Subject: Unpack a Picture using the PackBytes compression format
Posted by Oz on Sat, 11 Oct 2014 15:21:38 GMT
View Forum Message <> Reply to Message

Games are using a lot of graphic resources, most of the time stored as individual 320*200 pictures (using 1 or 16 palettes). Even if Apple has defined a standard named Apple Preferred Format (aka APF), most of the Apple IIgs Paint programs have the capability to save the pictures using the PackBytes format (PAK). The compression efficiency is about the same for both formats. The PackBytes format doesn't have the flexibility of the APF (where pictures can have any size) but it has the advantage to have its unpacking routine in the Rom of the Apple IIgs. We access such routine calling the UnPackBytes function from the Miscellaneous Tool Set.

One bank (BankLoad) has been allocated and the compressed graphic file (PAK) has been loaded at $0000. We unpack the data in the same memory bank, at location $8000. The Uncompressed picture will be 32 KB long. The only parameter sent to the UnpackPicture function is the size (in bytes) of the PAK file (=size of packed data), which is <= 32 KB.

```
*------------ Unpack a PackBytes Picture -------------------
UnpackPicture   STA       UP_PackedSize     ; A = Size of Packed Data
            LDA       #$8000          ; Size of output Data Buffer
            STA       UP_UnPackedSize
            LDA       BankLoad         ; XX/00 = Bank used for Loading ($0000-$7FFF) and
Decompression ($8000-$FFFF)
            STA       UP_Packed+1      ; Packed Data Address
            ORA       #$0080
            STZ       UP_UnPacked      ; Reset because these values are modified by the
_UnPackBytes call
            STZ       UP_UnPacked+2
            STA       UP_UnPacked+1     ; Unpacked Data buffer Address
*---
            PushWord   #0              ; Space for Result : Number of bytes unpacked
            PushLong   UP_Packed        ; Pointer to buffer containing the packed data
            PushWord   UP_PackedSize     ; Size of the Packed Data
            PushLong   #UP_UnPacked      ; Pointer to Pointer to unpacked buffer
            PushLong   #UP_UnPackedSize  ; Pointer to a Word containing size of unpacked data
            _UnPackBytes
            PLA                     ; Number of byte unpacked
*---
            RTS
*-----------
UP_Packed       HEX       00000000         ; Address of Packed Data
UP_PackedSize   HEX       0000            ; Size of Packed Data
UP_UnPacked     HEX       00000000         ; Address of Unpacked Data Buffer (modified by
_UnPackBytes)
UP_UnPackedSize HEX       0000            ; Size of Unpacked Data Buffer (modified by
_UnPackBytes)
*-------------------------------------------------------
```

We can chain the LoadFile and the UnpackPicture function to end up with a LoadPicture function :

```
*----------- Load and Unpack a PackBytes Picture --------------------
LoadPicture     JSR       LoadFile          ; X = Picture File Path, A = XX/00, Bank used for Loading
                BCS       LP_End
                JSR       UnpackPicture     ; A = Packed Data Size
                CLC                         ; no error
LP_End          RTS
*-------------------------------------------------------------
```

Olivier
