

T.P. 8 – Corrigé

Space Invaders (partie 1)

Étape 1

```

FillScreen      ; Sauvegarde les registres dans la pile.
                 movem.l d7/a0,-(a7)

                 ; Fait pointer A0 sur la mémoire vidéo.
                 ; Cette instruction est identique à : movea.l #VIDEO_START,a0
                 lea     VIDEO_START,a0

                 ; Initialisation du compteur de boucle (D7.W).
                 ; La copie se fera sur 32 bits, c'est-à-dire sur 4 octets.
                 ; Le nombre d'itérations est donc la taille en octets divisée par 4.
                 ; Le test de sortie se fera à l'aide de DBRA,
                 ; donc D7.W doit contenir le nombre d'itérations moins 1 (cf. cours).
                 move.w  #VIDEO_SIZE/4-1,d7

\loop           ; Copie la donnée dans la mémoire vidéo
                 ; et passe à l'adresse suivante.
                 move.l  d0,(a0)+
                 dbra    d7,\loop

                 ; Restaure les registres puis sortie.
                 movem.l (a7)+,d7/a0
                 rts

```

Étape 2

```

HLines      ; Sauvegarde les registres dans la pile.
             movem.l d0/d6/d7/a0,-(a7)

             ; Fait pointer A0 sur la mémoire vidéo.
             lea     VIDEO_START,a0

             ; 32 pixels blancs dans D0.
             move.l  #$ffffffff,d0

             ; D7.W = Compteur de boucle
             ;       = Nombre d'itérations - 1 (car DBRA).
             ; -----
             ; Nombre d'itérations = Nombre de rayures (blanches + noires)
             ; Hauteur d'un rayure = 8 pixels
             ; Nombre de rayures   = Hauteur de la fenêtre / 8
             move.l  #VIDEO_HEIGHT/8-1,d7

\loop        ; Dessine les rayures (avec les pixels contenus dans D0).
             ; -----
             ; D6.W = Compteur de boucles
             ;       = Nombre d'itérations - 1 (car DBRA)
             ; -----
             ; Nombre d'itérations = Nombre de mots longs
             ; Nombre de mots longs = Nombre d'octets / 4
             ; Nombre d'octets     = BYTE_PER_LINE x Hauteur d'une rayure
             ; Hauteur d'une rayure = 8 pixels
             move.w  #BYTE_PER_LINE*8/4-1,d6
\stripe_loop move.l  d0,(a0)+
             dbra    d6,\stripe_loop

             ; Inverse la couleur des pixels.
             not.l   d0

             ; Reboucle tant qu'il reste des rayures à dessiner.
             dbra    d7,\loop

             ; Restaure les registres puis sortie.
             movem.l (a7)+,d0/d6/d7/a0
             rts

```