

# T.P. 9 – Corrigé

## Space Invaders (partie 12)

### Étape 1

```

InitInvaderShots    ; Sauvegarde les registres.
                    movem.l d7/a0,-(a7)

                    ; Adresse des tirs -> A0.L
                    lea     InvaderShots,a0

                    ; Nombre d'itérations - 1 (car DBRA) -> D7.W
                    move.w  #INVADER_SHOT_MAX-1,d7

\loop               ; Initialise l'état et les bitmaps.
                    move.w  #HIDE,STATE(a0)
                    move.l  #InvaderShot1_Bitmap,BITMAP1(a0)
                    move.l  #InvaderShot2_Bitmap,BITMAP2(a0)

                    ; Passe au tir suivant.
                    adda.l  #SIZE_OF_SPRITE,a0
                    dbra    d7,\loop

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

```

### Étape 2

```

GetHiddenShot       ; Sauvegarde les registres.
                    move.l  d7,-(a7)

                    ; Adresse des tirs -> A0.L
                    lea     InvaderShots,a0

                    ; Nombre d'itérations - 1 (car DBRA) -> D7.W
                    move.w  #INVADER_SHOT_MAX-1,d7

\loop               ; Si le tir n'est pas visible, renvoie true.
                    ; (L'adresse du tir se trouve dans A0.L.)
                    cmp.w  #HIDE,STATE(a0)
                    beq     \true

                    ; Passe au tir suivant.
                    adda.l  #SIZE_OF_SPRITE,a0
                    dbra    d7,\loop

\false              ; Renvoie false (pas de tir disponible).
                    move.l  (a7)+,d7
                    andi.b  #%11111011,ccr
                    rts

\true               ; Renvoie true (un tir disponible a été trouvé).
                    move.l  (a7)+,d7
                    ori.b   #%00000100,ccr
                    rts

```

**Étape 3**

```

ConnectInvaderShot ; Sauvegarde les registres.
                   movem.l d1/d2/d3/a0/a1,-(a7)

                   ; Si l'envahisseur n'est pas visible, on quitte.
                   cmpi.w #HIDE,STATE(a1)
                   beq     \quit

                   ; Récupère l'adresse d'un tir disponible.
                   ; Si aucun tir disponible, on ne fait rien.
                   jsr     GetHiddenShot
                   bne     \quit

                   ; Place le tir au même endroit que l'envahisseur.
                   move.w X(a1),X(a0)
                   move.w Y(a1),Y(a0)

                   ; Détermine la largeur et la hauteur de l'envahisseur.
                   movea.l BITMAP1(a1),a1
                   move.w WIDTH(a1),d1
                   move.w HEIGHT(a1),d2

                   ; Détermine la largeur du tir.
                   movea.l BITMAP1(a0),a1
                   move.w WIDTH(a1),d3

                   ; (Largeur Envahisseur - Largeur Tir) / 2 -> D1.W
                   sub.w d3,d1
                   lsr.w #1,d1

                   ; On centre le tir sur les abscisses.
                   ; On descend le tir juste au dessous de l'envahisseur.
                   add.w d1,X(a0)
                   add.w d2,Y(a0)

                   ; Le tir est rendu visible.
                   move.w #SHOW,STATE(a0)

\quit              ; Restaure les registres puis sortie.
                   movem.l (a7)+,d1/d2/d3/a0/a1
                   rts

```