Image et Interaction - Graphismes avec OpenCV

Cyril Barrelet & Marc Hartley

December 2024

1 Introduction

Jusque là nos applications ne sont pas très attirantes. Ce TP vous propose quelques fonctions pour ajouter quelques éléments graphiques dans vos programmes.

1.1 Code de base

Nouveau morceau de projet, nouveau dépôt GitHub! Cette fois-ci, nous allons utiliser l'utilitaire présent dans le répertoire "OpenCVGraphics". Pour télécharger ce projet :

- Ouvrez un terminal.
- Déplacez-vous (avec `cd`) dans votre dossier de codes (par exemple "IG3/FASE/Image/")
- Récupérez le projet en lançant la commande
  `git clone https://github.com/marchartley/OpenCVGraphics.git`
- Un dossier "OpenCVGraphics" a du être créé. Ce sera la "racine du projet".
- Déplacez-vous à la racine du projet :
  `cd OpenCVGraphics`

2 Présentation du wrapper

Le code pour la gestion des éléments graphiques est contenu dans le fichier "Graphics.py". Importez ce code en ajoutant, en haut de votre fichier "main.py" :

```
from Graphics import Graphic, SceneRender
```

La classe `Graphic` propose une interface pour manipuler des images, et la classe `SceneRender` s'occupe de supperposer et afficher les `Graphic`s à l'écran.

Voyons ensemble un petit exemple d'utilisation :

```python
from Graphics import Graphic, SceneRender
import cv2
from FPSCounter import FPSCounter


def main():
    EPSILON = 1
    WIDTH, HEIGHT = 800, 600

    # On crée un élément graphique (un joli smiley)
    # cv2.IMREAD_UNCHANGED permet de conserver la transparence
    smiley = Graphic(cv2.imread("smiley.png", cv2.IMREAD_UNCHANGED))
    smiley.resize((50, 50), cv2.INTER_NEAREST)

    # Utilisation de la webcam
    cap = cv2.VideoCapture(0)

    # Instancie le "moteur de rendu"
    render = SceneRender((WIDTH, HEIGHT))

    fps = FPSCounter()

    while cap.isOpened():
        ret, img = cap.read()
        if not ret:
            break

        # On transforme "img" en un élément graphique manipulable
        webcam = Graphic(img)
        # webcam.fill((0, 0, 0))

        # On réduit l'image de la webcam
        webcam.resize((100, 100))
        # Ajout d'un petit effet visuel sur la camera
        webcam.apply_sketch_effect()
        # Et d'un peu de texte
        webcam.draw_text('Hello, OpenCV!', (50, 250), 'Fonts/Hollster.ttf', 30, (0, 0, 0), alpha = 0.8)

        # On crée un "tableau blanc"
        caneva = Graphic((WIDTH, HEIGHT))
        # On le rempli en blanc
        caneva.fill((255, 255, 255))

        # On dessine un rectangle de (10, 30) à (220, 170) bleu, rempli (thickness = -1)
```

```
        caneva.draw_rectangle((10, 30), (220, 170), (255, 100, 0), -1)
        # On y dessine un cercle centré en (90, 100), de 50 pixels de rayon, rouge, rempli (-1), et avec u
        caneva.draw_circle((90, 100), 50, (0, 0, 255), -1, alpha=0.9)
        # Un deuxième cercle centré en (100, 50), de 50 pixels de rayon, jaune, rempli (-1), et avec une p
        caneva.draw_circle((140, 100), 50, (0, 255, 255), -1, alpha=0.5)

        render.clear()
        # On place en fond le caneva
        render.add_layer(caneva)
        # Puis au premier plan la webcam (en bas à droite)
        render.add_layer(webcam, (WIDTH - 100, HEIGHT - 100))
        # Encore au dessus, un petit smiley
        render.add_layer(smiley, (WIDTH - 125, HEIGHT - 125))

        output = render.get_image()
        fps.update()
        output = fps.display(output, (0, 50))
        cv2.imshow("Resultat", output)

        key = cv2.waitKey(EPSILON) & 0xFF
        if key == ord("q") or key == 27:
            break

    cv2.destroyAllWindows()


if __name__ == "__main__":
    main()
```

Et normalement, déjà avec ça, vous pouvez déjà faire des applications plus vivantes!

3 Créer nos premiers objets de jeu

Notre but aujourd'hui : Coder un Pong. En une séance, c'est peut-être possible!

On va appliquer les principes de la POO (Programmation Orienté Objet) dans notre projet. L'intérêt de la POO, c'est de rendre votre code le plus lisible possible et le plus modulaire possible pour pouvoir le comprendre et le réutiliser dans le futur. On va commencer par créer les classes pour la gestion de la balle (Ball) et des raquettes (Paddle).

3.1 La classe Ball

Elle doit contenir un *sprite* (Graphic, qui est une image), une *position*, un *radius* (le rayon) et une *velocity* (la vitesse) qui sera un vecteur 2D défini par un np.array. Avec ça, on a toutes les informations pour gérer une balle.

On veut au moins les fonctions "update(dt)", "move(pos)", "get_graphic()".

La fonction "move(pos)" modifie la position de la balle vers la position "pos". Alternativement, vous pouvez considérer que la fonction "move(vec)" ajoute à la position le vecteur "vec" (i.e. applique une translation d'une

quantité "vec").

La fonction "update" doit modifier la position de la balle selon son attribut "velocity".

La fonction "get_graphic()" renvoie l'image de la balle.

Bien sûr, ne vous limitez pas seulement à ces 3 fonctions. Découpez votre code au maximum en sous-fonctions.

Si vous voulez avoir une super image pour votre balle, vous pouvez lancer le programme :

```
python3 face_selector.py
```

Le résultat devrait se retrouver enregistrée dans "output.png".

3.2 La classe Paddle

Elle doit contenir un sprite (Graphic), une position, une largeur et hauteur.

On veut les fonctions "update(dt)", "move(pos)", "get_graphic()".

4 Gestion de jeu

Dans notre programme principal, créez et affichez deux "Paddle" (un à droite et un a gauche de la fenêtre) et une "Ball" au centre. Donnez à la balle une velocité de (1, 1) (elle se déplacera en diagonal quand elle sera animée).

4.1 Animer le jeu

À chaque itération de la boucle principale, vous devez gérer plusieurs choses : déplacer la balle, et déplacer les paddles (si l'utilisateur appuie sur certaines touches).

Commençons par la balle. Dans la boucle principale, appelez la fonction "ball.update(EPSILON)". La position de la balle devrait se "translater" d'un pixel à droite et un pixel en bas. Verifiez que tout fonctionne.

Maintenant les paddles. Vous vous souvenez de la fonction "waitKey" dans notre boucle principale ? Elle permet de savoir si une touche du clavier est appuyée. Quand key == 82, c'est que la touche "flèche du haut" et key == 84 c'est que la touche "flèche du bas" est appuyée. Si l'une de ces touches est pressée, appelez la fonction "paddle.move()" pour déplacer le paddle vers le haut ou vers le bas.

4.2 Gérer les collisions

Si vous en êtes arrivé là, vous aurez peut-être remarqué que la balle ne rebondit pas du tout, ni avec les bords, ni avec les paddles... C'est le moment de régler ça!

Première étape : gérer la collision avec les bords. Dans la fonction "update()" de Ball, vérifiez sa position en X. Si $X < 0$ ou que $X > SCREEN\_WIDTH$ (qui pourrait être un paramètre de la fonction), alors inversez sa direction en modifiant la composante X de "self.velocity". De même pour la composante Y : si $Y < 0$ ou que $Y > SCREEN\_HEIGHT$, alors inversez la composante Y de "self.velocity".

Normalement, avec ça vous avez une balle qui rebondit un peu partout!

Maintenant il va falloir gérer les collisions avec les paddles. On va verifier si la boite englobante de la balle touche la boite englobante de chaque paddle. Voyons un peu comment faire ça.

Dans la classe Ball, créez une fonction "checkCollisionWithBox()". Cette fonction prend en paramètre les coordonnées d'un Paddle (coin en haut à droite, coin en haut à gauche, coin en bas à droite et coin en bas à gauche). À partir de la coordonnée de la balle, de son rayon, et de la boite englobante d'un Paddle, la fonction renvoie True s'il y a collision et False sinon. (Un exemple d'implementation en C est proposée ici : `https://www.jeffreythompson.org/collision-detection/circle-rect.php`). Utilisez cette fonction dans la fonction "update" de Ball.

4.3 Ajouter un score

Dans le jeu de Pong, la balle ne rebondit pas sur les côtés, vous pouvez alors retirer cette gestion de collision dans la fonction "update" de Ball.

Néanmoins, si la balle sort d'un côté, il faut ajouter des points à un joueur!

Exercices :

- Ajoutez une fonction "isOutside()" dans Balle qui retourne -1 si elle sort à gauche, 1 si elle sort à droite et 0 si elle est encore en jeu.
- Dans la boucle principale, verifiez que la balle est encore en jeu. Si ce n'est pas le cas, donnez un point à l'un des joueurs et remettez la position de la balle au centre.
- Inspiré de la classe FPSCounter ou CompteRebours du TP précédent, ajoutez un affichage de scores en haut de l'écran.

Pour aller plus loin, vous pouvez faire varier la gestion de la collision avec les Paddles : quand la balle tape au centre, la velocité en Y n'est pas vraiment affectée, mais si elle tape sur les bords du paddle, la velocité en Y devrait être exagérée.