

CORRECTION du Mini projet...



« **Space Invaders** est à l'origine un jeu vidéo de borne d'arcade programmé par le **Toshihiro Nishikado** pour des bornes Taito. Sorti en **1978** il s'agit de l'un des jeux vidéo les plus célèbres et des plus populaires jamais créés. » [Wikipédia](#)

Bien que simpliste selon les critères actuels, il rejoint Pac-Man et certains de ses contemporains au panthéon des classiques originels du jeu vidéo.

Travail demandé

1. Dessinez le personnage ci-dessus avec des caractères "#" sur l'écran du simulateur.

	I	N	V	A	D	E	R	S		
				#	A	T	T	A	C	K
FB00										
FB36										
FB6C		#						#		
FBA2			#				#			
FBD8		#	#	#	#	#	#	#		
FC0E		#	#		#	#	#	#	#	
FC44	#	#	#	#	#	#	#	#	#	#
FC7A	#		#	#	#	#	#	#		#
FCB0	#		#					#		#
FCE6			#	#		#	#			

2. Remplacez ce personnage par sa représentation lorsqu'il est détruit. Le passage d'une représentation à l'autre se fera dans une boucle infinie.

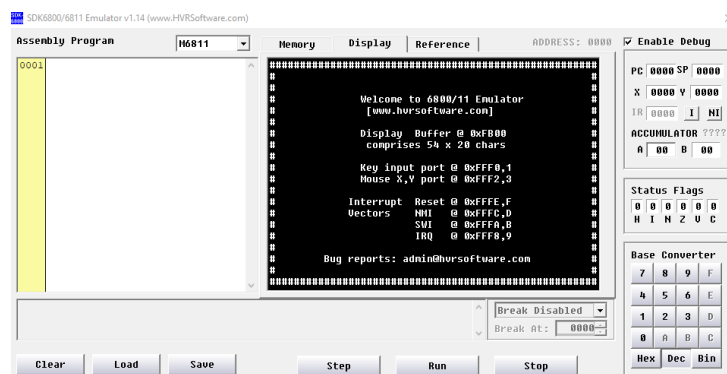
	I	N	V	A	D	E	R	S		
			#	D	E	S	T	R	O	Y
FB00										
FB36										
FB6C			-					-		
FBA2			-				-			
FBD8			-	-	-	-	-	-		
FC0E		-	-		-	-		-	-	
FC44	-	-	-	-	-	-	-	-	-	-
FC7A	-		-	-	-	-	-	-		-
FCB0	-		-					-		-
FCE6				-	-		-	-		

Conseils

Inspirez-vous des exercices du TP et utiliser le deuxième registre d'index y.

Ressource

Onglets **Display** et **Reference** du simulateur.



Fichier invaders.asm

```

; =====
; SPACE INVADERS - Motorola 6811
; Animation d'un alien (intact/détruit)
; Écran : 20 lignes x 54 caractères
; Position initiale : $FB00
; Vidéo : https://webge.fr/video/Invaders.mp4
; =====

; Constantes
ECRAN .equ $FB00 ; Adresse de début de l'écran
LARGEUR .equ 54 ; Largeur de l'écran en caractères ($36 en hexa)
DELAIR .equ $1 ; Constante pour temporisation

; =====
; PROGRAMME PRINCIPAL
; =====
main ldx #sprite1 ; X pointe sur le sprite intact
    bsr affiche ; Afficher le sprite
    bsr tempo
    ldx #sprite2 ; X pointe sur le sprite détruit
    bsr affiche ; Afficher le sprite
    bra main ; Boucle infinie

; =====
; SOUS-PROGRAMME : AFFICHE
; Affiche un sprite sur l'écran
; Entrée : X = adresse du sprite
; Utilise : A, B, X, Y
; =====
affiche ldy #ECRAN ; Y pointe sur la première ligne écran

nextline ldaa 0,x ; Charger le premier caractère de la ligne

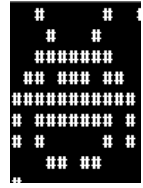
copychar staa 0,y ; Écrire le caractère à l'écran
    inx ; Caractère suivant dans le sprite
    iny ; Position suivante à l'écran
    ldaa 0,x ; Charger le caractère suivant
    bne copychar ; Si ≠ 0 (pas le nul de fin), continuer la ligne

; Fin de ligne atteinte (caractère nul automatique de .str)
    inx ; Passer le caractère nul
    ldd 0,x ; Charger l'adresse de la ligne suivante (D = A:B)
    beq faffiche ; Si = 0, fin du sprite

; Positionner Y sur la ligne d'écran suivante
; Transfert D vers Y sans xgdy
    std temp ; Sauvegarder D en mémoire temporaire
    ldy temp ; Charger Y depuis la mémoire
    inx ; Passer les 2 octets d'adresse
    inx
    bra nextline ; Traiter la ligne suivante
faffiche rts

; =====
; SOUS-PROGRAMME : TEMPO
; Temporisation simple (boucle imbriquée)
; Utilise : A, B
; =====
tempo ldaa #DELAIR ; Compteur externe
boucle1 ldab #DELAIR ; Compteur interne
boucle2 decb ; Décrémenter compteur interne
    bne boucle2 ; Continuer jusqu'à 0
    deca ; Décrémenter compteur externe
    bne boucle1 ; Continuer jusqu'à 0
    rts

```



```
; =====
; DONNÉES : SPRITES
; Format : .str ajoute automatiquement un caractère nul
;         .byte donne l'adresse écran (2 octets)
;         ou $0000 pour fin du sprite
;
; Écran 20 lignes x 54 caractères
; $FB00 = ligne 0, $FB36 = ligne 1, $FB6C = ligne 2,
; etc.
; Offset entre lignes = 54 ($36 en hexa)
; =====
.org $0090
```

```
; Alien intact - Ligne 0 : $FB00
sprite1 .str " INVADERS "
        .word $FB36          ; Ligne 1 : $FB00 + 54 ($36)

; Ligne 1
        .str " #ATTACK"
        .word $FB6C          ; Ligne 2 : $FB36 + 54 ($36)

; Ligne 2
        .str " #      #"
        .word $FBA2          ; Ligne 3 : $FB6C + 54 ($36)

; Ligne 3
        .str " #      #"
        .word $FBD8          ; Ligne 4 : $FBA2 + 54 ($36)

; Ligne 4
        .str " ##### "
        .word $FC0E          ; Ligne 5 : $FBD8 + 54 ($36)

; Ligne 5
        .str " ## ### ## "
        .word $FC44          ; Ligne 6 : $FB0E + 54 ($36)

; Ligne 6
        .str "#####"
        .word $FC7A          ; Ligne 7 : $FB44 + 54 ($36)

; Ligne 7
        .str "# ##### #"
        .word $FCB0          ; Ligne 8 : $FC7A + 54 ($36)

; Ligne 8
        .str "# #      # #" ; Ligne 9 : $FCB0 + 54 ($36)
        .word $FCE6

; Ligne 9
        .str " ## ## " ; Fin du sprite
        .word $0000
```

```
; Alien détruit - Ligne 0 : $FB00
sprite2 .str "INVADERS "
        .word $FB36          ; Ligne 1 : $FB00 + 54 ($36)

; Ligne 1
        .str " #DESTROY"
        .word $FB6C          ; Ligne 2 : $FB36 + 54 ($36)

; Ligne 2
        .str " -      - "
        .word $FBA2          ; Ligne 3 : $FB6C + 54 ($36)

; Ligne 3
        .str " -      - "
        .word $FBD8          ; Ligne 4 : $FBA2 + 54 ($36)

; Ligne 4
        .str " - - - - - "
        .word $FC0E          ; Ligne 5 : $FBD8 + 54 ($36)

; Ligne 5
        .str " -- --- -- "
        .word $FC44          ; Ligne 6 : $FB0E + 54 ($36)

; Ligne 6
        .str "-----"
        .word $FC7A          ; Ligne 7 : $FB44 + 54 ($36)

; Ligne 7
        .str "- - - - - -"
        .word $FCB0          ; Ligne 8 : $FC7A + 54 ($36)

; Ligne 8
        .str "- -      - -" ; Ligne 9 : $FCB0 + 54 ($36)
        .word $FCE6

; Ligne 9
        .str " -- -- " ; Fin du sprite
        .word $0000

; Variable temporaire pour transfert D vers Y
temp    .word 0
        .end
```