Methods and systems for alternating rendering of information of a common display for a video game are provided by identifying certain data relating to a scene as belonging to a first layer and identifying other certain data relating to a scene as belonging to a second layer. Rendered information of each layer are captured in memory, and rendered information for each layer are used for two successive video frames.
Start

Provide Object with Layer ID

Process Scene Data

Render Object

Store Rendered Object Information

Return

FIG. 3
**FIG. 4**

**FRONT LAYER**

<table>
<thead>
<tr>
<th>Image A1</th>
<th>Image A2</th>
<th>Image A3</th>
<th>Image A4</th>
<th>Image A5</th>
</tr>
</thead>
</table>

**BACK LAYER**

<table>
<thead>
<tr>
<th>Image B1</th>
<th>Image B2</th>
<th>Image B3</th>
<th>Image B4</th>
<th>Image B5</th>
</tr>
</thead>
</table>

frame count
Fig. 5

Layer A
2048 polygons

Layer B
2048 polygons

Composite
4096 polygons
Start

711 Render?

Y

A

Render A

Store A

Display Rendered A

and Stored B

B

Render A or B?

715

721

713

Render B

Store B

Display Rendered B

and Stored A

Swap Render Index

727

N

Exit?

N

Y

Return

FIG. 7
2048 polygons for 2 characters

2048 polygons for 1 character

FIG. 9
DOUBLE RENDER PROCESSING FOR HANDHELD VIDEO GAME DEVICE

BACKGROUND OF THE INVENTION

[0001] The present invention relates generally to handheld devices, and more particularly to image rendering for a handheld video game device.

[0002] Video games provide a source of enjoyment to users by allowing users to engage in simulated scenarios and situations the users may not otherwise be able to experience. Video games receive different types of interactive user inputs, and process the inputs into vibrant interactive visual displays and audio accompaniments for the users to enjoy.

[0003] Handheld video game devices, or other mobile devices providing video game functions, are often preferred over traditional video game consoles due to their convenience and mobility. Because of the relatively small size of most handheld video game devices, handheld video game devices allow for easy transport and playing flexibility in environments which would typically be unsuitable for video game play using traditional video game consoles.

[0004] The tradeoff for the small size and mobility of handheld video game devices is generally manifested in the processing power and video display capabilities of the handheld video game devices. The relatively small housings of handheld video game devices do not allow for the capacity of hardware and processing power of traditional video game consoles. In addition, the smaller platforms allow for only limited screen sizes, further reducing the video display capabilities of handheld video game devices. While recent years have seen marked improvements in the video display capabilities in a number of handheld video game device platforms, generally the capacity of video displays in handheld video game devices still falls far short of video display capabilities of more traditional video game consoles.

BRIEF SUMMARY OF THE INVENTION

[0005] The invention provides for displays of a video game. In one aspect the invention provides a method of providing images for a video game, comprising: associating objects with either a first layer or a second layer; rendering objects associated with the first layer; rendering objects associated with the second layer; displaying the rendered objects associated with the first layer on a display; and displaying the rendered objects associated with the second layer on the display.

[0006] In another aspect the invention provides a method of providing images for a music based video game, comprising: associating a first object with a first display layer, the first object representing a representative of a musician in the music based video game; associating a plurality of background objects with a second display layer, the plurality of background objects representative of a venue in the music based video game; iteratively rendering the first object, storing rendered information of the first object in a memory, and displaying rendered information of the first object on the display; iteratively rendering the plurality of background objects, storing rendered information of the plurality of background objects in a second memory, and displaying rendered information of the plurality of background objects on the display; with displaying rendered information of the first object on the display utilizing the information stored in the first memory in first alternating time periods and displaying rendered information of the plurality of background objects utilizing the information stored in the second memory in second alternating time periods in first alternating time periods, the first alternating time periods and the second alternating time periods occurring at different times.

[0007] In another aspect the invention provides a handheld game system, comprising: memory storing scene data, the scene data including first scene data and second scene data; a processor configured to render the first scene data and the second scene data; first video memory, coupled to the processor, configured to store rendered first scene data; second video memory, coupled to the processor, configured to store rendered second scene data; and a display coupled to the first video memory and the second video memory; the processor being further configured to alternately: a) render the first scene data, command display on the display of the rendered first scene data, command display on the display of the rendered second scene data in the second video memory, and command storage of the rendered first scene data in the first video memory, and b) render the second scene data, command display on the display of the rendered second scene data, command display on the display of the rendered first scene data in the first video memory, and command storage of the rendered second scene data in the second video memory.

[0008] In another aspect the invention provides a method of providing images for a video game, comprising: associating different objects with different layers; rendering objects associated with a first layer of the different layers; storing information of rendered objects associated with the first layer in a first memory; rendering objects associated with a second layer of the different layers; combining information of rendered objects associated with the first layer in the first memory with information of rendered objects associated with the second layer; and displaying the combined information.

[0009] These and other aspects of the invention are more fully comprehended on review of this disclosure.

BRIEF DESCRIPTION OF THE FIGURES

[0010] FIG. 1 illustrates a handheld video game system in accordance with aspects of the invention;

[0011] FIG. 2 is a block diagram of example video processing circuitry for a display of a handheld video game device in accordance with aspects of the invention;

[0012] FIG. 3 is a flow diagram of a process of image rendering in a handheld video game device in accordance with aspects of the invention;

[0013] FIG. 4 is a timeline of an image layer alternating process in accordance with aspects of the invention;

[0014] FIG. 5 is an illustration showing an image combining process in accordance with aspects of the invention;

[0015] FIG. 6 is a block diagram of a handheld video game device in accordance with an embodiment of the invention;

[0016] FIG. 7 is a flow diagram of a process of using separately rendered image layers in accordance with aspects of the invention;

[0017] FIG. 8 illustrates a flow of using separately rendered image layers in accordance with aspects of the invention;

[0018] FIG. 9 illustrates different levels of detail of rendered features in accordance with aspects of the invention;

[0019] FIG. 10 is a flow diagram of a further process of using separately rendered image layers in accordance with aspects of the invention; and

[0020] FIG. 11 illustrates a further flow of using separately rendered image layers in accordance with aspects of the invention.
DETAILED DESCRIPTION

[0021] FIG. 1 is an example of a handheld video game system in accordance with aspects of the invention. The handheld video game system includes a handheld video game device 101, including at least one display, at least one user input device, at least one speaker, and at least one removable memory interface. Each handheld video game device also includes internal circuitry generally associated with video game devices, such as processing circuitry for executing video game instructions and memory used to store video game information. In various embodiments, another handheld device capable of providing video game functions may be used instead of the handheld video game device of FIG. 1.

[0022] In some embodiments, for example, the embodiment as illustrated in FIG. 1, the handheld video game device is a Nintendo DS handheld video game device or a Nintendo DS Lite handheld video game device, both widely available in consumer electronics retail stores. In these embodiments, the handheld video game device incorporates a clamshell design, with a hinge 103 allowing for closure of the handheld video game device, thereby protecting the external components of the handheld video game device while in the closed position. The handheld video game device includes two 3-inch displays, each with a resolution of 256x192 pixels, with a first display 105 located on the top portion 107 of the clamshell housing, and a second display 109 located on the bottom portion 111 of the clamshell housing. In some embodiments, the bottom display may include touchscreen input capabilities.

[0023] In the embodiment as illustrated in FIG. 1, speakers 133 are located on each side of the top display, while a digital directional pad input 115 is located to the left of the bottom display, and a plurality of digital input buttons 117 is located to the right of the bottom display. In some embodiments, additional user inputs may be available on the handheld video game device. The handheld video game device also includes two removable memory interfaces. The first removable memory interface 119 is generally configured to read a removable memory cartridge holding video game instructions. The second removable memory interface 121 is configured to interact with either a removable memory cartridge used in conjunction with an older handheld video game device platform, or a peripheral used in conjunction with video game play of certain video games.

[0024] The handheld video game system illustrated in FIG. 1 also includes a peripheral input device 123 with a plurality of additional input buttons 125. The peripheral input device may be connected to the handheld video game device via the second removable memory interface. In the embodiment as illustrated in FIG. 1, the peripheral input device is used in conjunction with music based rhythm video game.

[0025] In the embodiment as illustrated in FIG. 1, the top display is displaying a screenshot from game play of a music based video game. As is consistent with the Nintendo DS and Nintendo DS Lite handheld video game devices, the graphics processing unit of the handheld video game device in FIG. 1 may be, for example, capable of rendering up to 2048 polygons per frame per image, at a rate of 60 frames per second. However, the screenshot displayed in the top display of FIG. 1 includes double the allowable number of polygons, with for example a maximum display capacity of 4096 total polygons. In accordance with embodiments of the invention, each display of the handheld video game device is capable of displaying at least two separate image layers at the same time. By alternating the generation of two separate image layers, each with display capacity of 2048 polygons, and combining the two image layers, the resulting composite image may be double the traditional image resolution, although the effective frame rate may be reduced. A reduced frame rate, however, may be acceptable, as the new frame rate may be comparable with television and movie frame rates, which typically have refresh rates of between 24 and 30 frames per second.

[0026] FIG. 2 is an example of a block diagram of the video processing circuitry for a display of a handheld video game device in accordance with aspects of the invention. The block diagram of FIG. 2 may represent, for example, the circuitry associated with the top display of the handheld video game device as illustrated in FIG. 1. Video display information is generated by processing video game instructions and user inputs. In some embodiments, a main processor housed in the handheld video game device may begin the video generation process by, for example, compiling and separating scene data and associated video generation information from other video game instructions, and storing the information into memory.

[0027] The scene data and other video generation information is stored in scene data memory 211. In some embodiments, scene data memory may be a dedicated portion of main memory in the handheld video game device. In other embodiments, scene data memory may be separate memory in the handheld video game device, allocated specifically for storage of scene data and other video generation information.

[0028] A graphics processing unit 213 retrieves scene data and other video generation information stored in the scene data memory. The graphics processing unit processes the information from the scene data memory and renders images, for example 2D or 3D images, for display on the video display based on the information. In one embodiment, the graphics processing unit in the handheld video game device performs image rendering at a rate of 60 images per second. The graphics processing unit may alternate image rendering between what may be considered a front layer and what may be considered a back layer. In the context of a music video game, the front layer may be used, exclusively in some embodiments, to render images of a lead character or characters, for example a simulated lead guitarist and a simulated lead singer, while the back layer may be used to render images of a background environment, for example a venue, other band members, and remaining image details. In most embodiments, the front layer may be considered on top of the back layer, thereby blocking display of portions of the back layer. In other words, if a particular pixel in a composite image includes image information for both the front layer and the back layer, the image information for the back layer is occluded. As the lead guitarist and lead singer in the front layer are the main features of the video game footage, by using this arrangement, it is possible to render the lead guitarist and the lead singer in greater detail than the rest of the image.

[0029] In the embodiment of FIG. 2, images rendered by the graphics processing unit are sent directly to a display screen 215 for immediate display purposes, and are simultaneously captured and saved to a bank of video memory 217. The video memory is memory dedicated to storing video processing information, and is generally capable of storing multiple completed video images at any given time. In some embodiments, the graphics processing unit renders images and sends the rendered images simultaneously to the display and video memory. In the embodiment of FIG. 2, a video
memory A, or VRAM A 219, may be dedicated to storing front layer images rendered by the graphics processing unit, and a video memory B, or VRAM B 221, may be dedicated to storing back layer images rendered by the graphics processing unit. In this embodiment, subsequently rendered front layers overwrite a previous front layer stored in VRAM A, and subsequently rendered back layers overwrite a previous back layer stored in VRAM B. Therefore, when a front layer is rendered by the graphics processing unit, the front layer is updated and replaced on the display and in VRAM A, while the back layer displayed is the back layer previously stored in VRAM B. In the next frame, the graphics processing unit renders a new back layer, and the back layer is updated and replaced on both the display and in VRAM B, while the front layer displayed remains the front layer stored in VRAM A from the previous frame.

[0030] The graphics processing unit may continually alternates rendering front layers and back layers in this manner. Therefore, on even frames, for example, the display of the handheld video game device may display a newly updated front layer, and a back layer reused from the previous frame and retrieved from VRAM B. On odd frames, for example, the display of the handheld video game device may alternatively display a newly updated back layer, and a front layer reused from the previous frame and retrieved from VRAM A. As stated previously, the graphics processing unit of the handheld video game device may only be capable of rendering image layers with 2048 polygons. However, the graphics processing unit may be capable of combining two previously rendered image layers into one composite image including more than 2048 polygons. Likewise, the video memory associated with each graphics processing unit is capable of storing images containing more than 2048 polygons, and the display is capable of displaying images containing more than 2048 polygons. Using an alternating layer rendering approach as described, the handheld video game device is therefore capable of displaying videos with double the original resolution capacity of the handheld video game device.

[0031] FIG. 3 is a flow diagram of a process of image rendering in a handheld video game device in accordance with aspects of the invention. The process may be performed, for example, using the graphics processing unit of the top display of the embodiment as described in FIG. 1. In block 311, the process associates objects with a layer. In some embodiments some objects may be associated with a first layer and some objects may be associated with a second layer. In some embodiments association of an object with a layer may be accomplished by storing information in memory correlating objects with a particular layer. In some embodiments the information may be stored in a table, for example, in memory separate from the objects.

[0032] In block 311, the process processes scene data. Scene data may include, for example, video game instructions from a removable memory including information for running the particular video game being played. Scene data may also include, for example, user inputs generated through video game play, retrieved from, for example, user input apparatuses built into the handheld video game device, or from, for example, a peripheral device as illustrated in FIG. 1, or from, for example, input signals from another handheld video game device via a wireless connection interface. Some data may also originate from the main processor of the handheld video game device, which also processes video game instructions and user inputs to generate, for example, video game states associated with the proper functionality of the video game, or from the main memory of the handheld video game device, which stores generated video game states that may include information on image rendering.

[0033] In block 313, the process, usually by way of a graphics processing unit, renders objects within an image. Generally, the process renders objects associated with different layers at different times. For example, using an example with two layers, the process may render objects associated with a first layer during a first time period, render objects associated with a second layer during a second time period, and repeat the rendering of objects in different layers in an alternating manner, thereby effectively rendering a first layer of an image and a second layer of an image in an alternating manner.

[0034] In block 317, the process stores the information associated with each object generated during the image rendering process, or in other words, stores results of rendering objects. In some embodiments rendered object information for different layers are stored in different memory. In addition, in some embodiments the process may also command display of the rendered objects. In embodiments where images are rendered polygon by polygon, the process may generate a variety of information pertaining to each polygon. For example, the polygon shape and color is generated in block 313, and the polygon layer identification information and exact display location of the polygon within the layer is generated in block 315. After the rendered object information has been generated and compiled, it is stored in memory until the entire image layer has been successfully rendered. For example, the object information may be stored in the video memory associated with the graphics processing unit, or alternatively, the object information may be temporarily stored in the main memory of the handheld video game device. An image layer may be considered to be successfully rendered when all the objects to be rendered associated with the layer have been compiled, and the graphics processing unit can use the compilation of object information to render a completed image layer.

[0035] It should be recognized that in some embodiments the process performs the operations of block 311, relating to association of objects with layers, prior to or when storing game data on, for example, a game cartridge or other memory storing video game instructions and data, and the process may thereafter repetitively perform the operations of blocks 313, 315, and 317 during game play.

[0036] The process afterwards returns. The process may be repeated based on the object generation progress of the image layer being rendered and on the image rendering requirements of the graphics processing unit.

[0037] FIG. 4 is a timeline of the image layer alternating process in accordance with aspects of the invention. In the embodiment as illustrated in FIG. 4, image A1 411, representing the first front layer, is rendered and sent to the display when the process begins at frame 0. In accordance with FIG. 2, A1 is also stored into VRAM A, the video memory slot allocated for storage of the most recent front layer image.

[0038] At the next frame, frame 1, image B1 413, representing the first back layer, is rendered and displayed on the video display, and stored into VRAM B, the video memory slot allocated for storage of the most recent back layer image. In accordance with embodiments of the invention, the display of the handheld video game device is capable of displaying at least two image layers at the same time. Therefore, during frame 1, image A1 is not replaced in VRAM A, and is instead
recycled and combined with the new image B1 into a composite image making up the complete screenshot.

[0039] At the next frame, frame 2, image A2 415, representing the second front layer, is rendered and displayed on the video display, while simultaneously stored into VRAM A. When stored into VRAM A, image A2 overwrites and replaces the previous image A1, so that there is only one front layer image stored in VRAM A at any given time. Image B1, which is still stored in VRAM B, is reused, and image A2 is rendered on top of image B1, completing the screenshot.

[0040] At frame 3, image B2 417, representing the second back layer, is rendered and displayed on the video display, and stored into VRAM B, overwriting the previous image B1. This process is similar to the storage process associated with VRAM A during frame 2. The new image B2 is layered with image A2 to create the composite screenshot at frame 3. As can be seen in FIG. 4, this process may be repeated, so that a new front layer image is rendered and stored on every even frame, and a new back layer image is rendered and stored on every odd frame. In an embodiment where 60 images are rendered per second, each layer, and consequently each complete screenshot, is only re-rendered 30 times per second. However, a refresh rate of 30 images per second is on par with, for example, television and movies, which typically employ a refresh rate of between 24 and 30 images per second.

[0041] FIG. 5 is an illustration showing the image combining process in accordance with aspects of the invention. Layer A is a rendered front layer 511, including a lead singer 513 and a lead guitarist 515. Layer A, as illustrated in FIG. 5, can display a maximum of 2048 polygons. As the layer A illustrates does not include any background imagery, such as for example, backup singers and venue imagery, the 2048 polygons are used exclusively for rendering of the lead singer and the lead guitarist. In contrast, when the entire screenshot is rendered as one image, the 2048 polygons are used for the whole image, including the lead singer, the lead guitarist, and all of the associated background imagery. Thus, the invention allows for the lead singer and lead guitarist to be rendered in much greater detail than would be possible when the entire screenshot is rendered as one image.

[0042] Likewise, layer B is a rendered back layer 517, including all the background imagery associated with the screenshot. Similar to layer A, layer B can display a maximum of 2048 polygons as well. In this fashion, the background imagery may utilize an increased number of polygons, as the polygons of the 2048 polygons that would otherwise be associated with the lead singer and/or lead guitarist can be used instead to enhance details in the background imagery.

[0043] The rendered layers A and B are combined into one composite screenshot 519 including both layers. With both the front layer and the back layer capable of displaying up to 2048 polygons, the composite layer thereby has a maximum rendering capability of 4096 polygons, equivalent to double the video display resolution of traditional image rendering on similar handheld video game devices. In the embodiment as illustrated in FIG. 5, the portions of layer B where features also exist in layer A are displayed as null 521, creating silhouette shapes of the lead guitarist and lead singer in the illustrated layer B. In some embodiments, pixels from layer B are occluded if the pixels already contain objects in layer A, so that layer B contains null spaces upon rendering similar to the null spaces illustrated in FIG. 5. The composite image is thus layered neatly upon combining of layers A and B, without any overlapping pixel information between the two layers. In other embodiments, layer B may be a fully rendered image layer, with the aforementioned null spaces also filled in with background image details. In these embodiments, front layer A may be laid atop back layer B for every frame, before being sent to the display, thereby covering and occluding all layer B pixels where layer A objects already exist.

[0044] FIG. 6 is an example of a block diagram of the internal circuitry of a handheld video game device in accordance with another embodiment of the invention. The internal circuitry includes a bus 601 coupling together a processor 603, a main memory 605, a graphics processing unit for a first display 607 and a graphics processing unit for a second display 609, an audio driver 611, and a plurality of different interfaces. The interfaces may include a user input/output (I/O) interface 613, a removable memory interface 615, a peripheral interface 617, and a wireless communication interface 619. In the embodiment as illustrated in FIG. 6, the graphics processing unit for the first display is coupled to a first allocation of video memory 621 dedicated to the first display through a dedicated bus 623. Likewise, the graphics processing unit for the second display is also coupled to a second allocation of video memory 625 dedicated to the second display through another dedicated bus 627. In some embodiments, the graphics processing units may be integrated into the processor, so that the processor performs all graphics processing tasks. In other embodiments, there may be multiple processors, with each processor having separate data bus connections.

[0045] Handheld video game devices generally integrate displays, speakers, and user inputs directly into the handheld video game device. FIG. 6 incorporates the integrated components, with the audio driver coupled to the speakers 629 in the handheld video game device, the graphics processing unit and video memory for a display 1 coupled to display 1 631, the graphics processing unit and video memory for a display 2 coupled to display 2 633, and the user I/O reflected as an integrated component rather than an interface.

[0046] The removable memory interface of the handheld video game device is configured to communicate with a removable memory, for example, a video game cartridge providing video game instructions related to the operation of a specific video game. The processor executes the video game instructions from the removable memory by communicating with each component, including the removable memory, via the bus. The main memory receives and stores information from the other components as needed for the video game to run properly. Stored information may include, for example, video game play instructions, input processing instructions, audio and video generation information, and configuration information from the removable memory, as well as user inputs from either the user I/O or the peripheral interface. The processor adjusts the video game's audio and video properties based in large part on a combination of video game processing instructions from the removable memory and the user inputs. A processor may also receive additional video game instructions and inputs from other handheld video game devices via the wireless communication interface, for example, during wireless multiplayer game play.

[0047] The processor of the handheld video game device receives and processes the video game instructions and inputs, and generates audio and video information for the video game based on the instructions and inputs. The audio
driver is configured to receive the audio information from the processor, and to translate the audio information into audio signals to be sent to the speakers.

[0048] In an embodiment of a handheld video game device associated with the invention, the graphics processing unit renders images with a maximum resolution of 2048 polygons per image, at a frame rate of 60 images per second. The graphics processing unit for each display is configured to retrieve video generation information, and to translate the video generation information into display images to be sent to the display coupled to the graphics processing unit, to the video memory coupled to the graphics processing unit, or to both. In some embodiments, such as embodiments of the invention and the embodiment as illustrated in FIG. 6, each display may be configured to receive the display images from both a graphics processing unit and a video memory. In these embodiments, each display may retrieve image information from both the display’s graphics processing unit and the display’s video memory substantially simultaneously. If two images, each with a maximum rendering capacity of 2048 polygons, are combined and displayed together, the resulting composite image would consequently have a maximum resolution of 4096 polygons, double the original image rendering capacity.

[0049] FIG. 7 is a flow diagram of a process of rendering separate image layers in accordance with aspects of the invention. In block 711, the process determines whether to perform image rendering. If the process determines not to render images, the process proceeds to block 729, and determines whether to exit the system or reinitiate the image rendering process. If, however, the process proceeds with image rendering, the image rendering process is performed and repeated.

[0050] In block 713, the process determines whether to render a first image layer A or a second image layer B. In some embodiments, the first image layer A may be a front layer, and the second image layer B may be a back layer, where the front layer is always layered atop the back layer, and features of the front layer occlude features of the back layer at pixels where objects are rendered for both layers. In some embodiments, certain features associated with each screenshot will be grouped into either a front layer A or a back layer B. For example, in the embodiments associated with the invention, polygons used to generate images of the lead singer and the lead guitarist are grouped into front layer A, and the remaining polygons, which may be used to generate image details including backup singers or venue, may be grouped into back layer B. In other embodiments, polygons falling within a predefined portion of the display, for example, polygons located in the left half of the display image, may be included in layer A, and polygons falling outside the predefined portion of the display, the right half of the display image in this example, may be included in layer B. The process may include a render index to help determine whether to render layer A or layer B. The initial render index value may be arbitrary, or may be preset to either layer A or layer B. If layer A is to be rendered, the process proceeds to block 715. Similarly, if layer B is to be rendered, the process proceeds to block 721.

[0051] In block 715, the process renders image layer A. The graphics processing unit retrieves video information associated with features included in image layer A, and processes the video information to generate polygon information for the image layer A features. Video information associated with image generation may be initially processed by the processor associated with the handheld video game device, and may include video game information stored in the removable memory and/or user input signals originating from input buttons, an attached peripheral, or a wireless communication interface. The video information may be temporarily stored in, and retrieved by a graphics processing unit from, the main memory of the handheld video game device. In embodiments associated with the invention, the process includes rendering a front layer A, including polygons used to generate a simulated lead singer and a simulated lead guitarist with respect to a music based video game. In other embodiments associated with other video games, different features of video display images may be grouped with and rendered as image layer A.

[0052] In block 717, the process stores image layer A. Storage space may be provided by a video memory connected to the graphics processing unit. The video memory may have the capacity to store and sort multiple images simultaneously. In embodiments of the invention, the video memory is capable of storing at least two images in two separate memory allocations at any given time. The process stores the rendered image layer A into one of the memory slots, which may be labeled, for example, video memory slot A. In some embodiments of the invention, the rendering process of block 715 and the storing process of block 717 may be performed in conjunction with each other. In other words, when an object in an image layer is rendered during the rendering process, the newly rendered object may immediately be stored into the associated video memory slot before another object in the image layer is rendered. In other embodiments, the entire image layer may be rendered before storage of the completed image layer into the video memory slot A. If video memory slot A is occupied with a previously rendered image layer A, the previously rendered image layer A is overwritten and replaced by the newly rendered image layer A.

[0053] In block 719, the process displays the rendered image layer A and a second image layer B stored in a second video memory slot dedicated to holding rendered layer B images. The process may combine the two image layers into one composite image before sending the display information to the video display. Alternatively, the process may send the image layers separately, and layer the images on top of one another on the display. If the process is in its first iteration, and no image layer B has yet been rendered, the process may display the rendered image layer A alone, or alternatively, the process may display a blank image layer B.

[0054] In block 721, the process renders image layer B. The rendering process closely mirrors the rendering process for image layer A as described in block 715. The graphics processing unit retrieves video information associated with features included in image layer B, and processes the video information to generate polygon information for the image layer B features. The features included in image layer B may be the features in a typical screenshot of the video game which were not rendered in image layer A. In the context of a music based video game, the image layer B may be a background layer, which includes polygons associated with background imagery, for example, background singers, the remaining band members, and the venue.

[0055] In block 723, the process stores image layer B. The storing process again closely mirrors the storage process for image layer A as described in block 717. Image layer B may be stored in a separate memory allocation in a video memory slot B, or a similar memory allocation dedicated to the storage
of newly rendered layer B images. In some embodiments, the image layer B storing process in block 723 may be performed in conjunction with the image layer B rendering process in block 721. In other embodiments, an entire image layer B may be rendered before storage into the video memory slot B. If video memory slot B already holds a previously rendered image layer B, the previously rendered image layer B is overwritten and replaced by the newly rendered image layer B.

In block 725, the process displays on a video display a composite image including the rendered image layer B and the image layer A stored in video memory slot A. In most embodiments, the graphics processing unit recombines the rendered image layer B and the stored image layer A before sending the composite image to the video display. The image layer defined to be the front layer, generally image layer A as has been described herein, may be layered atop the image layer defined to be the back layer, generally image layer B in the described embodiments, thereby occluding objects in the back layer. In other words, pixels where both layers have object information will display the pixel information of the front layer, and the object information for the pixel included in the back layer will not be displayed.

In block 727, the process swaps the render index. If image layer A was rendered and stored in the previous iteration, in other words, if the process performed the tasks associated with blocks 715, 717, and 719, the render index is switched from A to B. Likewise, if image layer B was rendered and stored in the previous iteration, meaning the process performed the tasks associated with blocks 721, 723, and 725, the render index is switched from B to A. Therefore, upon the next iteration of the process, the process will render the image layer which was not rendered in the previous iteration of the process. Swapping the render index allows for image layer A and image layer B to be alternately rendered, thereby allowing for generation of a completely original screenshot every two iterations, and 30 times every second for embodiments where rendering is performed at a rate of 60 images per second.

In block 729, the process determines whether to exit the image rendering process. If the process determines to remain in image rendering, the process cycles back to render either a new image layer A or a new image layer B, depending on the current render index. If the process determines to exit image rendering, the process returns.

FIG. 8 is an illustration of an embodiment of rendering separate image layers in accordance with aspects of the invention. For both even frames and odd frames, a front layer 811 and a back layer 813 are used to generate a composite image 815 making up a screenshot of the video game. The front image layer may include particular features of the screenshot, for example, polygons used to render the lead singer 817 and the lead guitarist 819 in the music based video game. The back image layer may include other features of the screenshot, for example, polygons used to generate the remaining aspects of the screenshot not rendered in the front image layer. In the embodiments as have been described herein, the back layer may include polygons used to render the backup singers, the remaining members of the band, and the stage or venue. Separate image layer rendering allows for more definition within each image layer, as the maximum number of polygons may be dedicated to the features of each layer rather than allocated across an entire screenshot of the video game. For example, the front image layer may dedicate all 2048 polygons in the image layer to rendering the lead guitarist and the lead singer, rather than allocating only a fraction of the 2048 polygons to the two main characters and using the rest of the polygons to generate the simulated surroundings around the two main characters.

In the embodiment of FIG. 8, on even frames, an image A 821 is rendered by the graphics processing unit of the handheld video game device. The newly rendered image A is both stored into a video memory slot A 823, and used to generate the composite image making up a screenshot of the video game. In the embodiment as illustrated in FIG. 8, the image A may be used as a front layer in the composite image screenshot, and may include, for example, polygons used to render the lead singer and lead guitarist, as has been previously described. During the even frames, the graphics processing unit also retrieves a stored image B 825 in a video memory slot B 827. In the embodiment as illustrated in FIG. 8, the retrieved image B may be used as a back layer in the composite image screenshot, and may include, for example, polygons used to render features of the screenshot not rendered in image A. The newly rendered image A and the retrieved image B are combined into a composite image and sent to the video display as a complete screenshot of the video game.

During odd frames, a new image B 827 is rendered by the graphics processing unit. In the embodiment as illustrated in FIG. 8, the newly rendered image B is used in the composite screenshot of the video game as the back layer of the composite screenshot. During the odd frames, the newly rendered image B is also stored into video memory slot B for later retrieval and use, for example, for composite image generation during the even frames. During the odd frames, the graphics processing unit also retrieves the image A which was stored in the video memory slot A during the previous even frame. The graphics processing unit combines the retrieved image A and the newly rendered image B into a composite image, and sends the composite image to the video display as a complete screenshot of the video game.

In the embodiment as illustrated in FIG. 8, the alternating rendering process may be completed repeatedly at a rate of 60 frames per second, generating a new image A on even frames and a new image B on odd frames, thereby refreshing the entire screenshot every two frames. The effective refresh rate of the video display in a handheld video game device which renders 60 images per second is therefore 30 new screenshots per second.

FIG. 9 is an illustration demonstrating different available levels of detail of rendered features in accordance with aspects of the invention. The first image 911 is an example of an image layer, a front image layer in most embodiments of the invention as described herein, and includes two characters from a music based video game. The characters may represent a lead guitarist 913 and a lead singer 915. In rendering a front layer such as the first image, 2048 polygons are used to render the two characters. The lead guitarist may be rendered using a predefined portion of the available polygons, for example, 60% of the available polygons. The lead singer may be rendered using the remaining polygons not used to render the lead guitarist. The two characters therefore split the available polygons, and each character is rendered using a significantly smaller number of polygons than the 2048 polygon limit for each image layer.

In some situations, a second level of detail of an image layer may be available. For example, a zoomed in shot
or close-up of one of the characters may be desired. A close-up of, for example, the lead guitarist character may be rendered when a particular task, such as a high score, has been achieved in the context of the video game. The second image 917 is an example of an image layer including only one character; a lead guitarist 919, from the music based video game. The second image illustrates a second level of detail at which the front layer may be rendered, and may be used interchangeably with the first image as a front layer in the embodiments of the invention described herein. Because there is no lead singer in the second image, the 2048 polygons in the image layer may be completely dedicated to rendering the lead guitarist, and the lead guitarist at the second level of detail is rendered at a much higher resolution than the lead guitarist at the first level of detail.

[0065] In various aspects, the invention may allow for the capability to generate different levels of detail for either one image layer alone, or for both image layers. In some embodiments, such as the embodiment as illustrated in FIG. 9, generation of different levels of detail may only apply to rendering of the front layer. Traditionally, when a single 2048 polygon image was rendered, any modification to the level of detail of one rendered feature impacted the rendering process of the rest of the image, whether it be a different polygon allocation for the other image features, or a different set of pixels occluded by the feature at the second level of detail. Consequently, the entire image would have to be re-rendered. In contrast, having different levels of detail for only one image layer allows for the second image layer to remain unmodified. The level of detail for the first image layer may be interchangeable and adjustable independent of the image rendering process of the second image layer. Therefore, for example, while the front layer in the embodiment illustrated in FIG. 9 may be switched between two different levels of detail, either front layer may be combined with the same back layer to create a composite screenshot of the video game. Furthermore, the level of detail of the front layer may be changed at any time without having to re-render the back layer to accommodate the front layer’s change in level of detail.

[0066] FIG. 10 is a further flow diagram of a process for providing images for a display, for example for a video game. The process may be performed, for example, by a hand-held video game device configured to execute a video game, for example a music based video game.

[0067] In block 1011 the process determines whether to render objects associated with a first layer or a second layer. If the process determines to render objects associated with the first layer, the process proceeds to block 1013. If the process determines to render objects associated with the second layer the process proceeds to block 1019. In most embodiments the process alternates between rendering of objects associated with the first layer and rendering of objects associated with the second layer. In various embodiments the process may maintain a flag, a register setting, or an index indicating whether to render objects of the first layer or the second layer, or which layer was last rendered.

[0068] In block 1013 the process renders objects associated with the first layer. The process may perform the rendering of objects associated with the first layer by way of use of a graphics processing unit, which may be a separate chip or portion of a chip configured to process graphic information. In block 1015 the process stores information of the rendered objects associated with the first layer in a first memory, which may be considered a memory A. In block 1017 the process displays information stored in a second memory, which may be denoted as a memory B.

[0069] In block 1025 the process swaps a render index. The purpose of the swapping of the render index is to indicate to the process that the process should thereafter render objects associated with a layer other than the objects associated with the layer just rendered. This may be done by way of an index, but many other ways of doing this may also be performed, for example a flag may be set, a register may be set, separate code sections may be used, or other methods may be used.

[0070] In block 1027 the process determines that the process should exit. If yes, the process thereafter exits. If no, the process returns to block 1011.

[0071] Upon returning to block 1011 the process again determines whether to render the objects associated with layer 1 or render the objects associated with layer 2. Assuming, for the sake of example, that the process had previously rendered objects associated with layer 1, the process proceeds to block 1019. In block 1019 the process renders objects associated with layer 2. In block 1021 the process layers rendered information of objects of layer 1 and the rendered information of objects of layer 2. In some embodiments the layering may be performed by the graphics processing engine. In some embodiments, the rendering may be performed by a 3D render engine and the layering may be performed by a 2D graphics engine. In other embodiments the process may be performed by another processor. In some embodiments the layering of information may be performed as part of performance of operations of block 1023, in which the layered information of objects associated with layer 1 and layer 2 are stored in the second memory, which may be denoted as memory B. For example, in some embodiments information associated with the rendered objects of layer 2 may be first stored in memory B, with the information stored in memory A thereafter but over writing information stored in memory B, thereby effectively occluding the information in layer B or vice versa.

[0072] The process then goes to block 1017 and displays the information stored in memory B, and thereafter continues as previously discussed.

[0073] FIG. 11 illustrates a flow of displaying images on, for example, a hand-held device, in accordance with the process of FIG. 10. As may be seen in FIG. 11, during even frames the process renders objects associated with a first layer 1111 and stores the rendered information in a video memory A 1113. Also during even frames, the process displays on a display 1117 information in a video memory B 1115. The information may be for a music based video game, with the display showing, for example, a note chart including graphical user instructions, a musician, and a background. In some embodiments the musician is an object associated with the first layer, and the remainder of the displayed objects are associated with the second layer. In some embodiments, the note chart, alone or in conjunction with the musician, is associated with the first layer, and the remainder of the displayed objects are associated with the second layer.

[0074] During odd frames, the process renders objects associated with a second layer, and layers, which in some embodiments comprises combines, the information stored in video memory A with the rendered information of objects of the second layer. The process stores the layered information
in video memory B. The process also, during odd frames, displays the information stored in video memory B on the display.

[0075] Thus, in every other frame, the process renders objects associated with different layers. In addition, in alternating frames, the process stores information that had previously been stored in video memory B, or displays information rendered during the frame and information previously stored in video memory A on the display.

[0076] The invention therefore provides an image rendering process for, for example, a handheld video game device. Although the invention has been described with respect to certain embodiments, it should be recognized that the invention may be practiced other than as specifically described, the invention comprising the claims and their insubstantial variations supported by this disclosure.

What is claimed is:

1. A method of providing images for a video game, comprising:
   - associating objects with either a first layer or a second layer;
   - rendering objects associated with the first layer;
   - rendering objects associated with the second layer;
   - displaying the rendered objects associated with the first layer on a display; and
   - displaying the rendered objects associated with the second layer on the display.

2. The method of claim 1, further comprising storing information of the rendered objects associated with the first layer in a first memory.

3. The method of claim 2, further comprising storing information of the rendered objects associated with the second layer in a second memory.

4. The method of claim 3 wherein rendering objects associated with the first layer occurs during a first time period.

5. The method of claim 4 wherein rendering objects associated with the second layer occurs during a second time period.

6. The method of claim 5 wherein the second time period follows the first time period.

7. The method of claim 3 wherein rendering objects associated with the first layer and rendering objects associated with the second layer occurs repetitively during play of the video game.

8. The method of claim 3 further comprising additionally rendering objects associated with the first layer, additionally storing information of the additionally rendered objects associated with the first layer in the first memory, displaying the additionally rendered objects associated with the first layer on the display and displaying the rendered objects associated with the second layer on the display using the information of the rendered objects associated with the second layer stored in the second memory.

9. The method of claim 8 wherein objects associated with the first layer include an object representative of a first musician in a music based video game and objects associated with the second layer include a venue in the music based video game.

10. A method of providing images for a music based video game, comprising:
   - associating a first object with a first display layer, the first object representative of a musician in the music based video game; associating a plurality of background objects with a second display layer, the plurality of background objects representative of a venue in the music based video game; iteratively rendering the first object, storing rendered information of the first object in a first memory, and displaying rendered information of the first object on the display;
   - iteratively rendering the plurality of background objects, storing rendered information of the plurality of background objects in a second memory, and displaying rendered information of the plurality of background objects on the display;
   - with displaying rendered information of the first object on the display utilizing the information stored in the first memory in first alternating time periods and displaying rendered information of the plurality of background objects utilizing the information stored in the second memory in second alternating time periods, the first alternating time periods and the second alternating time periods occurring at different times.

11. The method of claim 10 wherein displaying rendered information of the first object on the display and displaying rendered information of the plurality of background objects on the display occurs during both the first alternating time periods and the second alternating time periods.

12. The method of claim 11 wherein displaying rendered information of the first object on the display does not utilize the information stored in the first memory in the second alternating time periods.

13. The method of claim 12 wherein displaying rendered information of the plurality of background objects on the display does not utilize the information stored in the second memory in the first alternating time periods.

14. A handheld game system, comprising:
   - memory storing scene data, the scene data including first scene data and second scene data;
   - a processor configured to render the first scene data and the second scene data;
   - first video memory, coupled to the processor, configured to store rendered first scene data;
   - second video memory, coupled to the processor, configured to store rendered second scene data; and
   - a display coupled to the first video memory and the second video memory;
   - the processor being further configured to alternately:
     a) render the first scene data, command display on the display of the rendered first scene data, command display on the display of the rendered second scene data in the second video memory, and command storage of the rendered first scene data in the first video memory, and
     b) render the second scene data, command display on the display of the rendered second scene data, command display on the display of the rendered first scene data in the first video memory, and command storage of the rendered second scene data in the second video memory.

15. The handheld game system of claim 14, wherein the memory additionally stores layer information, the layer information identifying which of the scene data is first scene data and which of the scene data is second scene data.

16. The handheld game system of claim 14, wherein the first scene data includes information of a representation of an individual and a musical instrument.
17. The handheld game system of claim 16, wherein the second scene data includes information of a representation of a musical venue.

18. A method of providing images for a video game, comprising: associating different objects with different layers; rendering objects associated with a first layer of the different layers; storing information of rendered objects associated with the first layer in a first memory; rendering objects associated with a second layer of the different layers; combining the information of rendered objects associated with the first layer in the first memory with information of rendered objects associated with the second layer; and displaying the combined information.

19. The method of claim 18, further comprising storing at least some of the information of rendered objects associated with the second layer in a second memory.

20. The method of claim 19, wherein combining the information of rendered objects associated with the first layer in the first memory with information of rendered objects associated with the second layer comprises storing at least some of the information of the rendered objects associated with the first layer in the second memory.

21. The method of claim 20 wherein displaying the combined information comprises displaying the combined information stored in the second memory.

22. The method of claim 18, further comprising storing the combined information in a second memory.

* * * * *