

T.P. 7 – Corrigé

Space Invaders (partie 10)

Étape 1

```

GetInvaderStep    ; Sauvegarde les registres.
                  move.l  d0,-(a7)

                  ; Nouvelle abscisse globale -> D0.W
                  move.w  InvaderX,d0
                  add.w   InvaderCurrentStep,d0

                  ; Si l'abscisse globale est trop petite,
                  ; les envahisseurs ont atteint le bord gauche.
                  ; Il faut donc changer de direction.
                  cmpi.w  #INVADER_X_MIN,d0
                  blt     \change

                  ; Si l'abscisse globale est trop grande,
                  ; les envahisseurs ont atteint le bord droit.
                  ; Il faut donc changer de direction.
                  cmpi.w  #INVADER_X_MAX,d0
                  bgt     \change

\noChange         ; Pas de changement de direction.
                  ; Mouvement relatif -> D1.W et D2.W.
                  ; L'abscisse globale est mise à jour.
                  move.w  InvaderCurrentStep,d1
                  clr.w   d2
                  move.w  d0,InvaderX
                  bra     \quit

\change           ; Changement de direction.
                  ; Mouvement relatif -> D1.W et D2.W.
                  ; L'ordonnée globale est mise à jour.
                  ; Le signe du pas est inversé.
                  clr.w  d1
                  move.w  #INVADER_STEP_Y,d2
                  add.w  d2,InvaderY
                  neg.w  InvaderCurrentStep

\quit             ; Restaure les registres puis sortie.
                  move.l  (a7)+,d0
                  rts

```

Étape 2

```

MoveAllInvaders    ; Sauvegarde les registres.
                   movem.l d1/d2/a1/d7,-(a7)

                   ; Récupère les déplacements relatifs dans D1.W et D2.W.
                   ; (La position globale est mise à jour.)
                   jsr      GetInvaderStep

                   ; Fait pointer A1.L sur le premier envahisseur.
                   lea      Invaders,a1

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

\loop              ; Si l'envahisseur n'est pas affiché, on passe au suivant.
                   cmp.w  #HIDE,STATE(a1)
                   beq      \continue

                   ; Déplace l'envahisseur.
                   jsr      MoveSprite

\continue         ; Pointe sur le prochain envahisseur.
                   adda.l  #SIZE_OF_SPRITE,a1

                   ; On reboucle tant qu'il reste des envahisseurs.
                   dbra    d7,\loop

\quit             ; Restaure les registres puis sortie.
                   movem.l (a7)+,d1/d2/a1/d7
                   rts

```

Étape 3

```

MoveInvaders      ; Décrémente la variable "skip",
                   ; et ne fait rien si elle n'est pas nulle.
                   subq.w  #1,\skip
                   bne      \quit

                   ; Réinitialise "skip" à sa valeur maximale.
                   move.w  #8,\skip

                   ; Appel de MoveAllInvaders.
                   jsr      MoveAllInvaders

\quit             ; Sortie du sous programme.
                   rts

\skip             ; Compteur d'affichage des envahisseurs
                   dc.w    1

```