

---

Subject: Re: Tools or No Tools ?

Posted by [Oz](#) on Mon, 06 Oct 2014 19:26:07 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Like most of you, I enjoy FaceBook for its capability to let people present on a daily basis how they interact with the Apple II world (retro bright session, Ebay auctions, pictures, video, misc news) but it is the worst place to search for something which is 1 month old. Because Google do not index FaceBook content, everything disappear very quickly. It is impossible to use such space for References or long term work. A forum like this one is perfect to focus on one specific direction (IIgs programming). What we are saying here will still be valid in 10 years.

Quote:someone was showing The Programmer's Introduction to the Apple IIgs. Then someone said most of what was in there is obsolete. Specifically, Sheppy (IIRC) said that new GS software should be written using TaskMaster which was not covered by that book.

Sheppy is right when he speaks about how TaskMaster has changed the way the Apple IIgs programs have to be written. On the early ages of the Apple IIgs (Prodos 16 was the first 16 bit Os delivered with the IIgs), the Apple IIgs programming was very close to the Macintosh programming. Thanks to its small market, the Apple IIgs system developers had the flexibility to add features that the Mac was missing : The TaskMaster, a new way to deal with Graphic Interface events (mouse click, keyboard key press...). The Apple IIgs programming was easier and more elegant than the Macintosh one.

Everything is true, but only if you consider writing applications using the Apple Graphic User Interface (Apple Menu bar, pull down menu, window with check box, radio button, Text Edit, QuickDraw II...). Everything that make an Apple IIgs program looks the same than another Apple IIgs program. For video games, excepts few of them like Balance of Power, ChessMaster, Full Metal Planete, we don't use the Apple Interface but we take control of the whole screen, remove mouse cursor and Menu Bar and handle all the events ourselves.

QuickDraw II routines are tailor made for Windows and 'serious' applications, not suited for action games where sprites have to be drawn very quickly on the screen. The only events we have to handle when we write a game is the keyboard (Dark Castle), the joystick (Rastan) or the mouse (Zany golf) for controlling the character on the screen. Reading the right memory locations to find out which key has been pressed, what is the direction of the Joystick and where is the mouse is easy and can be done directly in assembler without having to involve the TaskMaster.

We will publish here the low levels routines to read such events. Unlike TaskMaster, in low level assembly language we choose to see if something has occurred (keyboard key, joystick direction, mouse status) instead of being notified that something has happen. If you don't go to read the mouse position, nothing will happen if the user try to click or move the mouse. On the TaskMaster, the events are kept by the system and deliver (in the right order) to the process if required.

As conclusion, we won't start the TaskMaster in arcade video game because we don't need it.

Olivier

---