

T.P. 10

Space Invaders (partie 13)

Étape 1

Notre programme doit maintenant décider quels envahisseurs doivent tirer et à quel moment. Pour cela, nous procéderons de la manière suivante :

Pour chaque itération de la boucle principale :

- Génération d'un nombre aléatoire compris entre 0 et 2 047.
- Si le nombre aléatoire est supérieur ou égal au nombre total d'envahisseurs (INVADER_COUNT), alors aucun tir n'est connecté à un envahisseur.
- Si le nombre aléatoire est inférieur au nombre total d'envahisseurs, alors on utilise ce nombre comme indice pour déterminer une adresse d'envahisseur (par exemple, l'envahisseur d'indice 0 se trouve à l'adresse `Invaders`, l'envahisseur d'indice 1 à l'adresse `Invaders+SIZE_OF_SPRITE`, etc.), puis si cet envahisseur est visible, on lui connecte un tir disponible (appel à **ConnectInvaderShot**).

Pour générer le nombre aléatoire, vous utiliserez le sous-programme suivant (ne cherchez pas à comprendre son fonctionnement) :

Random	<code>move.l</code>	<code>\old,d0</code>
	<code>muls.w</code>	<code>#16807,d0</code>
	<code>and.l</code>	<code>#\$7fffffff,d0</code>
	<code>move.l</code>	<code>d0,\old</code>
	<code>lsr.l</code>	<code>#4,d0</code>
	<code>and.l</code>	<code>#\$7ff,d0</code>
	<code>rts</code>	
<code>\old</code>	<code>dc.l</code>	<code>425625</code>

Réalisez le sous-programme **NewInvaderShot** qui réalise les opérations décrites ci-dessus. Pour finir, effectuez un appel à ce sous-programme dans la boucle principale de votre fichier source.

Main	<code>jsr</code>	<code>InitInvaders</code>
	<code>jsr</code>	<code>InitInvaderShots</code>
\loop	<code>jsr</code>	<code>PrintShip</code>
	<code>jsr</code>	<code>PrintShipShot</code>
	<code>jsr</code>	<code>PrintInvaders</code>
	<code>jsr</code>	<code>BufferToScreen</code>
	<code>jsr</code>	<code>DestroyInvaders</code>
	<code>jsr</code>	<code>MoveShip</code>
	<code>jsr</code>	<code>MoveInvaders</code>
	<code>jsr</code>	<code>MoveShipShot</code>
	<code>jsr</code>	<code>NewShipShot</code>
	<code>jsr</code>	<code>NewInvaderShots</code>
	<code>jsr</code>	<code>SpeedInvaderUp</code>
	<code>bra</code>	<code>\loop</code>

Étape 2

En vous inspirant du sous-programme **PrintInvaders**, réalisez le sous-programme **PrintInvaderShots** qui affiche tous les tirs d’envahisseurs.

Ajoutez ensuite un appel à ce sous-programme dans votre boucle principale :

```

Main          jsr    InitInvaders
              jsr    InitInvaderShots

\loop         jsr    PrintShip
              jsr    PrintShipShot
              jsr    PrintInvaders
              jsr    PrintInvaderShots
              jsr    BufferToScreen

              jsr    DestroyInvaders

              jsr    MoveShip
              jsr    MoveInvaders
              jsr    MoveShipShot

              jsr    NewShipShot
              jsr    NewInvaderShot

              jsr    SpeedInvaderUp

              bra    \loop

```

Lancez votre programme principal et voyez si cinq tirs d’envahisseurs apparaissent sur l’écran. Pour l’instant, ils resteront figés. Nous les ferons se déplacer dans l’étape suivante.

Étape 3

Dans cette étape nous allons nous occuper du déplacement des tirs d’envahisseurs. Pour commencer, nous allons définir une constante qui contiendra le pas de déplacement d’un tir (comme nous l’avions fait pour les déplacements du vaisseau et des envahisseurs).

```

; Pas d'incrémentation en pixels
; -----

SHIP_STEP      equ    4                ; Pas du vaisseau
SHIP_SHOT_STEP equ    4                ; Pas d'un tir de vaisseau
INVADER_SHOT_STEP equ  1            ; Pas d'un tir d'envahisseur
; ...

```

En vous inspirant du sous-programme **MoveShipShot**, qui gère le déplacement du tir du vaisseau, réalisez le sous-programme **MoveInvaderShots** qui déplace les tirs d’envahisseurs. Évidemment, une fois qu’un tir atteint le bas de l’écran, il redevient invisible et donc disponible. Attention, **MoveShipShot** gère un seul sprite (le tir du vaisseau), alors que **MoveInvaderShots** doit gérer plusieurs sprites (tous les tirs d’envahisseurs situés à partir de l’adresse InvaderShots).

Indications :

- Pour le déplacement d'un sprite, vous pourrez appeler le sous-programme **MoveSprite** déjà réalisé dans une étape précédente.
- Pour l'instant, l'animation des sprites (alterner les bitmaps 1 et 2) n'est pas demandée. Elle se fera dans l'étape suivante. Seul le bitmap 1 sera affiché.

Ajoutez ensuite un appel à ce sous-programme dans votre boucle principale et testez votre sous-programme :

Main	jsr	InitInvaders
	jsr	InitInvaderShots
\loop	jsr	PrintShip
	jsr	PrintShipShot
	jsr	PrintInvaders
	jsr	PrintInvaderShots
	jsr	BufferToScreen
	jsr	DestroyInvaders
	jsr	MoveShip
	jsr	MoveInvaders
	jsr	MoveShipShot
	jsr	MoveInvaderShots
	jsr	NewShipShot
	jsr	NewInvaderShot
	jsr	SpeedInvaderUp
	bra	\loop