

T.P. 10 – Corrigé

Space Invaders (partie 13)

Étape 1

```

NewInvaderShot    ; Sauvegarde les registres.
                  movem.l d0/d1/a1,-(a7)

                  ; Récupère un nombre aléatoire.
                  jsr      Random

                  ; Si ce nombre est supérieur ou égale
                  ; au nombre d'envahisseurs, on ne fait rien.
                  cmp.w    #INVADER_COUNT,d0
                  bhs      \quit

                  ; Détermine l'adresse de l'envahisseur.
                  mulu.w   #SIZE_OF_SPRITE,d0
                  lea      Invaders,a1
                  adda.l   d0,a1

                  ; Connecte un tir à cet envahisseur.
                  jsr      ConnectInvaderShot

\quit             ; Restaure les registres puis sortie.
                  movem.l (a7)+,a1/d0/d1
                  rts

```

Étape 2

```

PrintInvaderShots ; Sauvegarde les registres.
                  movem.l d7/a1,-(a7)

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

                  ; Adresse des tirs d'envahisseurs -> A1.L
                  lea      InvaderShots,a1

\loop             ; Affiche un tir d'envahisseur.
                  jsr      PrintSprite

                  ; Passe au prochain tir et reboucle.
                  adda.l   #SIZE_OF_SPRITE,a1
                  dbra     d7,\loop

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

```

Étape 3

```

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

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

                    ; Adresse des tirs d'envahisseurs -> A1.L
                    lea     InvaderShots,a1

\loop                ; Si le tir n'est pas affiché, on ne fait rien.
                    cmp.w  #HIDE,STATE(a1)
                    beq     \continue

                    ; Déplace un tir.
                    clr.w   d1
                    move.w  #INVADER_SHOT_STEP,d2
                    jsr     MoveSprite
                    beq     \continue

\outOfScreen         ; Le tir sort de l'écran (on le rend invisible).
                    move.w  #HIDE,STATE(a1)

\continue            ; Passe au prochain tir et reboucle.
                    adda.l  #SIZE_OF_SPRITE,a1
                    dbra     d7,\loop

                    ; Échange les bitmaps.
                    jsr     SwapInvaderShots

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

```