

T.P. 11 – Corrigé

Space Invaders (partie 14)

Étape 1

```

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

                   ; Réinitialise la variable \skip.
                   move.w #6,\skip

                   ; 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                ; Échange les bitmaps 1 et 2 pour tous les tirs.
                   jsr     SwapBitmap
                   adda.l #SIZE_OF_SPRITE,a1
                   dbra    d7,\loop

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

\skip                ; Initialise une variable \skip à 6.
                   dc.w    6

```

Étape 2

```

IsShipHit      ; Sauvegarde les registres.
               movem.l d7/a1/a2,-(a7)

               ; Adresse du vaisseau -> A1.L
               lea    Ship,a1

               ; Adresse des tirs d'envahisseurs -> A2.L
               lea    InvaderShots,a2

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

\loop          ; Si un tir entre en collision avec le vaisseau,
               ; on renvoie true.
               jsr    IsSpriteColliding
               beq    \true

               ; Passe au tir suivant.
               adda.l #SIZE_OF_SPRITE,a2
               dbra   d7,\loop

\false         ; Renvoie false (aucune collision).
               andi.b #%11111011,ccr
               bra    \quit

\true          ; Renvoie true (collision).
               ori.b  #%00000100,ccr
\quit         movem.l (a7)+,d7/a1/a2
               rts

```

Étape 3

```
IsShipColliding    ; Sauvegarde les registres.
                   movem.l d7/a1/a2,-(a7)

                   ; Adresse du vaisseau -> A1.L
                   lea    Ship,a1

                   ; Adresse des envahisseurs -> A2.L
                   lea    Invaders,a2

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

\loop              ; Si un envahisseur entre en collision avec le vaisseau,
                   ; on renvoie true.
                   jsr    IsSpriteColliding
                   beq    \true

                   ; Passe à l'envahisseur suivant.
                   adda.l #SIZE_OF_SPRITE,a2
                   dbra   d7,\loop

\false            ; Renvoie false (aucune collision).
                   andi.b #%11111011,CCR
                   bra    \quit

\true             ; Renvoie true (collision).
                   ori.b  #%00000100,CCR
\quit            movem.l (a7)+,d7/a1/a2
                   rts
```

Étape 4

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

                  ; Adresse des envahisseurs -> A0.L
                  lea    Invaders,a0

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

\loop              ; Si l'envahisseur n'est pas affiché,
                  ; on passe à l'envahisseur suivant.
                  cmp.w  #HIDE,STATE(a0)
                  beq    \next

                  ; If the invader is too low, return true.
                  cmpi.w #280,Y(a0)
                  bhi    \true

\next              ; Passe à l'envahisseur suivant.
                  adda.l #SIZE_OF_SPRITE,a0
                  dbra  d7,\loop

\false             ; Renvoie false.
                  andi.b #%11111011,CCR
                  bra    \quit

\true              ; Renvoie true.
                  ori.b  #00000100,CCR
\quit              movem.l (a7)+,d7/a0
                  rts
```