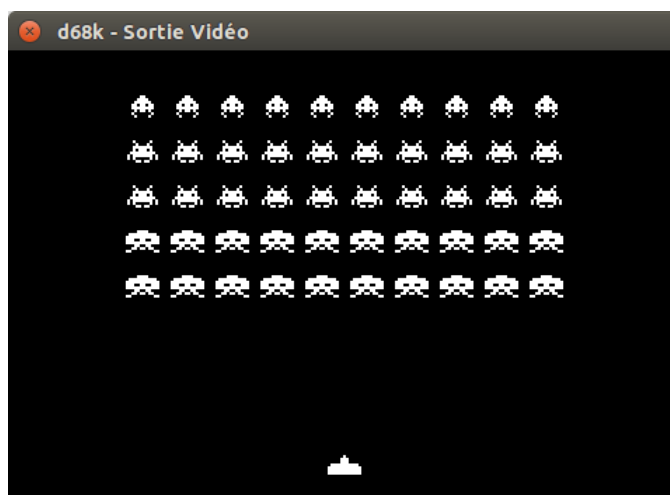


T.P. 6

Space Invaders (partie 9)

Étape 1

Il nous faut maintenant créer tous les sprites d'envahisseurs. Il y aura en tout 50 envahisseurs répartis sur 5 lignes. Nous aurons donc 10 envahisseurs par ligne. Voici un aperçu de ce que nous souhaitons obtenir :



Il serait possible d'initialiser ces 50 sprites statiquement, mais cela prendrait un certain temps. Il faudrait en effet créer les structures de 50 sprites, puis remplir manuellement leurs différents champs, calculer les positions, etc. Ceci serait bien trop long. Nous allons donc réserver un espace mémoire suffisant afin de stocker les 50 sprites et nous les initialiserons dynamiquement.

Pour commencer, il nous faut définir quelques constantes qui serviront à manipuler les sprites d'envahisseurs.

La première constante à ajouter est la taille d'un sprite en octets (`SIZE_OF_SPRITE`). Nous pouvons ajouter cette constante dans la partie *Sprite* de nos constantes.

<i>; Sprites</i>			
<i>; -----</i>			
STATE	equ	0	<i>; État de l'affichage</i>
X	equ	2	<i>; Abscisse</i>
Y	equ	4	<i>; Ordonnée</i>
BITMAP1	equ	6	<i>; Bitmap no 1</i>
BITMAP2	equ	10	<i>; Bitmap no 2</i>
HIDE	equ	0	<i>; Afficher le sprite</i>
SHOW	equ	1	<i>; Ne pas afficher le sprite</i>
SIZE_OF_SPRITE	equ	14	<i>; Taille d'un sprite en octets</i>

Ajoutons ensuite quelques constantes sur le nombre d'envahisseurs :

```

; Envahisseurs
; -----
INVADER_PER_LINE    equ    10
INVADER_PER_COLUMN  equ    5
INVADER_COUNT       equ    INVADER_PER_LINE*INVADER_PER_COLUMN

```

Nous pouvons maintenant réserver un espace mémoire qui pourra contenir tous les sprites d'envahisseurs. Nous utiliserons pour cela la directive d'assemblage DS (*cf.* cours). La taille de cet espace mémoire sera égale au nombre d'envahisseurs multiplié par la taille d'un sprite.

Ajoutez la ligne suivante dans la partie *Données* de votre fichier source :

```
Invaders          ds.b    INVADER_COUNT*SIZE_OF_SPRITE
```

Cet espace mémoire sera divisé de la façon suivante :

- L'adresse `Invaders` contiendra la structure du premier envahisseur ;
- L'adresse `Invaders+SIZE_OF_SPRITE` contiendra la structure du deuxième envahisseur ;
- L'adresse `Invaders+2*SIZE_OF_SPRITE` contiendra la structure du troisième envahisseur ;
- L'adresse `Invaders+3*SIZE_OF_SPRITE` contiendra la structure du quatrième envahisseur ;
- Etc.
- L'adresse `Invaders+49*SIZE_OF_SPRITE` contiendra la structure du dernier envahisseur.

Il faut maintenant initialiser cet espace mémoire. Chaque envahisseur sera représenté dans un carré de 32 pixels sur 32 pixels. Un envahisseur sera centré horizontalement dans ce carré. Voyez l'image ci-dessous qui représente en rouge la grille qui contiendra les envahisseurs. Attention, cette grille est purement imaginaire ; elle nous servira à initialiser la position des envahisseurs.



Afin de simplifier cette initialisation, nous allons commencer par initialiser la première ligne d’envahisseur. Pour cela, vous devez réaliser le sous-programme **InitInvaderLine** qui initialise une ligne d’envahisseurs.

Entrées : **D1.W** = Abscisse de la ligne (abscisse du coin supérieur gauche du premier carré).

D2.W = Ordonnée de la ligne (ordonnée du coin supérieur gauche du premier carré).

A0.L = Adresse de la structure du premier envahisseur de la ligne.

A1.L = Adresse du bitmap 1 des envahisseurs.

A2.L = Adresse du bitmap 2 des envahisseurs.

Pour rappel, la structure d’un sprite est la suivante :

Champ	Taille (bits)	Codage	Description
STATE	16	Entier non signé	État d’affichage du sprite. Seulement deux valeurs possibles : HIDE = 0 ou SHOW = 1
X	16	Entier signé	Abscisse du sprite
Y	16	Entier signé	Ordonnée du sprite
BITMAP1	32	Entier non signé	Adresse du premier bitmap
BITMAP2	32	Entier non signé	Adresse du second bitmap

Indications :

- Modifier **D1.W** afin de centrer le sprite sur une largeur de 32 pixels. La largeur du sprite peut être récupérée à partir du bitmap 1.
- Il faudra ensuite pour tous les sprites de la ligne :
 - Positionner le champ STATE à la valeur SHOW ;
 - Positionner le champ X à la même valeur que **D1.W** ;
 - Positionner le champ Y à la même valeur que **D2.W** ;
 - Positionner le champ BITMAP1 à la même valeur que **A1.L** ;
 - Positionner le champ BITMAP2 à la même valeur que **A2.L** ;
- Incrémenter **D1.W** et **A0.L** avec les valeurs correctes de façon à traiter le sprite suivant.

Afin de tester le sous-programme **InitInvaderLine**, il vous faudra réaliser le sous-programme **PrintInvaders** qui affichera tous les envahisseurs présents dans l’espace mémoire **Invaders**. Il suffira pour cela d’appeler le sous-programme **PrintSprite** pour les 50 envahisseurs.

Vous testerez vos sous-programmes à l’aide du programme principal ci-après. Essayez également de bien comprendre le fonctionnement de ce dernier.

Remarques :

- Pour l’instant, le second bitmap n’est pas utilisé. C’est pour cela que le registre **A2** est initialisé à zéro.
- Pour l’instant, seulement 10 sprites sur 50 sont initialisés à l’aide de **InitInvaderLine**. Les sprites inutilisés seront simplement ignorés par le sous-programme **PrintSprite** qui considérera que les états de ces sprites sont positionnés à HIDE.

```

Main      move.w #(VIDEO_WIDTH-(INVADER_PER_LINE*32))/2,d1
          move.w #32,d2
          lea  Invaders,a0
          lea  InvaderC_Bitmap,a1
          lea  0,a2
          jsr  InitInvaderLine

\loop    jsr  PrintShip
          jsr  PrintShipShot
          jsr  PrintInvaders
          jsr  BufferToScreen

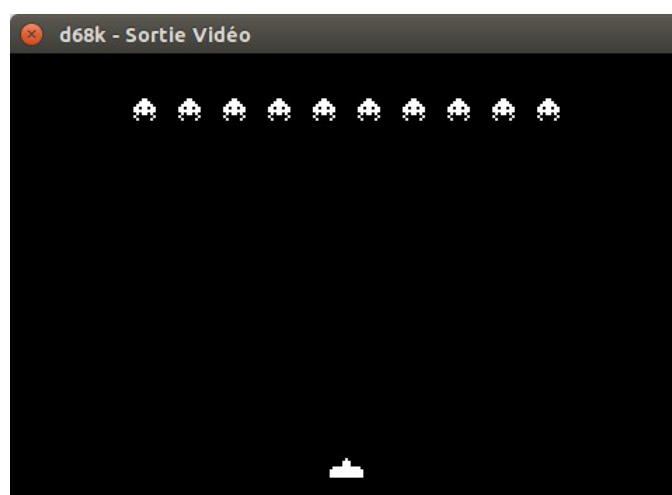
          jsr  MoveShip
          jsr  MoveShipShot

          jsr  NewShipShot

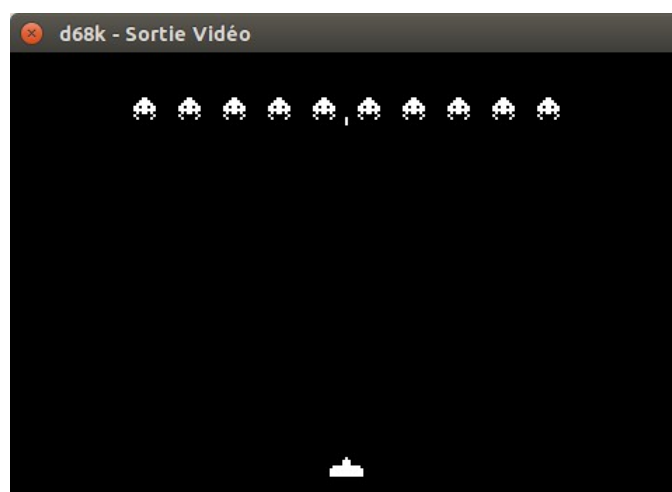
          bra  \loop

```

Capture d'écran du résultat attendu :



Normalement, vous pouvez toujours déplacer le vaisseau et tirer (même si aucun test de collision n'est effectué). Pour voir si vos envahisseurs sont correctement positionnés, tirez dans la position initiale du vaisseau et vérifiez que votre tir passe bien au centre des envahisseurs :



Étape 2

Pour cette étape, nous allons commencer par stocker la position initiale de l'ensemble des envahisseurs directement en mémoire (il s'agit des coordonnées du coin supérieur gauche du tout premier carré). Cela nous permettra de la modifier dynamiquement dans les prochaines étapes. Ajoutez les deux lignes ci-dessous dans la partie *Données* de votre fichier source :

```
InvaderX      dc.w    (VIDEO_WIDTH-(INVADER_PER_LINE*32))/2    ; Abscisse globale
InvaderY      dc.w    32                               ; Ordonnée globale
```

Réalisez maintenant le sous-programme **InitInvaders** qui initialise tous les sprites d'envahisseurs. Vous effectuerez cinq appels à **InitInvaderLine**. Les bitmaps 1 associés aux différents sprites seront les suivants (les bitmaps 2 seront mis à 0) :

```
Ligne 0 : InvaderC_Bitmap ;
Ligne 1 : InvaderB_Bitmap ;
Ligne 2 : InvaderB_Bitmap ;
Ligne 3 : InvaderA_Bitmap ;
Ligne 4 : InvaderA_Bitmap.
```

Vous testerez votre sous-programme à l'aide du programme principal ci-dessous :

```
Main      jsr    InitInvaders
\loop     jsr    PrintShip
          jsr    PrintShipShot
          jsr    PrintInvaders
          jsr    BufferToScreen

          jsr    MoveShip
          jsr    MoveShipShot

          jsr    NewShipShot

          bra   \loop
```

Tirez dans la position initiale du vaisseau et vérifiez que votre tir passe bien au centre des envahisseurs sans les toucher. Vérifiez également que votre vaisseau se déplace toujours correctement.

