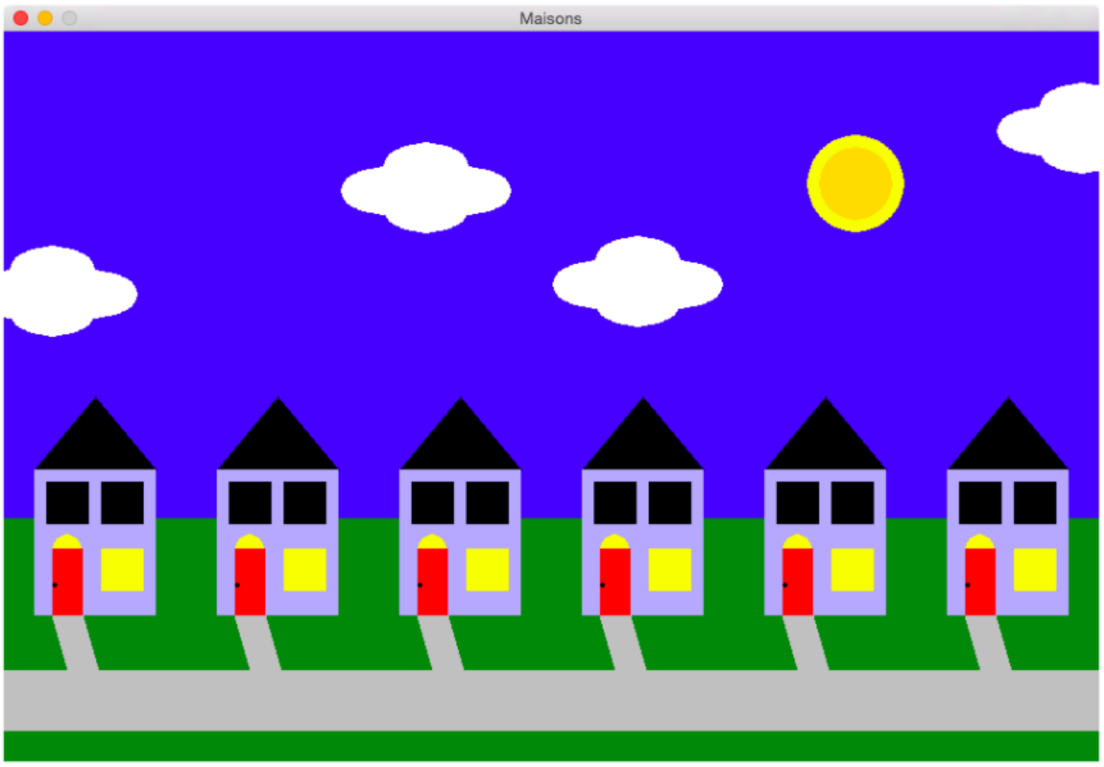
Travaux pratiques
7 Figure

Aujourd'hui, vous allez créer un dessin en Python avec Pygame. L'objectif est de pratiquer l'utilisation des modules et des fonctions graphiques. Voici une photo du dessin que vous allez créer.



En effet, vous devez créer un dessin qui ressemble à peu prêt celui-ci, mais vous n'êtes pas obligé de la faire au pixel prêt! Si vos nuages sont un peu plus grandes, votre soleil n'est pas exactement au même endroit, ou bien le maisons n'ont pas toute à fait de la même couleur, ce n'est pas grave! L'important est de programmer une animation qui ressemble celle si en respectant les quelques restrictions imposées.

Pour réaliser ce dessin, vous devez utiliser les fonctions `pygame.draw.rect`, `pygame.draw.circle`, `pygame.draw.polygon` etc. de Pygame. Vous pouvez consulter la documentation de la bibliothèque Pygame (en anglais):

`http://www.pygame.org/docs/ref/draw.html`

Par ailleurs, n'oubliez pas de faire la documentation de vos fonctions.

1 Le programme

Le programme que vous allez créer est composé de trois fichiers:

- `__main__.py` (module principal)
- `couleur.py` (module couleur)
- `dessin.py` (module dessin)

Piste: Vous pouvez télécharger les exemples du cours pour vous inspirer.

2 Le module principal

Pour pouvoir tester votre animation au fur et à mesure de ça création, vous devez commencer par le module principal du programme. La première chose à faire est donc de créer un dossier appelé `tp07` et à l'intérieur de ce dossier vous devez créer le fichier `__main__.py`. Ce fichier constitue le module principal du programme.

2.1 La fonction principale

Dans le module principal, écrivez la fonction `main` qui ne reçoit pas d'argument et n'a pas de valeur de retour. Cette fonction doit:

1. Initialiser Pygame.
2. Ouvrir la fenêtre de l'application et l'attribuer un titre.
3. Initialiser l'horloge.
4. Tant que le programme n'est pas terminé:
   (a) Traiter l'événement `pygame.QUIT`
   (b) Effacer la surface de l'application.
   (c) Mettre à jour la surface de l'application.
   (d) Ajuster la vitesse de la boucle.
5. Terminer Pygame.

2.2 Appel à la fonction principale

Ajoutez l'appel à votre fonction principale dans le module principal.

Testez votre programme maintenant et vérifiez que la fenêtre de l'application s'ouvre. À ce stade, elle est encore vide.

3 Module couleur

Vous devez créer un module `couleur.py` dans le dossier `tp07`. Définissez toutes les couleurs utilisées dans le scénario dans ce module. Comme dit auparavant, le dessin que vous allez créer doit ressembler celui ci-dessus. Néanmoins, ne perdiez pas trop de temps pour trouver les couleurs exactes. Vous n'êtes pas obligé d'utiliser exactement les mêmes couleurs dans votre figure.

Pour trouver le code RVB des couleurs dont vous aurez besoin dans votre animation, consultez la liste de couleurs sur wikipedia:

`http://fr.wikipedia.org/wiki/Liste_de_noms_de_couleur`

4 Module dessin

Vous devez créer un module `dessin.py` dans le dossier `tp07`. Importez le module scénario dans le module principal (`__main__.py`):

```
import dessin
```

4.1 Dessin

Écrivez la fonction `dessine` qui reçoit `surface` en argument. Cette fonction fera appel aux fonctions définies ci-dessous pour dessiner. Ajoutez l'instruction:

```
dessin.dessine(surface)
```

dans la fonction principal `main` du module principal du programme, où `surface` correspond à la surface de l'application. Placez cette instruction dans la boucle principale juste avant la mise à jour de la surface de l'application.

La fonction `dessine` est donc responsable pour dessiner tout le scénario. Au fur et à mesure que vous créez les autres fonctions de dessin (`dessine_ciel`, `dessine_maison`, etc.) ajoutez ses appels dans la fonction `dessine` et testez votre programme.

4.2 Le ciel

Nous allons maintenant commencer à dessiner. Écrivez la fonction `dessine_ciel` qui reçoit `surface` en argument. La fonction doit dessiner le ciel du scénario (sans les nuages). Ajoutez l'appel à à cette fonction dans la fonction `dessine` et testez votre programme.

4.3 Le soleil

Écrivez une fonction `dessine_soleil` qui reçoit `surface` argument. La fonction doit dessiner le soleil. Ajoutez l'appel à cette fonction dans la fonction `dessine` et testez votre programme.

4.4 Le sol

Écrivez une fonction `dessine_sol` qui reçoit `surface` argument. La fonction doit dessiner le sol (vert) et le trottoir. Pourtant, celle-ci ne doit pas dessiner le petit trottoir devant la porte de chaque maison (car cela sera fait dans la fonction qui dessine la maison). Ajoutez l'appel à cette fonction dans la fonction `dessine` et testez votre programme.

4.5 Les maisons

Écrivez une fonction `dessine_maison` qui reçoit un couple de deux entiers `(x, y)` et `surface` en arguments. La fonction doit dessiner une seule maison dans la position passée en argument, ainsi que le petit trottoir devant la porte. Ajoutez plusieurs appels à cette fonction dans la fonction `dessine`. Variez les positions des maisons pour créer l'impression d'une rue résidentielle. Testez votre programme.

4.6 Les nuages

Écrivez la fonction `dessine_nuages` qui reçoit un couple `(x, y)` et `surface` en arguments. La fonction doit dessiner tous les nuages à leur positions respectives sur l'écran. Ajoutez plusieurs appels à la fonction `dessine_nuage` dans la fonction `dessine` pour dessiner les nuages.

Piste: Les nuages sont dessinées avec des ellipses.