

T.P. 10 – Corrigé

Space Invaders (partie 3)

Étape 1

```
Main      ; Fait pointer A0 sur la matrice de points de l'envahisseur.
          lea    Invader_Bitmap,a0

          ; Fait pointer A1 sur la mémoire vidéo.
          lea    VIDEO_START,a1

          ; D7.W = Compteur de boucles
          ;       = Nombre d'itérations - 1 (car DBRA)
          ; Nombre d'itérations = Nombre de lignes
          move.w #16-1,d7

\loop      ; Affiche une ligne de l'envahisseur.
          ; (22 pixels nécessitent 3 octets.)
          move.b (a0)+,(a1)
          move.b (a0)+,1(a1)
          move.b (a0)+,2(a1)

          ; Passe à l'adresse vidéo de la ligne suivante.
          adda.l #BYTE_PER_LINE,a1

          ; Reboucle tant qu'il reste des lignes à afficher.
          dbra   d7,\loop

          illegal
```

Étape 2

```

PixelToByte      ; Taille en pixels + 7 -> D3.W
                 addq.w #7,d3

                 ; D3.W/8 -> D3.W
                 lsr.w #3,d3

                 ; Sortie du sous-programme.
                 rts

```

```

CopyLine          ; Sauvegarde les registres dans la pile.
                 movem.l d3/a1,-(a7)

                 ; Nombre d'itérations = Largeur en octets
                 ; Nombre d'itérations - 1 (car DBRA) -> D3.W
                 subq.w #1,d3

\loop             ; Copie tous les octets de la ligne.
                 move.b (a0)+,(a1)+
                 dbra    d3,\loop

                 ; Restaure les registres puis sortie.
                 movem.l (a7)+,d3/a1
                 rts

```

```

CopyBitmap        ; Sauvegarde les registres dans la pile.
                 movem.l d3/d4/a0/a1,-(a7)

                 ; Largeur en octets -> D3.W
                 move.w WIDTH(a0),d3
                 jsr     PixelToByte

                 ; Nombre d'itérations - 1 (car DBRA) -> D4.W
                 ; Nombre d'itérations = Hauteur en pixels
                 move.w HEIGHT(a0),d4
                 subq.w #1,d4

                 ; Adresse de la matrice de points -> A0.L
                 lea     MATRIX(a0),a0

\loop             ; Copie une ligne de la matrice.
                 jsr     CopyLine

                 ; Passe à l'adresse vidéo de ligne suivante.
                 adda.l #BYTE_PER_LINE,a1

                 ; Reboucle tant qu'il y a des lignes à afficher.
                 dbra    d4,\loop

                 ; Restaure les registres puis sortie.
                 movem.l (a7)+,d3/d4/a0/a1
                 rts

```

Étape 3

```

; =====
; Données
; =====

InvaderA_Bitmap  dc.w    24,16
                  dc.b    %00000000,%11111111,%00000000
                  dc.b    %00000000,%11111111,%00000000
                  dc.b    %00111111,%11111111,%11111100
                  dc.b    %00111111,%11111111,%11111100
                  dc.b    %11111111,%11111111,%11111111
                  dc.b    %11111111,%11111111,%11111111
                  dc.b    %11111100,%00111100,%00111111
                  dc.b    %11111100,%00111100,%00111111
                  dc.b    %11111111,%11111111,%11111111
                  dc.b    %11111111,%11111111,%11111111
                  dc.b    %00000011,%11000011,%11000000
                  dc.b    %00000011,%11000011,%11000000
                  dc.b    %00001111,%00111100,%11110000
                  dc.b    %00001111,%00111100,%11110000
                  dc.b    %11110000,%00000000,%00001111
                  dc.b    %11110000,%00000000,%00001111

InvaderB_Bitmap  dc.w    22,16
                  ; ...
                  ; ...

InvaderC_Bitmap  dc.w    16,16
                  dc.b    %00000011,%11000000
                  dc.b    %00000011,%11000000
                  dc.b    %00001111,%11110000
                  dc.b    %00001111,%11110000
                  dc.b    %00111111,%11111100
                  dc.b    %00111111,%11111100
                  dc.b    %11110011,%11001111
                  dc.b    %11110011,%11001111
                  dc.b    %11111111,%11111111
                  dc.b    %11111111,%11111111
                  dc.b    %00110011,%11001100
                  dc.b    %00110011,%11001100
                  dc.b    %11000000,%00000011
                  dc.b    %11000000,%00000011
                  dc.b    %00110000,%00001100
                  dc.b    %00110000,%00001100

Ship_Bitmap      dc.w    24,14
                  dc.b    %00000000,%00011000,%00000000
                  dc.b    %00000000,%00011000,%00000000
                  dc.b    %00000000,%01111110,%00000000
                  dc.b    %00000000,%01111110,%00000000
                  dc.b    %00000000,%01111110,%00000000
                  dc.b    %00000000,%01111110,%00000000
                  dc.b    %00111111,%11111111,%11111100
                  dc.b    %00111111,%11111111,%11111100
                  dc.b    %11111111,%11111111,%11111111
                  dc.b    %11111111,%11111111,%11111111
                  dc.b    %11111111,%11111111,%11111111
                  dc.b    %11111111,%11111111,%11111111
                  dc.b    %11111111,%11111111,%11111111
                  dc.b    %11111111,%11111111,%11111111

```