

# PROGRAMMATION ORIENTÉE JEUX

[ACCUEIL](#)[COURS THÉORIQUES](#)[APPLICATIONS PRATIQUES](#)

## MOTEUR DE JEU - LES MAINS

[INTRO](#)[LE COEUR](#)[LES MAINS](#)[LES PIEDS](#)[LES YEUX](#)[LES OREILLES](#)[CONCLUSION](#)

### QUE SONT LES 'MAINS' D'UN MOTEUR DE JEU ?

Les mains permettent au Moteur de se servir des ressources, de les charger en mémoire. Un jeu est avant tout constitué d'images !

#### 1) LES IMAGES

Une *Image* est une représentation mémoire d'un fichier BMP, PNG... permettant des actions simples:

- Chargement à partir d'un chemin d'accès
- Affichage à des coordonnées spécifiées
- Retourne ses dimensions (largeur/hauteur)

Une *Image* n'est donc pas modifiable. On peut s'en servir pour afficher une image de fond, ou tout élément à vocation statique.

Voici un exemple d'image:



Maintenant nous allons voir une forme évoluée des images.

#### 2) LES SPRITES

Les *Sprites* reprennent les propriétés des *Images*, et permettent des modifications, telles que le redimensionnement:

- Reprend les propriétés des Images
- Redimensionnement (largeur/hauteur)
- Rotation (sur 360°)
- Mirroir vertical et horizontal
- Filtrage (Bilinéaire, Bicubique)

Généralement, les *Sprites* sont utilisés pour un Menu, ou pour tout élément qui pourra être amené à être modifié par la suite.

Exemple de Sprite, redimensionné:



Il existe des *Sprites* plus avancés, réalisant d'autres fonctions, que nous allons voir.

#### 3) LES SPRITES ANIMÉS

Les *Sprites Animés* reprennent les propriétés des Sprites, permettant en plus la gestion des animations (les frames):

- Reprend les propriétés des Sprites
- Le chargement nécessite le nombre de frames horizontales et verticales
- Lors du rendu, seule la frame est affichée
- Joue une animation (simple ou en boucle)

- Stoppe une animation
- Retourne les dimensions d'une frame (hauteur/largeur)

Un *Sprite Animé* est un *Sprite*, découpé virtuellement en plusieurs parties (frames), représentant une animation. Généralement, les frames sont stockées de gauche à droite.

Exemple de *Sprite Animé*:



Un simple *Sprite* pourrait simuler ce comportement, en créant un tableau de *Sprite*, avec une suite d'images, et en les affichant successivement (mais ce procédé est moins efficace et plus coûteux en mémoire).

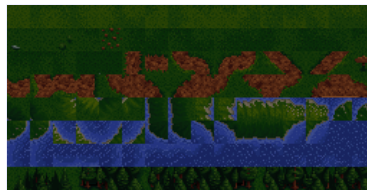
#### 4) LES TILES

Les *Tiles* reprennent les propriétés des Sprites, et pourraient ressembler aux *Sprites Animés*. En effet, les Tiles sont en fait des portions de Sprite, découpés en 'petits carrés', qui serviront à restituer généralement un niveau:

- Reprend les propriétés des Sprites
- Le chargement nécessite la taille d'un tile (largeur/hauteur)
- Lors du rendu, seul le Tile voulu est affiché
- Retourne les dimensions d'un Tile (hauteur/largeur)

C'est un peu comme les legos, il suffit d'avoir les pièces de référence pour bâtir n'importe quelle forme souhaitée.

Sprite contenant les tiles uniques:



Résultat qu'il est possible d'obtenir lorsque les données sont organisées:



Cette notion de *Tile* est très utile pour tout Jeu 2D, dont 99% des usages sont pour l'affichage de niveaux (beaucoup plus efficace que de traiter une seule grosse image). De plus, avec un *éditeur de niveau* (nous en reparlerons), il est très facile de créer de nouvelles cartes, simplement en faisant appel aux tiles !

Il existe d'autres types de *Tiles*, qui donne une impression de perspective, les *Tiles Isométriques*. Cependant, nous ne verrons pas le fonctionnement tout de suite.

Exemple de *Tile Isométrique*:



#### 5) LE PARALLAXE

Cette notion est plus compliquée que les précédentes, d'autant plus qu'il ne s'agit pas d'un type d'image à proprement parlé, mais plutôt d'un effet visuel. Il s'agit ni plus ni moins d'un effet de perspective en profondeur simulé par de la 2D. Avez-vous déjà regardé le défilement du

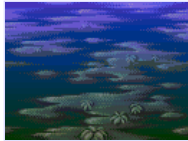
paysage depuis la voiture ou le train (très visible sur un terrain plat) ? Ce qui est au fond défile plus lentement que ce qui est proche de l'observateur. Le but recherché ici est le même, mais avec une image 2D.

La forme géométrique qui caractérise bien cet effet, n'est autre qu'un trapèze: / \_\_ \

Pour faciliter le traitement du rendu, on peut imaginer que ce trapèze est découpé en lignes parallèles (plus les lignes sont fines, plus l'effet sera fidèle). Ensuite, il faut que la ligne la plus en bas soit la ligne la plus large, et que la plus haute soit la plus courte. Puis, les lignes doivent défiler dans le sens contraire du mouvement du sujet (ce qui donne l'effet de déplacement, par rapport à un référentiel). Lorsqu'une ligne arrive en bout d'écran (droite ou gauche), elle revient à l'opposé, et continue de défiler, donnant l'impression qu'il en défile une infinité.

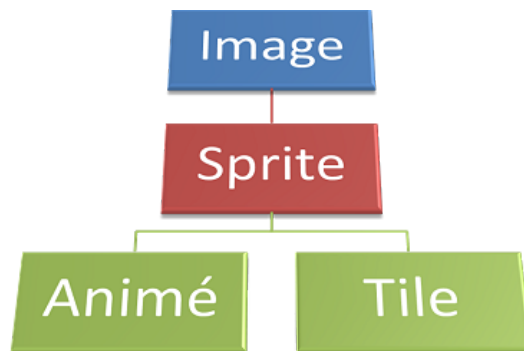
La seule difficulté, est de conserver la forme représenté par l'image (étant donnée que les lignes n'ont pas la même taille ni la même vitesse, il y a déformation). Il faut pour cela caler toutes les lignes sur la première ligne, qui servira de référence (quand la première ligne arrive en bout, la ligne du bas doit aussi y être arrivée).

Quand tout cela fonctionne bien, on obtient un résultat sympa (après de longues bidouilles plus ou moins explicables):

IMAGE DE BASE:	EXEMPLE EN ACTION:
	

## 6) CONCLUSION

Voilà les principaux composants permettant l'utilisation de ressources graphique. Il nous sera aisé de créer des Menus, d'appliquer une animation sur un personnage, et de gérer l'affichage d'un niveau.



Lire la suite: *Moteur de jeu - Les Pieds*