based media information recorded for a game may also be processed after the execution of the game to detect occurrences of events. The events data may then be used to generate the static representation for the time-based media information recorded for the game.

As described above, a user may specify the type of events to be detected in the time-based media information and to be included in the static representation. For example, a user interface such interface 200 depicted in FIG. 2 may be provided that allows a user to select the type of events to be included from menu 206. In other embodiments, one or more events may be automatically detected and selected for inclusion in the static representation. For example, events may be selected at periodic time intervals.

Further information related to static representations that include timelines and events mapped to the timeline is provided in U.S. application Ser. No. 11/021,303 filed Dec. 23, 2004.

FIG. 4 depicts another paper document 400 generated by printing a static representation for time-based media information recorded for a simulation on a paper medium according to an embodiment of the present invention. Page 400 depicted in FIG. 4 is merely illustrative of an embodiment of the present invention and is not intended to limit the scope of the present invention as recited in the claims. Other variations, modifications, and alternatives are also within the scope of the present invention.

Page 400 depicts a static representation generated for a video game “Hero Rescue”. Information related to the game is printed at the top of page 400. This information includes information identifying the name of the game, information identifying a player of the game, and information identifying the period of time over which the player played the game.

As shown in FIG. 4, a two-dimensional chart is printed where the X-axis of the chart represents the game levels and the Y-axis represents the duration (time) taken by the player for each level. The game depicted in FIG. 4 comprises seventeen levels (L1 through L17). In FIG. 4, the time axis is in increments of 15 minutes. For each level, the time 406 taken by the user to complete the level is provided. Further, for each level, for each time increment along the time axis, a representative keyframe 404 (identified by letter “K” in FIG. 4) is extracted from the video information generated by game and printed on page 400 for that level. A tag 402 is printed for each keyframe. In one embodiment tag 402 encodes a time during the level when the corresponding keyframe was extracted from the video information. Depending upon the mode of operation, time tag 402 may be scanned using a barcode reader to either restart the game from a time corresponding to the scanned tag or to initiate replay of previously recorded time-based media information from a time corresponding to the scanned time tag. In this manner, page 400 provides a paper-based interface for replaying or restarting the game.

A tag 408 ("mode tag") is also provided for switching between replay mode and restart mode. Scanning mode tag 408 toggles the present mode of operation. For example, if the present mode of operation is “replay” mode, then scanning mode tag 408 changes the mode to a “restart” mode, and vice versa. In alternative embodiments, more than two modes of operation may be provided. Further, multiple mode tags may be provided in alternative embodiments.

The static representation printed on page 400 provides information related to a player’s game playing experience. The static representation shows the number of levels that a player has completed and also the time taken by the player to complete each level. By glancing at such a graph, one can see which levels the player had difficulty completing. For example, in FIG. 4, player “Jamey” took the longest time to complete level five (L5). The keyframes depicted for each level also provide a context for the level. Tags 402 printed in the static representation also enable a user to either replay time-based media information recorded for the game or to restart the game from a particular time point.

FIGS. 5A and 5B depict an example of a static representation that may be generated and printed on a paper medium for a simulation according to an embodiment of the present invention. The embodiments depicted in FIGS. 5A and 5B are merely illustrative of embodiments of the present invention and are not intended to limit the scope of the present invention as recited in the claims. Other variations, modifications, and alternatives are also within the scope of the present invention.

As depicted in FIG. 5A, a guide book or booklet 500 is generated for a player 502 to document the player’s game playing experience. In FIG. 5A, booklet 500 is generated for the “Hero Rescue” game played by a player “Jamey”.

FIG. 5B depicts two pages 504-1 and 504-2 from booklet 500. The static representation printed on the page is a collection of levels portrayed like chapters in a book. For each level, relevant events are shown. Information identifying each event is provided. A visual representation 506 is provided for each event. In FIG. 5B, the visual representation for an event is a keyframe extracted from the video information portion of the time-based media information generated by execution of the game approximately around the time when the event occurs. The time 510 when each event approximately occurs in the time-based media information is also provided next to information identifying the event.
A tag 508 is also printed for each event. In FIG. 5B, barcodes are used to implement the tags. Other techniques may be used in alternative embodiments. Tag 508 for an event encodes the approximate time during the game play when the event occurs. Depending upon the mode of operation, tags 508 may be scanned using a barcode reader to either restart the game from a time corresponding to the scanned tag or to initiate replay of previously recorded time-based media information from a time corresponding to the scanned tag. In this manner, pages 504-1 and 504-2 provide a paper interface for replaying time-based media information recorded for a game execution or restarting the game. A set of tags 514 is also provided for controlling the playback of the information.

A tag 512 (“mode tag”) is provided for switching between replay mode and restart mode. Scanning mode tag 512 toggles the present mode of operation. For example, if the present mode of operation is “play” mode, then scanning the mode tag 512 changes the mode to a “restart” mode, and vice versa. In alternative embodiments, more than two modes of operation may be provided and multiple tags may be used for switching between modes.

A user may wish to generate representations such as those depicted in FIGS. 3, 4, 5A, and 5B for one or more games completed by the user to document the user’s game playing experience. For example, the booklet depicted in FIGS. 5A and 5B may act as a walkthrough of the game for the player. Walkthroughs are very popular in the gaming industry as they enable players to show their expertise in games. Walkthroughs also act to guide other players through a game and thus improve the satisfaction and reduce the frustration of playing a game. Walkthroughs include screenshots with descriptions of the various levels. Walkthroughs are commonly shared online through game web sites. They are also used at video gaming tournaments. Conventionally, walkthroughs are manually generated by players. Embodiments of the present invention enable paper documents for walkthroughs to be automatically generated. Further, unlike conventional walkthroughs, the walkthrough paper documents generated by embodiments of the present invention provide a paper-based interface for either replaying a previously recorded portion of the game from a player-selectable time point or for restarting the game from a player-selectable time point.

As described above, various paper-based static representations may be generated for simulations such as video games. These paper-based representations may be used to perform various operations on the simulation or the time-based media information generated by the simulations. For example, a user may use the paper-based interface to replay a portion of previously recorded time-based media information for a simulation. A user may also use the paper-based interface to restart execution of the simulation program from a particular user-selectable time point. FIGS. 6A, 6B, and 6C depict various execution paths that may result from restarting a game from different time points according to an embodiment of the present invention. The figures also help to show the differences between a replay operation and a restart operation.

FIG. 6A represents a path of execution of a simulation program by simulation engine 102. The simulation represented by FIG. 6A may be the execution of a video game. As depicted in FIG. 6A, the simulation executes from a time point T1 to a time point T2 that is temporally after T1. For example, if the simulation is a video game comprising a total of ten different levels of play, a user may start the game from the first level and then stop playing the game after the user finishes the third level. In this scenario, T1 marks the time at the start of the game and T2 mark the time when the user ends the third level. The line depicted in FIG. 6A from T1 to T2 represents the execution path for the execution of the simulation program from T1 to T2.

Time-based media information and state information may be recorded during the simulation from T1 to T2. The state information may be temporally synchronized with time-based media information. A visual static representation may then be generated for the simulation run from T1 to T2. The static representation may then be printed on a paper medium.

As described above, the static representation may comprise time tags corresponding to times T1 and T2 and time points between T1 and T2. A user may then select a particular time tag printed on the paper document and replay a portion of the previously recorded time-based media information for the simulation run from T1 to T2. For example, if a user selects a time tag corresponding to time T3, then the portion of time-based media information recorded from T3 to T2 may be played back, possibly using a media player.

According to the teachings of the present invention, the user may also restart execution of the simulation program from a time point between T1 and T2 using the paper document on which the static representation of time-based media information for the run from T1 to T2 is printed. For example, the user may scan a “mode” tag on the paper document to enter “restart” mode. The user may then scan a time tag corresponding to a time point T3, where T3 is between T1 and T2 (T3 may also be equal to T1 or T2). Upon scanning a tag corresponding to time T3, the execution of the simulation program is restarted from time T2.

As depicted in FIG. 6B, restarting a simulation at T3 causes a new execution path to be branched from the first simulation path (T1 to T2) at approximately time T3. The starting state of the simulation at the start (i.e. at T3) of the new execution path is the same as the state of the simulation at time T3 during the previously executed path from T1 to T3. The state information captured for the execution of the simulation program from T1 to T2 may be used to determine the state of the simulation at time T2. The new execution path is started based upon the state information determined for T2 during the previous execution. A user may then continue to run the simulation until time point T4. For example, if the simulation is a video game, and T1 represents the time point at the start of the first level of the game and T2 represents the time point at the end of the third level, the user may wish to restart from the third level and time T3 may correspond to the time at the start of the third level. The user may then continue playing the game until level 4 is completed, in which case T4 represents the time at the completion of level 4. Time-based media information and state information may be recorded for the simulation execution path from T1 to T4.

A static representation may then be generated for the latest execution path of the game. The latest execution path is from time T1 to T3 followed by the execution path from time T3 to T4. The time-based media information corresponding to the latest simulation execution path is the time-based media information recorded for the execution of the simulation from T1 to T3 and the time-based media information recorded for the execution of the simulation from T3 to T4. The static representation for the latest execution may then be printed on a paper medium. As described above, the static representation may comprise time tags corresponding to times T1 and T4 and time points between T1 and T4.

A user may then select a particular time tag printed on the paper document and replay a portion of the time-based media information recorded for the simulation run from T1 to T4. If time tag T1 is selected, then the time-based media information played back will be the time-based media information for
T1 to T3 followed by the time-based media information for T3 to T4. If a tag corresponding to time T5 is selected for replay, then a portion of the time-based media information recorded from time T5 to T4 is replayed. Instead of replaying the time-based media information from time T5, the user may instead request generation of the simulation from time T5. This causes a new execution path to be branched from the execution path from T3 to T4 at approximately time T5. The starting state of the simulation at the start (i.e., T5) of the new execution path is the same as the state of the simulation at time T5 during the previously executed path from T3 to T4. The state information captured up to time point T5 from a previous execution(s) may be used to determine the state of the simulation at time T5. The new execution path is based upon the state information determined for T5. Simulation engine 102 may then execute a new path from time T5 to time point T6, as shown in FIG. 6C. Time-based media information and state information may be recorded for the simulation execution path from T5 to T6.

As depicted in FIGS. 6A, 6B, and 6C and described above, various execution paths may result from restarting the execution of the simulation program from specified time points during previous execution paths of the simulation program. For the execution paths depicted in FIGS. 6A, 6B, and 6C, the “latest” run execution path is the path from T1 to T3, followed by T3 to T5, followed by T5 to T6. Accordingly, if the user requests generation of a static representation for the “latest” run, then the time-based media information used for generating the static representation is the time-based media information recorded for the execution from T1 to T3, followed by T3 to T5, followed by T5 to T6. A portion of the time-based media information may be replayed using the paper medium on which the static representation is printed. The paper document may also be used to restart the simulation from a time point from T1 to T3, T3 to T5, and T5 to T6. The simulation can thus be restarted using a paper interface from any time point during the duration of a previously completed simulation.

FIG. 7 is a simplified high-level flowchart 700 depicting a method of generating a static representation for a simulation according to an embodiment of the present invention. The method depicted in FIG. 7 may be performed by software code modules or instructions executed by a processor, hardware modules, or combinations thereof. Flowchart 700 depicted in FIG. 7 is merely illustrative of an embodiment of the present invention and is not intended to limit the scope of the present invention. Other variations, modifications, and alternatives are also within the scope of the present invention. The method depicted in FIG. 7 may be adapted to work with different implementation constraints.

As depicted in FIG. 7, processing is initiated upon receiving a signal to generate a static representation for a simulation (step 702). Static representations may be generated for various different kinds of simulations such as video games, flight simulations, etc. The signal may identify the simulation for which the static representation is to be generated.

Information may be received specifying the time duration(s) of the simulation for which the static representation is to be generated (step 704). Various user interfaces may be provided to enable the user to specify the time duration(s) for which the static representation is to be generated. For example, a user interface such as the print dialog box depicted in FIG. 2 may be provided to enable the user to specify the time durations. Using print dialog box 202, a user may specify a time duration for which a static representation is to be generated by specifying the start point and the end point of the time duration. Non-contiguous time durations may also be specified. For example, for a particular simulation, multiple non-contiguous time durations may be specified, each duration identified by a start time and an end time. In situations where a specific time duration is not specified, the static representation may be generated for the full time-based media information.

Other information that influences the generation of the static representation may also optionally be received (step 706). The other information may include for example layout and style information, print parameters, etc. The layout and style information may specify the layout, format, and styles to be used for generating the static representation. Print parameters may identify the paper medium on which the static representation is to be printed, the size of the paper medium (e.g., A4 size paper, 8½ by 11 size paper), whether the static representation is to be printed in landscape or portrait format, the properties of a printer that is to be used for the printing, etc.

If specific events that occur during the simulation are to be included in the static representation then information identifying those events may be received. U.S. application Ser. No. 11/021,303, depicts examples of the different kinds of information that influences generation of static representation and that may be received in 706 according to an embodiment of the present invention.

Although steps 702, 704, and 706 are shown as being performed sequentially in FIG. 7, the steps may be performed in any order or even in a single step.

The time-based media information to be used for generating the static representation is then determined (step 708). As described above, with regards to FIGS. 6A, 6B, and 6C, multiple execution paths may exist for a simulation application. Accordingly, in 708, a specific execution path and time-based media information corresponding to the execution path is determined.

In one embodiment, in the default mode, the “latest” execution run may be selected for generation of the static representation. For example, in FIG. 6C, the execution path from T1 to T3, T3 to T5, and T5 to T6 represents the “latest” path and its corresponding time-based media information may be selected for generation of the time-based media information. A portion (which may be the entire time-based media information) of the determined time-based media information is then selected corresponding to the time durations specified for static representation generation. In alternative embodiments, if multiple execution paths exist, the user may be prompted to select a particular execution path for generation of the static representation from a list of available execution paths.

A static representation is then generated for the time-based media information determined in 708 (step 710). The information received in 706 may be used to generate the static representation. Examples of techniques that may be used to generate a static representation in 710 are described in the various applications incorporated herein by reference. The static representation generated in 710 may have all or some of the features of static representations described above.

The static representation generated in 710 may be output to a user via a display device (step 712) or may be printed on a paper medium (step 714) to generate a paper document on which the static representation is printed. The printed paper document may then be used to replay or restart the simulation.

FIG. 8 is a simplified high-level flowchart 800 depicting a method of processing tags read from a static representation printed on a paper medium according to an embodiment of the present invention. The method depicted in FIG. 8 may be
performed by software code modules or instructions executed by a processor, hardware modules, or combinations thereof. Flowchart 800 depicted in FIG. 8 is merely illustrative of an embodiment of the present invention and is not intended to limit the scope of the present invention. Other variations, modifications, and alternatives are also within the scope of the present invention. The method depicted in FIG. 8 may be adapted to work with different implementation constraints.

Processing is initiated upon receiving information identifying a tag read from a paper medium (step 802). The tag may be read from a paper document on which a static representation is printed. However, the static representation is not essential for the present invention. For example, a paper page may simply have one or more tags printed on it, each tag corresponding to a time point. The information received in 802 may be received responsive to a tag being read from such a paper page. Various different types of tags and tag readers may be used. In the static representations shown in FIGS. 3, 4, and 5B, the tags are in the form of barcodes. These tags may be read by a device configured to read barcodes (e.g., a barcode reader).

A check is then made to see if the tag that is read is a tag that enables activation of modes (referred to as a “mode tag”) (step 804). If the tag is determined in 804 to be a mode tag, then the present mode of operation is determined (step 806). If the mode tag is used to toggle between modes, then the present mode is then switched to the next mode (step 808). Alternatively, if the mode tag identifies a specific mode then the mode corresponding to the mode tag may be activated in 808.

The number of modes that are provided may be different in different embodiments. As described above, in one embodiment, a replay mode and a restart mode are provided, and a single mode tag is provided for toggling between the two modes. For example, if the present mode determined in 806 is “replay” mode, then in 808 the mode is switched to “restart” mode. If the present mode determined in 806 is “restart” mode, then in 808 the mode is switched to “replay” mode.

In alternative embodiments, separate mode tags may be provided for activating different modes. For example, a “replay mode tag” may be printed in the static representation for activating replay mode and a separate “restart mode tag” may be printed for activating replay mode. Based upon the scanned mode tag, the corresponding mode is activated.

More than two modes may be provided in alternative embodiments. The static representation printed on the paper medium may comprise a single mode tag such that scanning the tag cycles through the various available modes. In alternative embodiments, a separate tag may be provided for each mode for activating that mode.

If it is determined in 804 that the tag is not a mode tag, then a check is made to see if the tag is a time tag (step 810). A time tag is a tag encoding or representing a particular time, for example, example tags 308 (FIG. 3), tags 402 (FIG. 4), tags 508 (FIG. 5B). An action tag is a tag that encodes information identifying an action to be performed, for example tags 316 (FIG. 3), tags 514 (FIG. 5B).

If the tag is determined in 810 to be a time tag, then the time corresponding to the tag is determined (step 812). Assuming that there are two possible modes, i.e., replay mode and restart mode, a check is then made to determine whether the present mode is a replay mode or restart mode (step 814).

If the mode is determined to be “replay” mode, then a portion of time-based media information to be replayed in response to the scanned tag is determined (step 816). The processing in 816 involves determining time-based media information for the simulation for which replay is requested and determining the portion of the time-based media information to be replayed.

As described above (as depicted in FIGS. 6A, 6B, and 6C), multiple execution paths may exist for a simulation program and the time determined for the time tag in 812 may represent a time point in more than one execution path. In such a scenario, as part of 816, a determination is made as to which specific execution path is to be selected for replay. In one embodiment, by default, the latest simulation path is selected.

As previously described, the time-based media information for the latest execution path of the simulation may be the concatenation of time-based media information recorded previously for separate execution path time segments concatenated together. For example, for the execution paths depicted in FIG. 6C, the time-based media information for the latest execution path is the concatenation of the time-based media information recorded from T1 to T3, followed by T3 to T5, followed by T5 to T6.

Embodiments of the present invention also enable replay of time-based media information for execution paths other than the latest execution path. For example, in one embodiment, in addition to specifying a time point, the time tag may also convey information identifying a specific execution path for which replay is requested. The specific execution path may be some path other than the latest path. For example, for the execution paths depicted in FIG. 6C, the execution path from T1 to T3 and T3 to T4 may be specified. In this case, the time-based media information for the execution path is the time-based media information recorded from T1 to T3 concatenated with the time-based media information recorded from T3 to T4. A portion of time-based media information corresponding to this execution path may be replayed from a time corresponding to the scanned time tag.

In alternative embodiments, information may be stored identifying the various execution paths for a simulation program. For a particular time point corresponding to the tag, a lookup table may be consulted to determine all the available execution paths for that time point. Information identifying the available execution paths may be provided to a user via a user interface. The user may then select a particular execution path for replay. Time-based media information for the user-selected execution path is then determined in 816 from previously recorded time-based media information for the simulation. The time-based media information for a specific path may comprise time-based media information concatenated from various time segments.

Time-based media information is accessed for the specific execution path for which information is to be replayed. From the time-based media information for the selected execution path, a portion of the time-based media information corresponding to the time represented by the scanned time tag is then determined. The entire time-based media information may be selected for replay if the time represented by the scanned time tag is the starting time of the simulation. Alternatively, a portion of the time-based media information (from the time represented by the scanned time tag) may be selected for replay.

Playback or replay of the time-based media information determined in 816 is then initiated (step 818). Replaying the time-based media information causes playback of the previously recorded time-based media information determined in 816. Various techniques may be used to replay the time-based media information. In one embodiment, an application (e.g., a media player) may be invoked for replaying the time-based media information. In another embodiment, the simulation engine 102 may be used to replay the time-based media.
information. For example, the replay may be output via output device 104 (see FIG. 1) coupled to simulation engine 102. Other techniques may also be used to replay the time-based media information corresponding to the time tag.

Referring back to step 814, if it is determined that the present mode is the restart mode, then state information corresponding to the time represented by the time tag is determined from previously recorded state information (step 820). As previously indicated, multiple execution paths may exist for a simulation program. In such a scenario, the time point determined in 812 for the tag may represent a time point in more than one execution path. In such a scenario, as part of 820, a determination is made as to which specific execution path is to be selected for restarting execution of the simulation program. In one embodiment, by default, the latest simulation path is selected and state information determined for the selected path. The state information for the latest execution path of the simulation may be the concatenation of state information recorded previously for separate execution path time segments. For example, for the execution paths depicted in FIG. 6C, the state information for the latest execution path is the concatenation of state information recorded from T1 to T3, followed by T3 to T5, followed by T5 to T6.

Embodiments of the present invention also enable restart from an execution path other than the latest execution path. For example, in one embodiment, in addition to specifying a time point, the time tag may also convey information identifying a specific execution path from which restart is desired. The specific execution path may be some path other than the latest path. For example, for the execution paths depicted in FIG. 6C, the execution path from T1 to T3 and T3 to T4 may be specified. In this case, the state information for the specified execution path is the state information recorded from T1 to T3 concatenated with the state information recorded from T3 to T4.

In alternative embodiments, as previously described, information may be stored identifying the various execution paths for a simulation program. For a particular time point corresponding to the tag, a lookup table may be consulted to determine all the available execution paths for that time point. Information identifying the available execution paths may be provided to a user via a user interface. The user may then select a particular execution path to be used for the restart operation. State information for the user-selected execution path is then determined in 820 from previously recorded state information for the simulation. The state information for a specific path may comprise state information concatenated from various time segments. The state information determined for a specific selected execution path is then used to determine the state information for the time corresponding to the tag.

Execution of the simulation program is then restarted from a time point corresponding to the tag using the state information determined for the time point in 820 (step 822). The restart causes the execution of the simulation program such that the starting state of the simulation at the start (i.e., at the time corresponding to the tag) of the new execution path is the same as the state of the simulation at the time corresponding to the tag during a previously executed simulation path that is specified or selected as the branch out point for the simulation restart. In one embodiment, the time corresponding to the tag and the relevant state information is communicated to simulation engine 102 which then restarts the simulation from the time corresponding to the tag. Other techniques may also be used to restart the simulation in alternative embodiments.

Referring back to 810, if it is determined that the tag is an action tag, then the action corresponding to the tag is performed (step 824). Various actions may be performed. For example, the actions performed may be related to controlling the replay or restart of the time-based media information.

In the manner described above, the paper medium on which a static representation is printed provides a paper-based interface for replaying portions of previously recorded time-based media information for a simulation from a particular user-selected time point (corresponding to the user-selected tag). The paper medium also provides a paper-based interface for restarting execution of a simulation program from a particular user-selected time point (corresponding to the user-selected tag) during a previous execution path of the simulation program. In this manner, random access to a simulation is provided for replay or for restart operations using a paper interface.

As described above, embodiments of the present invention provide a paper-based interface for restarting a simulation, with all the necessary simulation state information, from a user-selectable time point. For example, a player can restart a video game from a particular time point and continue playing the game from that time point onwards. Thus random access to the game itself is enabled. Various techniques have been described above for recording time-based media information generated by the execution of a simulation program such as a video game. Various techniques may also be used for capturing state information for a simulation, e.g., for capturing state information generated by execution of a game. In some embodiments, the capture of state information is built in to the simulation engine 102 that executes the game. The captured state information may then be used for subsequent restarts.

Many games also generate temporary state files during game execution. The state files are named according to the progressive state of the game. These state files are sometimes used as "map" (or level) files in a game as they contain the data corresponding to a particular player's state for that level. A level in a game may comprise several sublevels. For example, if a particular game has 18 levels, each level may contain several sublevels which must be completed to complete the entire level. During game play, state files may be generated and stored for the various mini-levels within a single level. During a player's game play in a level, and as sublevels within that level are completed, several state files are generated after each mini-level completion and stored. The state files may be stored in a special directory called SAVE. Accordingly, for each level, several files may be stored which save the state of the game for that level. The files are generally uniquely named to identify the level and the sublevel to which they correspond. For example, the following files may be stored:

L1A.HL1, L1A.HL2, L1A.HL3—Files for sublevel “A” within level L1
L1B.HL1, L1B.HL2, L1B.HL3—Files for sublevel “B” within level L1, etc.

In addition to the files for the levels and sublevels, state information for a game execution may also be stored in other files. For example, a portion of the state information may be stored in the “autosav.sav” file.

A special daemon or background process may be configured to detect the creation of such state files and other files storing state information during the execution of a game. Such files may be captured and stored for the various execution paths. An entry for each captured file may be made in a log file. Such a log file could be a XML file used for recording state information. Accordingly, using this technique, the state information for a game is captured by capturing the files generated during the game containing the state information.
In alternative embodiments, a special program may be written (e.g., a hook) to capture the state information from the simulation engine itself during execution of the simulation. For example, for a video game, system development kits (SDKs) provided by gaming companies may be used to develop such hook programs. The “hooks” may be used to capture state information.

When a signal is received to restart a game from a particular time point, the previously captured state files are determined for a previous execution path from which restart is to be performed. These state files are used to determine the state for the particular time point. The state information determined for the particular time point is then used to restart the simulation program execution. The determined state files may then be copied to a startup directory that is used by the simulation engine to load the game. Special command files (e.g., autexec.cfg) that are automatically loaded when a game is started may also be created to facilitate the loading process. Customization of some game parameters may be performed. For example, the following command is placed in the autexec.cfg file: “map L1B”. This automatically loads files captured for level and sublevel corresponding to L1B from a directory (e.g., theSAVE directory) to restart the game from that level and sublevel.

In some games, a player is able to manually save state information, for example by manually pressing the F6 button while playing a game. The state information is usually stored in a file called “quicksave.sav”. This file may also be captured and used to restart the game.

User interface input devices 912 may include a keyboard, pointing devices such as a mouse, trackball, touchpad, or graphics tablet, a scanner, a barcode scanner, a touchscreen incorporated into the display, audio input devices such as voice recognition systems, microphones, and other types of input devices. In general, use of the term “input device” is intended to include all possible types of devices and mechanisms for inputting information to computer system 900.

User interface output devices 914 may include a display subsystem, a printer, a fax machine, or non-visual displays such as audio output devices, etc. The display subsystem may be a cathode ray tube (CRT), a flat-panel device such as a liquid crystal display (LCD), or a projection device. In general, use of the term “output device” is intended to include all possible types of devices and mechanisms for outputting information from computer system 900. The output devices may be used to playback time-based media information.

Storage subsystem 906 may be configured to store the basic programming and data constructs that provide the functionality of the present invention. Software (code modules or instructions) that provides the functionality of the present invention may be stored in storage subsystem 906. These software modules or instructions may be executed by processor(s) 902. Storage subsystem 906 may also provide a repository for storing data used in accordance with the present invention. Storage subsystem 906 may comprise memory subsystem 908 and file/disk storage subsystem 910.

Computer system 900 can be of various types including a personal computer, a portable computer, a workstation, a network computer, a mainframe, or any other data processing system. Due to the ever-changing nature of computers and networks, the description of computer system 900 depicted in FIG. 9 is intended only as a specific example for purposes of illustrating the preferred embodiment of the computer system. Many other configurations having more or fewer components than the system depicted in FIG. 9 are possible.

Although specific embodiments of the invention have been described, various modifications, alterations, alternative constructions, and equivalents are also encompassed within the scope of the invention. The described invention is not restricted to operation within certain specific data processing environments, but is free to operate within a plurality of data processing environments. Additionally, although the present invention has been described using a particular series of transactions and steps, it should be apparent to those skilled in the art that the scope of the present invention is not limited to the described series of transactions and steps.

Further, while the present invention has been described using a particular combination of hardware and software, it should be recognized that other combinations of hardware and software are also within the scope of the present invention. The present invention may be implemented only in hardware, or only in software, or using combinations thereof.

The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. It will, however, be evident that additions, subtractions, deletions, and other modifications and changes may be made
thereunto without departing from the broader spirit and scope of the invention as set forth in the claims.

What is claimed is:

1. A method performed by a computer system of initiating execution of a video game, the method comprising:
   - receiving information identifying a bar code read from a paper document, the bar code comprising time information corresponding to an event in an execution path of a previously executed video game;
   - determining a first time point using the time information;
   - determining, from state information stored for the previously executed video game, first state information corresponding to a particular state of execution of the previously executed video game based on the first time point;
   - accessing the first state information from the stored state information;
   - causing a data processor in the computer system to execute the video game from the first time point using the first state information; and
   - outputting images generated during execution of the video game to an output device.

2. The method of claim 1 wherein determining the first state information comprises:
   - determining a first execution path of the video game for the first time point from a plurality of execution paths of the video game;
   - determining state information for the first execution path from the state information stored for the video game; and
   - determining the first state information for the first time point from the state information determined for the first execution path.

3. The method of claim 2 wherein determining the first execution path comprises:
   - outputting information identifying the plurality of execution paths; and
   - receiving information indicating selection of the first execution path from the plurality of execution paths.

4. The method of claim 2 wherein determining the first execution path comprises:
   - determining the first execution path from the first tag.

5. The method of claim 1 further comprising:
   - accessing time-based media information recorded for the previous execution of the video game;
   - generating a static representation for the time-based media information, wherein the static representation comprises:
     - a visual representation of a portion of the time-based media information;
     - a set of tags, each tag representing a time point, the set of tags including the bar code; and
     - printing the static representation on a paper medium to produce the paper document.

6. A method of replaying information, the method executed on a computer system, the method comprising:
   - receiving information identifying a bar code read from a paper document, the bar code comprising time information corresponding to an event in an execution path of a previously executed video game;
   - determining a first time point using the time information;
   - determining, based upon the first time point, a portion of time-based media information recorded for the previously executed video game corresponding to a particular state of execution of the previously executed video game based on the first time point;
   - accessing the portion of the time-based media information; and
   - initiating replay of the portion of the time-based media information using an output device.

7. The method of claim 6 wherein determining the portion of time-based media information comprises:
   - determining a first execution path of the video game for the first time point from a plurality of execution paths of the video game;
   - determining time-based media information for the first execution path from the time-based media information recorded for the video game; and
   - determining the portion of time-based media information from the time-based media information determined for the first execution path.

8. The method of claim 7 wherein determining the first execution path comprises:
   - outputting information identifying the plurality of execution paths; and
   - receiving information indicating selection of the first execution path from the plurality of execution paths.

9. The method of claim 7 wherein determining the first execution path comprises:
   - determining the first execution path from the bar code.

10. The method of claim 6 further comprising:
    - generating a static representation based upon the time-based media information for the previous execution of the video game, wherein the static representation comprises:
      - a visual representation of the portion of the time-based media information; and
      - a set of tags, each tag representing a time point, the sets of tag including the bar code; and
    - printing the static representation on a paper medium to produce the paper document.

11. A method of performing an operation based upon information read from a paper document, the method comprising:
    - receiving information identifying a bar code read from a paper document, the bar code comprising time information corresponding to an event in an execution path of a previously executed video game;
    - determining a first time point using the time information; and
    - determining a present mode;
    - if the present mode is determined to be a first mode:
      - determining, from state information stored for a previous execution of the previously executed video game, first state information for the first time point corresponding to a particular state of execution of the previously executed video game based on the first time point;
      - causing a data processor to execute the video game from the first time point using the first state information, including accessing the first state information from the stored state information;
      - outputting images generated during execution of the video game to an output device; and
    - if the present mode is determined to be a second mode:
      - determining, based upon the first time point, a portion of time-based media information recorded for the previously executed video game corresponding to the particular state of execution of the previously executed video game as indicated by the first time point;
      - accessing the portion of the time-based media information;
      - replaying the portion of the time-based media information; and
25. outputting images generated during replay of the portion of the time-based media information to an output device.

12. The method of claim 11 further comprising:
receiving information identifying a second tag read from
the paper medium; and
changing the present mode responsive to receiving the
information identifying the second tag.

13. The method of claim 11 further comprising:
generating a static representation based upon the time-
based media information for the previous execution of
the video game, wherein the static representation comprises:
a visual representation of the portion of the time-based
media information;
a set of tags, each tag representing a time point, the set of
tags including the bar code; and
a second tag for switching between the first mode and the
second mode; and
printing the static representation on a paper medium to pro-
duce the paper document.

14. A system for executing a video game, the system comprising:
a data processing system;
a simulation engine in communication with the data pro-
cessing system; and
an output device;
wherein the data processing system is configured to:
receive information identifying a bar code read from a
paper document, the bar code comprising time infor-
mation corresponding to an event in an execution path
of a previously executed video game;
determine a first time point using the time information;
determine, from state information stored for the previ-
ously executed video game, first state information
corresponding to a particular state of execution of the
previously executed video game based on the first
time point; and
access the first state information from the stored state
information;
wherein the simulation engine is configured to cause
the data processing system to initiate execution of the video
game from the first time point using the first state infor-
mation;
wherein images generated during execution of the video
game are presented on the output device.

15. The system of claim 14 wherein the data processing
system is configured to determine the first state informa-
tion by:
determining a first execution path of the video game for the
first time point from a plurality of execution paths of
the video game;
determining state information for the first execution path
from the state information stored for the video game; and
determining the first state information for the first time point
from the state information determined for the first execution
path.

16. The system of claim 15 wherein determining the first
execution path comprises:
outputting information identifying the plurality of execu-
tion paths; and
receiving information indicating selection of the first
execution path from the plurality of execution paths.

17. The system of claim 15 wherein determining the first
execution path comprises determining the first execution path
from the bar code.

18. The system of claim 14 further comprising:
a printer device in communication with the data processing
system;
wherein the data processing system is further configured to:
access time-based media information recorded for the
previous execution of the video game;
generate a static representation for the time-based media
information, wherein the static representation com-
prises:
a visual representation of a portion of the time-based
media information;
set of tags, each tag representing a time point, the set of
tags including the bar code; and
wherein the printer device is configured to print the static
representation on a paper medium to produce the paper
document.

19. A system for replaying information, the system com-
prising:
a data processing system; and
an output device coupled to the data processing system;
wherein the data processing system is configured to:
receive information identifying a bar code read from a
paper document, the bar code comprising time infor-
mation corresponding to an event in an execution path
of a previously executed video game;
determine a first time point using the time information;
determine, based upon the first time point, a portion of
the time-based media information recorded for the previ-
ously executed video game corresponding to a partic-
ular state of execution of the previously executed video
game as indicated by the first time point; and
access the portion of the time-based media information;
wherein the output device is configured to replay the portion
of the time-based media information.

20. The system of claim 19 wherein the data processing
system is configured to determine the portion of time-based
media information by:
determining a first execution path of the video game for the
first time point from a plurality of execution paths of the
video game;
determining time-based media information for the first
execution path from the time-based media information
recorded for the video game; and
determining the portion of time-based media information
from the time-based media information determined for the
first execution path.

21. The system of claim 19 further comprising a printer
device coupled to the data processing system, wherein:
the data processing system is configured to generate a static
representation based upon the time-based media informa-
tion for the previous execution of the video game,
wherein the static representation comprises:
a visual representation of the portion of the time-based
media information; and
a set of tags, each tag representing a time point, the sets
of tag including the bar code; and
the printer device is configured to print the static represen-
tation on a paper medium to produce the paper document.

22. A system for performing an operation based upon infor-
mation read from a paper document, the system comprising:
a data processing system;
a simulation engine; and
an output device;
wherein the data processing system is configured to:
receive information identifying a bar code read from a
paper document, the bar code comprising time infor-
mation corresponding to an event in an execution path of a previously executed video game; determine a first time point using the time information; determine a present mode; determine, from state information stored for the previously executed video game, first state information corresponding to the particular state of execution of the previously executed video game as indicated by the first time point and access the first state information from the stored state information, if the present mode is determined to be a first mode; and determine, based upon the first time point, a portion of time-based media information recorded for the previously executed video game corresponding to the particular state of execution of the previously executed video game as indicated by the first time point and access the portion of the time-based media information, if the present mode is determined to be a second mode;

wherein the simulation engine is configured to initiate execution of the video game from the first time point using the first state information if the present mode is determined to be the first mode; and

wherein the output device is configured to replay the portion of the time-based media information if the present mode is determined to be a second mode.

23. The system of claim 22 wherein the data processing system is configured to:

receive information identifying a second tag read from the paper medium; and

change the present mode responsive to receiving the information identifying the second tag.

24. The system of claim 22 further comprising a printer device, wherein:

the data processing system is configured to:

generate a static representation based upon the time-based media information for the previous execution of the video game, wherein the static representation comprises:

a visual representation of the portion of the time-based media information;

a set of tags, each tag representing a time point, the set of tags including the bar code; and

a second tag for switching between the first mode and the second mode; and

the printer device is configured to print the static representation on a paper medium to produce the paper document.

25. An apparatus for initiating execution of a video game, the apparatus comprising:

means for receiving information identifying a bar code read from a paper document, the bar code comprising time information corresponding to an event in an execution path of a previously executed video game;

means for determining a first time point using the time information;

means for determining, from state information stored for a previously executed video game, first state information corresponding to the particular state of execution of the previously executed video game based on the first time point;

means for accessing the first state information from the stored state information;

means for causing a data processor to execute of the video game from the first time point using the first state information; and

an output device to display images generated during execution of the video game.

26. A computer-readable medium storing a plurality of instructions for controlling a data processor for initiating execution of a video game, the plurality of instructions comprising:

instructions that cause the data processor to receive information identifying a bar code read from a paper document, the bar code comprising time information corresponding to an event in an execution path of a previously executed video game;

instructions that cause the data processor to determine a first time point using the time information;

instructions that cause the data processor to determine, from state information stored for the previously executed video game, first state information corresponding to a particular state of execution of the previously executed video game based on the first time point;

instructions that cause the data processor to access the first state information from the stored state information;

instructions that cause the data processor to initiate execution of the video game from the first time point using the first state information; and

instructions that cause the data processor to output images generated during execution of the video game to an output device.

* * * * *