

store the commanded effect. The host therefore marks the commanded effect as “waiting” and monitors the device memory to determine if the commanded effect can be later loaded to the device and played. The host internally maintains the duration of the ConstantForce1 effect so as to output the correct force magnitude, direction, etc. at the point in time when the waiting effect can be actually loaded to the device.

[0142] **FIG. 10** is a diagrammatic illustration of an alternate “playlist” embodiment of the device memory of device **11**. In the above embodiments as shown in steps 362-378 of **FIG. 6**, the device **11** examined each effect slot in order and checked whether each effect was tagged (playing); if the effect were tagged, a force based on that effect was added to a total force that was output on the user manipulatable object **12**. **FIG. 10** illustrates an alternate method in which a playlist **500** is stored in device memory. An effect block **502** is stored on the device and host as explained above (a parameter block (not shown) may also be stored). When an effect is tagged by process **310** (e.g. in step **328** of **FIG. 5**), a pointer to that effect or index into the effect block is stored in the next available slot in the playlist **500**. Thus, only the topmost slots of the playlist are preferably filled, with any open slots at the bottom of the list. The total number of tagged effects is stored as a number in a memory location **504**, and is updated whenever an effect is tagged or untagged. In most implementations, the number of slots in the playlist **500** can be less than the number of effect slots implemented in the effect block **502**, since the number of playing effects is likely to be smaller than the total number of effects stored on the device. For example, the device may be able to store 30 effects, but the playlist might only require 10 slots.

[0143] When an effect finishes or is stopped by a command, the effect is removed from the playlist. If there are other effects still playing which are located further down in the list past the removed effect, then one or more of these later effects can be moved to maintain a continuous playlist without gaps. For example, the last effect in the playlist can be moved to the location at which the removed effect used to be stored. In addition, after the effect is removed from the playlist, the total number of effects in location **504** is decremented.

[0144] The efficiency of the playlist **500** is demonstrated when the playing process **350** of **FIG. 6** examines the device memory to determine which effects are to be output as forces. Instead of sequentially examining each slot in the effect block **502** as described in **FIG. 6**, the process instead simply examines the memory location **504** for the number of effects currently tagged (playing). Once this number T is known, the process then looks at the top T entries in the playlist **500** to determine which particular effects are playing, and calculates forces for those effects. This is much more efficient than examining the tag field for each entry in the effect block **502**, especially when there are many effects in the effect block **502**. Furthermore, if no effects or only a small number of effects are playing, no processing time is wasted checking each slot in the effects table.

[0145] While this invention has been described in terms of several preferred embodiments, it is contemplated that alterations, permutations and equivalents thereof will become apparent to those skilled in the art upon a reading of the

specification and study of the drawings. Also, the various features of the embodiments herein can be combined in various ways to provide additional embodiments of the present invention. Furthermore, certain terminology has been used for the purposes of descriptive clarity, and not to limit the present invention. It is therefore intended that the following appended claims include all such alterations, permutations, and equivalents as fall within the true spirit and scope of the present invention.

What is claimed is:

1. A method for providing force effects with a force feedback device having local control of the output of force sensations, said force feedback device coupled to a host computer, the method comprising:

creating a representation of device memory, said device memory being provided on said force feedback device, said representation being allocated in memory of said host computer, wherein an application program is running on said host computer;

receiving a force effect load command from said application program, said force effect load command instructing that data for a force effect be stored in said device memory;

determining whether said device memory can store said force effect by examining said representation of device memory; and

if said device memory can store said force effect, sending said data for said force effect to said force feedback device to be stored in said device memory, wherein said force feedback device uses said data to control a force output to a user of said force feedback device.

2. A method as recited in claim 1 wherein if said device memory can store said force effect, further comprising storing said data for said force effect in a location of said representation of device memory.

3. A method as recited in claim 1 wherein if said device memory is full and cannot store said force effect, further comprising delaying the sending of said force effect to said force feedback device.

4. A method as recited in claim 1 wherein said data for said force effect includes at least one parameter characterizing said force effect.

5. A method as recited in claim 1 further comprising:

receiving a force effect play command from said application program, said force effect play command instructing that said loaded force effect be output to said user by said device; and

sending said force effect play command to said force feedback device, wherein said force feedback device outputs said force based on said loaded force effect.

6. A method as recited in claim 5 further comprising:

receiving a force effect stop command from said application program, said force effect stop command instructing that said force effect stop being output by said device to said user;

sending said force effect stop command to said force feedback device, wherein said force feedback device stops output of said force corresponding to said stopped force effect.