

#001

```

FUNCTION _INIT()
END

FUNCTION _DRAW()
END

FUNCTION _UPDATE()
END

```

L'éditeur de code de **PICO-8** permet de décrire (en Lua) le comportement et les interactions du programme chargé à partir d'une cartouche ou en cours de réalisation. Trois fonctions clef sont requises :

- **_INIT()** → procède à l'initialisation du contexte
- **_DRAW()** → dessine l'image affichée sur l'écran
- **_UPDATE()** → gère les évènements

_INIT s'exécute une seule fois (après saisie de **RUN**)

_DRAW et **_UPDATE** sont appelées 30 fois par seconde

```

FUNCTION _INIT()
END

FUNCTION _DRAW()
  RECTFILL(0,0,127,32,12)
  LINE(0,33,127,33,3)
  RECTFILL(0,34,127,127,11)
END

FUNCTION _UPDATE()
END

```

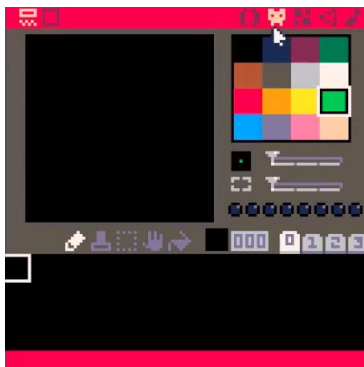
Les instructions pour dessiner à l'écran doivent être saisies dans le corps de la fonction **_DRAW**.

- **RECTFILL(x0,y0,x1,y1,c)** → dessine un rectangle plein de couleur **c**
- **LINE(x0,y0,x1,y1,c)** → dessine une ligne de couleur **c**

(x0,y0) sont les coordonnées du point supérieur gauche*.

(x1,y1) sont les coordonnées du point inférieur droit*.

* Dans un système orthonormé où (0,0) et (127,127) sont respectivement les coins supérieur gauche et inférieur droit de l'écran.



L'éditeur de sprites/tuiles permet de dessiner des figurines (de 8x8 pixels) à déplacer à l'écran ou des tuiles (de même taille) à juxtaposer pour former une mosaïque (carte).

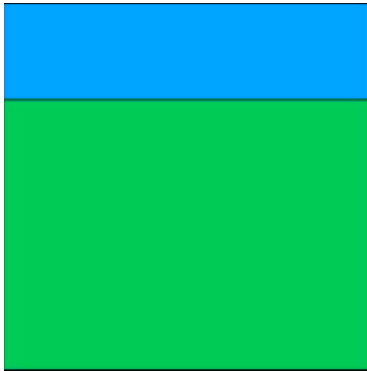
La palette de couleur est affichée dans le coin supérieur droit. 16 couleurs (non modifiables) sont disponibles : du noir (n°0) au beige (n°15) en passant par le vert foncé (n°3), le blanc (n°7), le vert clair (n°11), etc.

```

PICO-8
PICO-8 0.1.3
(C) 2014-15 LEXALOFFLE GAMES LLP
TYPE HELP FOR HELP
> RUN

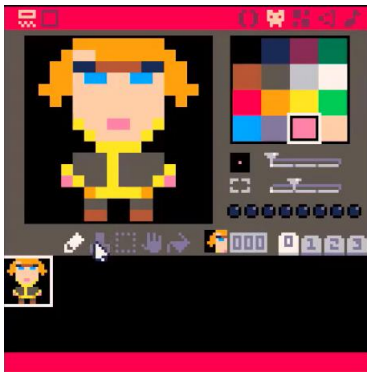
```

La mise au point d'un programme en cours de réalisation est grandement facilitée par la possibilité d'en exécuter tout ou partie à partir de la console. Pour avoir une idée de l'avancée du projet, il suffit d'accéder à la console (**[Echap.]**) et de lancer le programme : **RUN**.



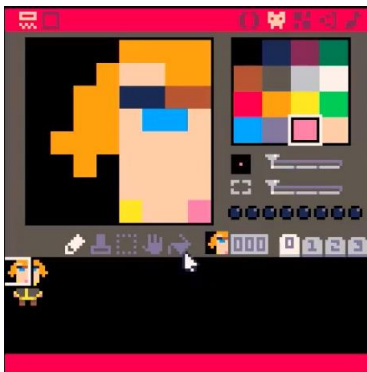
Le programme s'exécute. La fonction **_DRAW** est appelée (30 fois par seconde). L'image s'affiche : une magnifique prairie avec un ciel dégagé au-dessus d'une belle ligne d'horizon.

L'usage de la touche [Echap.] une première fois donne accès à la console. Un second appui active le mode d'édition (avec les 5 onglets : Code, Sprites, Carte, Effets, Musiques).

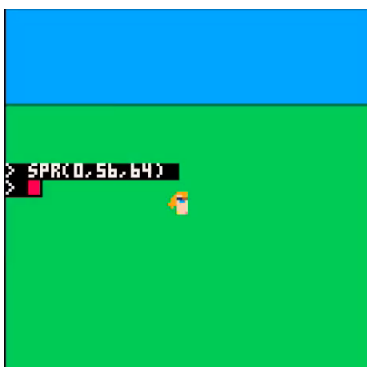


L'éditeur de sprites/tuiles permet de dessiner toutes sortes de choses avec un sprite (8x8) ou plusieurs. Dans le cas de cet exemple, le personnage utilise 4 sprites (emplacement 8x8 pixels).

Sous la palette de couleur, deux sélecteurs définissent respectivement la taille du pinceau et celle de l'espace de travail en haut à gauche. Les outils disponibles sont : le crayon, le tampon (coller), le tampon (copier), la main (déplacer) et le seau (remplir avec une couleur).



Les sprites (et/ou tuiles) sont numérotés à partir de 0 (premier en haut à gauche). Le personnage occupe les positions : 0, 1, 16 et 17. Quatre banques de sprites (0 à 3) sont disponibles. Les deux dernières (2 et 3) partagent leur emplacement mémoire avec les deux dernières portions de la carte. Par commodité, il est recommandé d'utiliser en priorité les deux premières banques (0 et 1).



L'instruction **SPR(s,x,y)** affiche le sprite à la position **s** (dans l'éditeur de sprites) en position **(x,y)** à l'écran.

- **SPR(0,56,64)** affiche donc le côté gauche de la tête du personnage à partir de la position (56,64).

L'espace (56,64) à (63,71) accueille la portion du dessin. La couleur noir (dite « de transparence ») laisse passer la couleur du fond.



Lorsque le dessin occupe plusieurs positions contiguës, la largeur et la hauteur peuvent être précisées : **SPR(s,x,y,l,h)**. Le personnage occupant une zone de 2x2 sprites peut ainsi être dessiné en une seule opération :

- **SPR(0,56,64,2,2)**

L'espace (56,64) à (71,79) accueille le dessin du personnage.



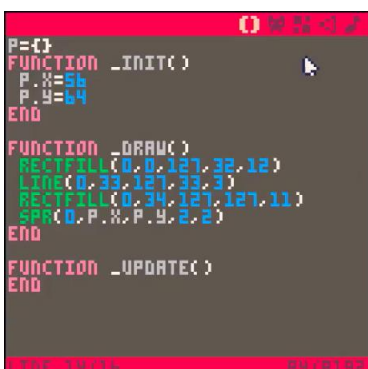
L'instruction peut être ajoutée dans l'éditeur de code à celles dessinant le paysage. Le dessin s'effectuant dans l'ordre, le personnage apparaîtra au-dessus de la prairie.



Une pression sur **[Echap.]**, et l'introduction de **RUN** à l'invite de commande permettent de le vérifier rapidement.

- **[Echap.]** interrompt l'exécution
- **[Echap.]** ramène à l'éditeur de code précédemment utilisé

Les choses sérieuses vont pouvoir commencer...



Lua gère différents types de variables : des tables ($P=\{\}$), des nombres ($P.X=56$) et des textes ($MSG="BONJOUR !"$).

L'étiquette **P** désigne un ensemble initialement vide : $\{\}$. La fonction d'initialisation **_INIT** place deux variables **X** et **Y** dans l'ensemble **P** (respectivement dotées des valeurs 56 et 64).

```

LINE(0,33,127,33,3)
RECTFILL(0,34,127,127,11)
SPR(0,P.X,P.Y,2,2)
END
FUNCTION _UPDATE()
IF BTN(0) THEN
P.X--2
END
IF BTN(1) THEN
P.X++2
END
IF BTN(2) THEN
P.Y--2
END
IF BTN(3) THEN
P.Y++2
END
END
END

```

La fonction **_UPDATE** comporte les traitements utiles à l'interaction avec le joueur. Les touches fléchées (gauche, droite, haut, bas) sont utilisées pour déplacer le personnage à l'écran.

- La fonction **BTN(n)** avec n compris entre 0 et 5 renvoie l'état de la touche : FAUX → relâché, VRAI → enfoncé.
- **BTN(4)** et **BTN(5)** correspondent aux touches **C** et **X**.

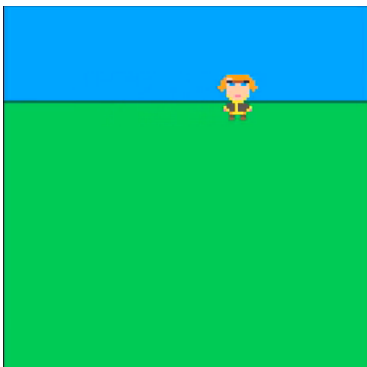
```

LINE(0,33,127,33,3)
RECTFILL(0,34,127,127,11)
SPR(0,P.X,P.Y,2,2)
END
FUNCTION _UPDATE()
IF BTN(0) AND P.X>1 THEN
P.X--2
END
IF BTN(1) AND P.X<112 THEN
P.X++2
END
IF BTN(2) AND P.Y>25 THEN
P.Y--2
END
IF BTN(3) AND P.Y<112 THEN
P.Y++2
END
END
END

```

A ce stade, un bref test (**[Echap.]**, **RUN**) met en évidence que le personnage peut se déplacer et même quitter l'écran. Pour remédier à cela, des limites doivent être définies pour limiter les déplacements en abscisse et en ordonnée.

- X compris entre 0 et 114
- Y compris entre 23 et 114



Le programme peut de nouveau être lancé (**[Echap.]**, **RUN**). Le personnage évolue maintenant librement à l'écran et ne peut plus en sortir (ni même s'élever dans le ciel).

A partir de l'invite de commande (accessible par **[Echap.]**), la réalisation peut être sauvegardée sous la forme d'une cartouche :

- **SAVE 001**