

T.P. 4

Space Invaders (partie 7)

Étape 1

Réalisez le sous-programme **GetRectangle** qui renvoie les coordonnées du rectangle qui délimite un sprite. La taille du rectangle prise en compte sera celle du bitmap 1 du sprite (largeur et hauteur en pixels).

Entrée : **A0.L** = Adresse d'un sprite.

Sorties : **D1.W** = Abscisse du point supérieur gauche du sprite.

D2.W = Ordonnée du point supérieur gauche du sprite.

D3.W = Abscisse du point inférieur droit du sprite.

D4.W = Ordonnée du point inférieur droit du sprite.

Étape 2

Réalisez le sous-programme **IsSpriteColliding** qui détermine si deux sprites entrent en collision.

Entrées : **A1.L** = Adresse du premier sprite.

A2.L = Adresse du second sprite.

Sorties : **Z** renvoie *false* (0) si les deux sprites n'entrent pas en collision.

Z renvoie *true* (1) si les deux sprites entrent en collision.

Indications :

- Nous considérerons que deux sprites entrent en collision si leurs rectangles se chevauchent (il sera inutile de tester si les pixels d'un sprite sont blancs ou noirs).
- Nous considérerons que si la propriété STATE d'au moins un sprite possède la valeur HIDE, alors il n'y aura aucune collision, quelle que soit la position des sprites.
- Aidez-vous du sous-programme **GetRectangle** afin de récupérer les coordonnées des sprites et effectuer des comparaisons sur ces différentes coordonnées afin de déterminer si les bitmaps se chevauchent.

Utilisez le programme principal et les données ci-dessous afin de tester votre sous-programme. Ce programme principal affiche deux sprites à l'écran : un sprite mobile qu'il est possible de déplacer à l'aide du clavier et un sprite immobile au centre de l'écran. Vous pourrez déplacer le sprite mobile afin de le faire entrer en collision avec le sprite central. Si votre sous-programme fonctionne correctement, un petit carré blanc s'affichera dans le coin supérieur gauche de l'écran au moment de la collision.

```

; =====
; Programme principal
; =====

org      $500

Main     lea      FixedSprite,a2
\loop    movea.l  a2,a1
          jsr      PrintSprite

          lea      MovingSprite,a1
          jsr      PrintSprite

          jsr      IsSpriteColliding
          bne      \notColliding

\colliding lea      VIDEO_BUFFER,a0
          move.w   #15,d7
\square   move.w   #$ffff,(a0)
          adda.l   #BYTE_PER_LINE,a0
          dbra     d7,\square

\notColliding jsr      BufferToScreen
          jsr      MoveSpriteKeyboard
          bra      \loop

; ...

; =====
; Données
; =====

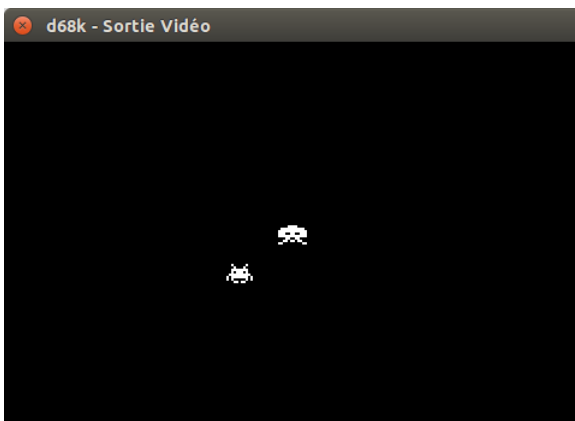
; Sprites
; -----

MovingSprite dc.w    SHOW
             dc.w    0,152
             dc.l    InvaderB_Bitmap
             dc.l    0

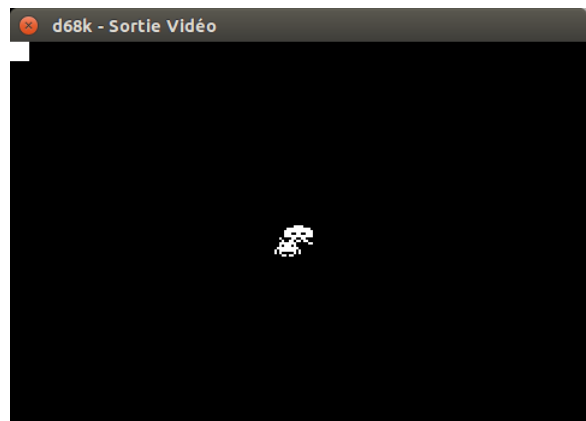
FixedSprite  dc.w    SHOW
             dc.w    228,152
             dc.l    InvaderA_Bitmap
             dc.l    0

```

Pas de collision



Collision (carré blanc en haut à gauche)



Étape 3

Réalisez le même test que l'étape précédente, mais le bitmap 1 du sprite mobile sera cette fois le tir du vaisseau. La matrice de point vous est donnée ci-dessous.

| | | |
|-----------------|------|-----------|
| ShipShot_Bitmap | dc.w | 2,6 |
| | dc.b | %11000000 |
| | dc.b | %11000000 |
| | dc.b | %11000000 |
| | dc.b | %11000000 |
| | dc.b | %11000000 |
| | dc.b | %11000000 |