



République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

---

# Université des Sciences et de la Technologie Houari Boumediene

Faculté d'informatique

Systèmes informatiques intelligents

**VISION PAR ORDINATEUR**

---

## PROJET DU TP

---

RÉALISÉ PAR LE QUADRINOME :

**YAHIAOUI MOHAMED ISLAM**

Matricule : **191931050515**

Groupe de TP : **02**

**DJELLAB NESRINE**

Matricule : **19199DZA6079**

Groupe de TP : **02**

**MEKDOUD RACHID**

Matricule : **191931092011**

Groupe de TP : **02**

**MEKKI FERIEL**

Matricule : **19199DZA6076**

Groupe de TP : **02**

**Promotion : 2023/2024**

# Sommaire

<b>Sommaire.....</b>	<b>0</b>
<b>I. Interface graphique :.....</b>	<b>1</b>
A- Implémentation des filtres vus en classe.....	1
B- Fonction de détection d'objet par couleurs et ses quelques utilisations :.....	1
C- Jeu interactif de course de voiture: L'utilisateur peut utiliser un objet physique pour guider la voiture dans le jeu, offrant ainsi une expérience immersive. De plus, une option de contrôle via le clavier est également disponible, offrant une flexibilité dans les méthodes d'interaction.....	2
<b>II. Description des filtres :.....</b>	<b>2</b>
II.1 Résultats des traitements sur Image.....	2
II.2 Description des filtres proposés:.....	3
<b>III. Description de la fonction object color detection :.....</b>	<b>4</b>
<b>IV. Invisibility cloak :.....</b>	<b>5</b>
IV.1 Principes et Architecture de la Fonction:.....	5
<b>V. Green screen :.....</b>	<b>5</b>
<b>Partie 2 , Brick racing game:.....</b>	<b>6</b>
A-Introduction :.....	6
B- Développement de jeu :.....	7
C- Utilisation de object detection :.....	7
D- Mesure de performance de jeu :.....	8
E- Les fonctions de base utilisé dans le jeu :.....	8
Conclusion :.....	9
<b>ANNEXE.....</b>	<b>10</b>
Guide d'utilisation de la GUI.....	10
Amélioration de la Fonction de Détection d'Objet par Couleur :.....	10
Guide d'utilisation de jeu - Brick Racing Game :.....	10

## I. Interface graphique :

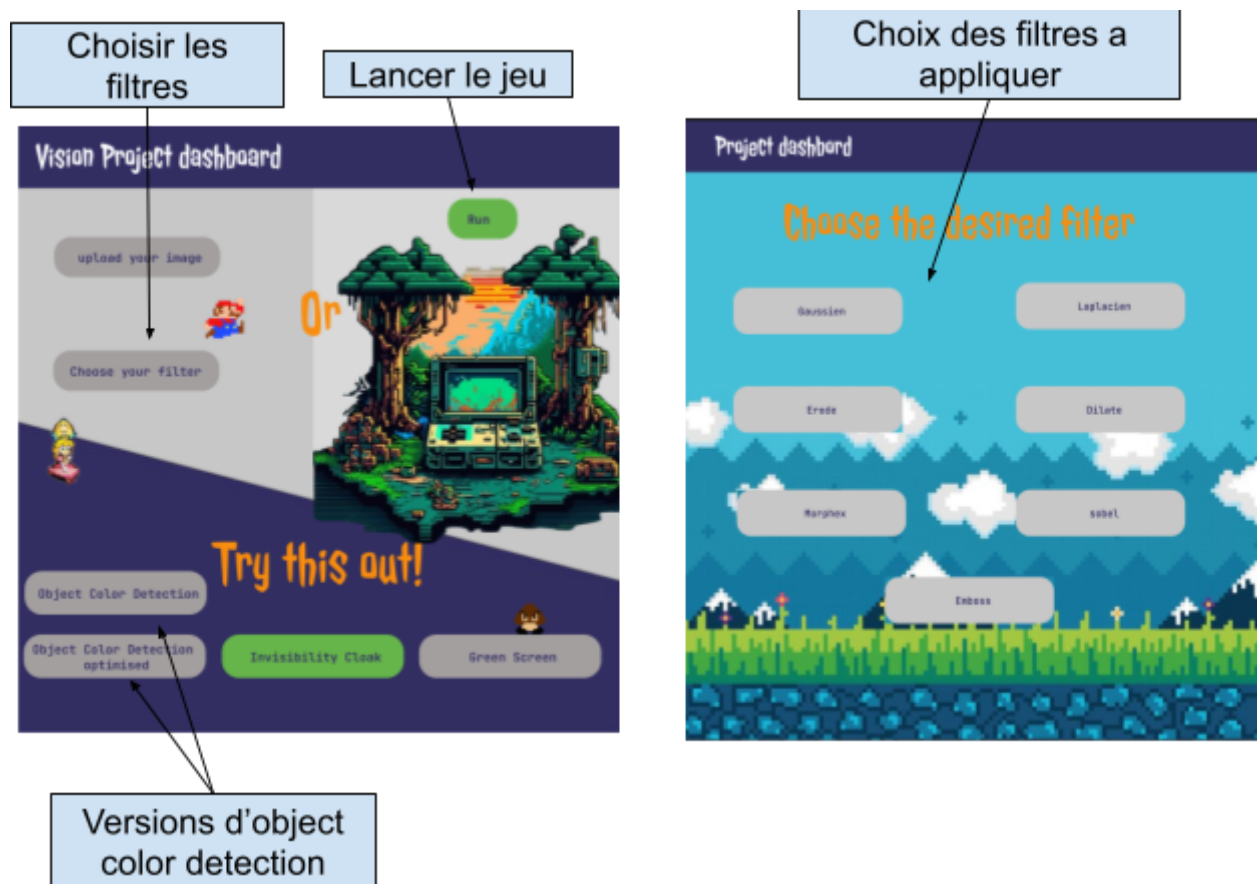


Figure 1: première Interface Graphique Utilisateur 1

Figure 2: Interface Graphique Utilisateur 2

Notre interface utilisateur est divisée principalement en 3 partie majeurs

### A- Implémentation des filtres vus en classe

L'utilisateur a la possibilité de sélectionner une photo et d'appliquer différents filtres selon des paramètres personnalisés. Cette fonctionnalité vise à permettre une manipulation visuelle des images.

Choix des paramètres: Struct: 0=Rectangle, 1=Cross, Struct\_Morph: 0=Open, 1=Closed.

### B- Fonction de détection d'objet par couleurs et ses quelques utilisations :

Cette partie se concentre sur une fonction de détection d'objet par couleurs. Trois options sont mises à la disposition de l'utilisateur : la détection d'objet par couleur, la cape d'invisibilité, et la substitution de l'arrière-plan par un fond vert.

**C- Jeu interactif de course de voiture:** L'utilisateur peut utiliser un objet physique pour guider la voiture dans le jeu, offrant ainsi une expérience immersive. De plus, une option de contrôle via le clavier est également disponible, offrant une flexibilité dans les méthodes d'interaction.

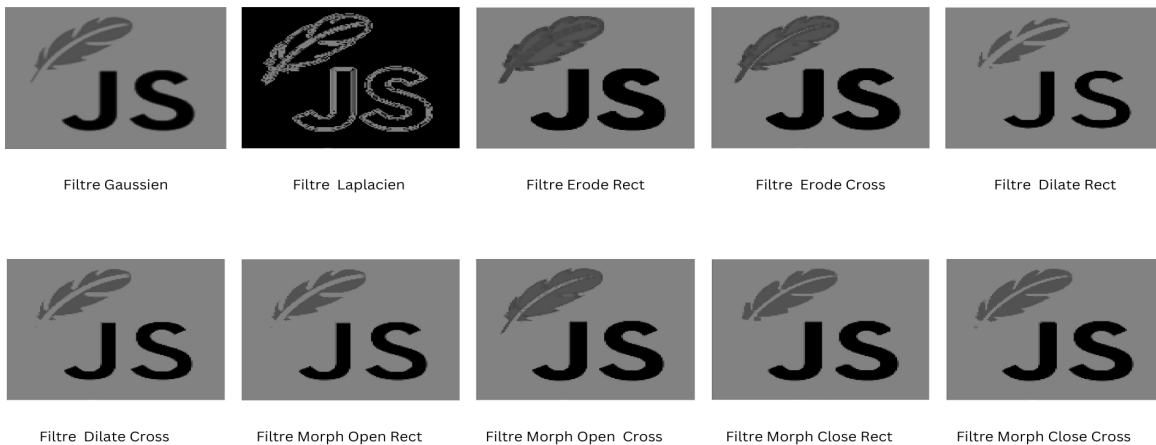
## II. Description des filtres :

- **Filtre 2D :** Applique un noyau de filtre sur chaque pixel pour effectuer des opérations telles que l'amélioration de la netteté, la détection de contours, et la suppression du bruit.
- **Filtre Laplacien :** Accentue les variations d'intensité pour mettre en évidence les bords et les détails fins dans une image.
- **Filtre Gaussien :** Floute une image en atténuant les hautes fréquences, réduisant ainsi le bruit.
- **Érosion et Dilatation :** Manipulent la forme des objets dans une image en réduisant ou augmentant les contours.
- **Ouverture et Fermeture :** Techniques avancées utilisant des opérations combinées d'érosion et de dilatation pour éliminer le bruit, séparer les objets proches, et fermer les lacunes dans les contours.

### II.1 Résultats des traitements sur Image



Image originale en grayscale



Résultats des applications des filtres vus en TP sur une image avec un thresholding de 130 de type 2

## II.2 Description des filtres proposés:

**Filtre de Sobel** : Calcule les gradients horizontaux et verticaux d'une image pour mettre en évidence les contours.

Le filtre de Sobel est choisi lorsque l'objectif est de détecter les contours et les variations d'intensité dans une image. C'est une option robuste pour la segmentation d'objets et la détection de bordures, ce qui peut être essentiel dans des applications telles que la vision par ordinateur, la détection d'objets et le traitement d'images médicales.

**Filtre d'Embossage** : Accentue les variations d'intensité pour créer un effet tridimensionnel dans une image.

Le filtre d'embossage est choisi lorsque l'objectif est d'ajouter des effets visuels tridimensionnels à une image. Il est utilisé à des fins esthétiques et artistiques pour créer des effets de relief. Ce filtre peut être approprié dans des applications telles que la retouche d'images artistiques, la création de contenu visuel et la génération d'effets spéciaux.



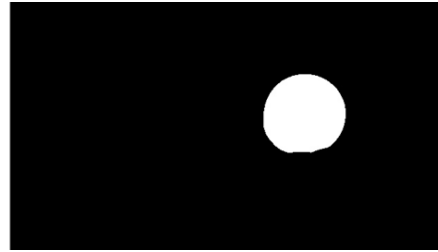
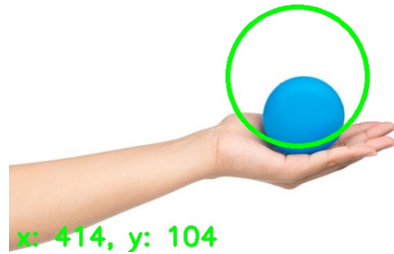
Résultats des applications des filtres de sobel et d'embossage sur une image avec un thresholding de 130 de type 2

---

## III. Description de la fonction object color detection :

La fonction **object color detection**, a pour rôle de détecter un objet dans une image en se basant sur sa couleur. Voici une brève description de ses composants :

- **Fonctionnement :**
  - La fonction opère une conversion de l'image BGR à HSV, puis parcourt les pixels, vérifiant s'ils se situent dans la plage de couleurs définie. Elle construit ainsi un masque et une liste de points repérant l'objet. En actualisant l'image originale pour marquer les pixels de l'objet, la fonction renvoie le masque, l'image modifiée avec l'objet détecté, et la liste des points associés à la détection.
- **Fonction auxiliaire :** La fonction **Launch\_object\_color\_detection** capture la vidéo depuis la caméra, applique la détection d'objet en temps réel, et affiche les résultats, y compris la position de l'objet détecté. L'exécution de cette fonction lance la capture vidéo en direct et la détection d'objet en utilisant la fonction principale



La fonction actuelle de traitement d'images présente des défis majeurs. Tout d'abord, le temps de traitement est significativement long, particulièrement en temps réel, en raison du nombre élevé de calculs nécessaires pour chaque image. De plus, la précision des résultats est compromise, la méthode actuelle montrant une lacune notable dans la capacité à repérer avec précision l'objet dans l'image. Ces deux aspects soulignent la nécessité pressante d'optimiser la fonction pour améliorer à la fois la rapidité de traitement et la précision des résultats obtenus, cette amélioration on va la présenter dans l'annexe

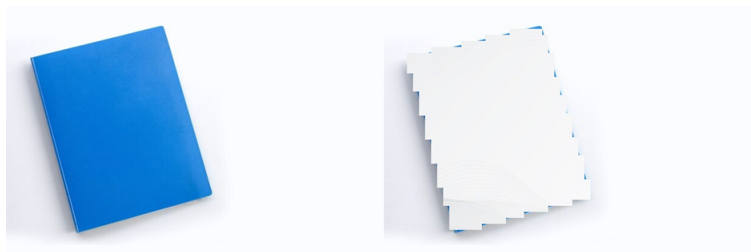
---

## IV. Invisibility cloak :

Cette fonction, appelée "**invisibility\_cloak**", a pour rôle de rendre un objet détecté dans une image capturée par une caméra invisible en le remplaçant par un arrière-plan déjà enregistré . Voici une description détaillée de la fonction :

### IV.1 Principes et Architecture de la Fonction:

La fonction a pour rôle de rendre un objet invisible dans une image en remplaçant ses pixels par ceux de l'arrière-plan fourni. Pour ce faire, elle prend en entrée une image (frame), un arrière-plan, une liste de points représentant l'objet détecté, et un masque de l'objet. En suivant son principe opérationnel, la fonction utilise les points et le masque pour localiser la position de l'objet dans l'image, puis parcourt les pixels de l'image en remplaçant les pixels de l'objet par ceux correspondants de l'arrière-plan. De plus, elle effectue une légère propagation du fond sur les pixels environnants pour obtenir un effet plus naturel. L'architecture générale de la fonction intègre la détection de la position de l'objet à partir des points identifiés, l'utilisation du masque pour cibler les pixels de l'objet, et le remplacement de ces pixels par ceux de l'arrière-plan, avec une amélioration de la transition grâce à la légère propagation du fond sur les pixels environnants



Cette fonctionnalité globale crée un effet de "cape d'invisibilité" en temps réel à l'aide d'une caméra, en remplaçant l'objet détecté par un arrière-plan capturé au préalable

---

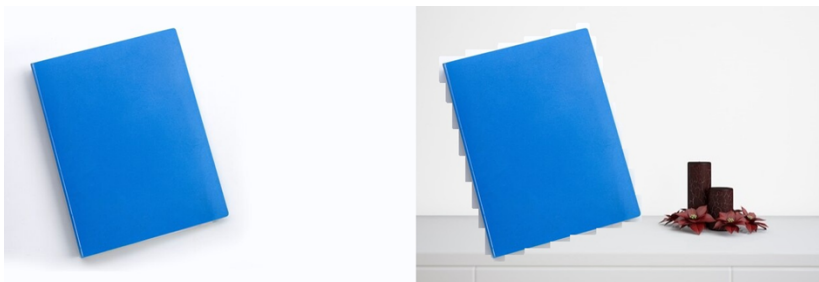
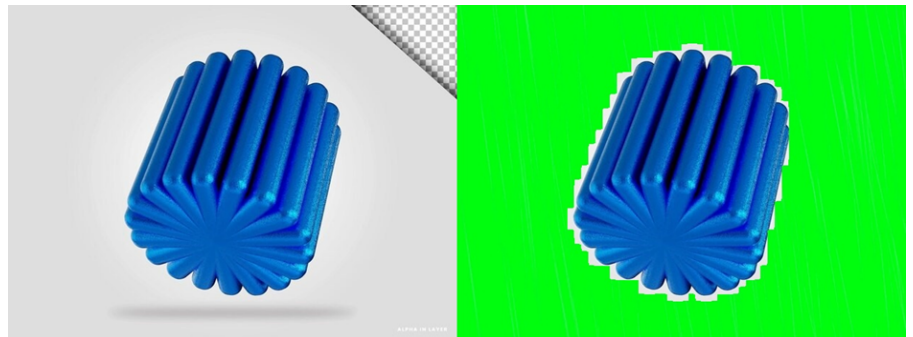
## V. Green screen :

Cette fonction, appelée "**Green\_screen**", vise à appliquer un effet de fond vert à l'objet détecté dans une image capturée par une caméra. Voici une description détaillée de la fonction

**i. Rôle :** La fonction prend en entrée une image (frame), une liste de points qui repère l'objet détecté, un arrière-plan (background), et un masque de l'objet. Elle applique l'effet de fond vert à l'objet détecté, en remplaçant l'arrière-plan par celui fourni

**ii. Fonctionnement :** La fonction commence par redimensionner le masque et l'étendre pour inclure quelques pixels supplémentaires, dans le but de mieux délimiter l'objet et de conférer une touche plus naturelle à l'image. Ensuite, elle extrait la région exacte de l'objet dans la trame, s'assurant que l'arrière-plan a la taille appropriée, et crée un masque inversé. En utilisant ces composants, elle réalise un mélange (blending) des pixels de l'objet avec ceux de l'arrière-plan pour obtenir une nouvelle région résultante. Enfin, elle met à jour l'arrière-plan avec cette région résultante

Comme on peut l'observer sur la photo à droite, c'est une application de la fonction sur une image où la couleur détectée est le bleu. On peut clairement noter le résultat fourni, où l'arrière-plan est transformé en vert (effet de green screen )



À gauche, dans la photo adjacente, on remarque que l'arrière-plan a été altéré et remplacé par un autre, illustrant ainsi l'effet de fonction green screen

## Partie 2 , Brick racing game:

---

### A-Introduction :

On va entamer dans cette partie le développement d'un jeu de voiture qui vise à éviter des obstacles utilisant la détection d'objets par la vision par ordinateur. Ce jeu combine des éléments de réalité virtuelle capturés par la caméra pour créer une expérience interactive. La détection d'objets est employée pour manipuler le déplacement d'un véhicule virtuel à l'intérieur du jeu, tandis que le score et la vitesse du jeu sont calculés en temps réel.

### B- Développement de jeu :

Le jeu s'appuie sur la capture vidéo en temps réel provenant de la caméra intégrée. À chaque itération, la caméra détecte un objet, utilisant les informations de couleur en HSV base. Ces données guident le déplacement horizontal du véhicule dans le jeu. La voiture réagit en temps réel aux mouvements de l'objet détecté, créant ainsi une expérience de jeu interactive et dynamique.

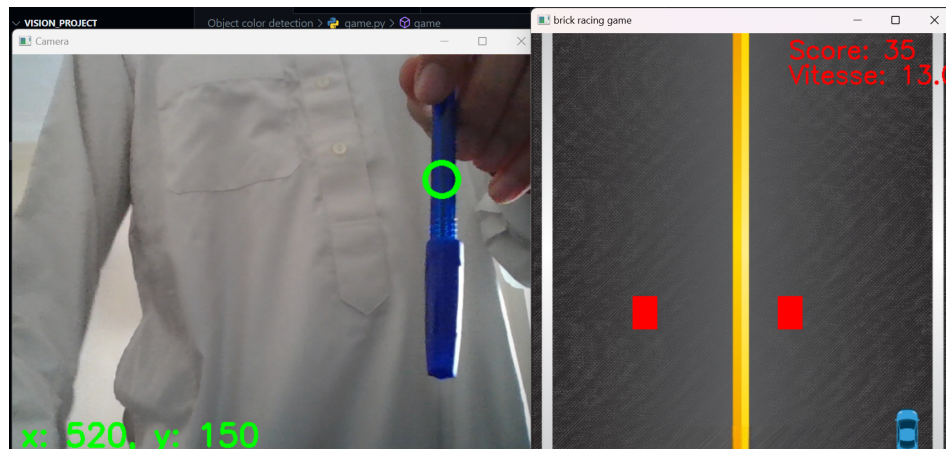


Figure 1: Interface Graphique Utilisateur

Au début du programme, les images et les vidéos nécessaires au jeu doivent être chargées. Dans ce cas, il est nécessaire de charger les images suivantes :

- L'image de la voiture avec canal alpha : cette image est utilisée pour créer l'effet de transparence de la voiture.
- L'image de contour : cette image est utilisée comme arrière-plan du jeu.
- La vidéo depuis la caméra : cette vidéo est utilisée pour détecter l'objet.

L'image de la voiture est reproduite, pixel par pixel, afin d'être superposée au contour. Ce dernier constitue le fond de la fenêtre de jeu.

La boucle principale du jeu englobe plusieurs étapes cruciales pour son fonctionnement fluide.

Initialement, le frame capturé depuis la caméra guide l'ensemble du processus. Ensuite, la détection de l'objet, réalisée par une méthode de détection de contours, détermine les coordonnées de l'objet,



influençant ainsi la position horizontale de la voiture dans le jeu. Les obstacles, représentés par des formes rouges, se déplacent vers le bas de l'écran à une vitesse constante, ajoutant un élément de défi. Le score est continuellement mis à jour chaque fois que le joueur n'a pas heurté un obstacle, renforçant l'expérience du joueur. La voiture, positionnée au bas de l'écran, se déplace vers la gauche ou la droite en fonction des coordonnées de l'objet détecté. Les collisions entre la voiture et les obstacles sont vérifiées régulièrement, et en cas de collision, le jeu affiche "*Game Over*" avant de se fermer. Enfin, la fenêtre du jeu est constamment actualisée, assurant une expérience visuelle dynamique et immersive pour le joueur.

### **C- Utilisation de object detection :**

En détectant l'objet à chaque itération et en enregistrant la position de cet objet pour chaque image capturée, le jeu réagit de manière dynamique. Si la position de l'objet indique que celui-ci se trouve dans une certaine plage, la voiture ajuste sa position en conséquence. Sinon, la voiture conserve sa position actuelle. Cette approche permet une interaction fluide entre le joueur et le jeu, créant ainsi une expérience de jeu immersive.

### **D- Mesure de performance de jeu :**

Le jeu implémente un système de score basé sur la durée de survie du joueur. À chaque itération, le joueur gagne un point, et la vitesse du jeu augmente progressivement. Le score et la vitesse sont affichés en temps réel dans le coin supérieur droit de la fenêtre du jeu.

### **E- Les fonctions de base utilisé dans le jeu :**

#### **La fonction check\_collision :**

L'objectif principal de la fonction check\_collision est de déterminer si la voiture (représentée par un rectangle défini par ses coordonnées, largeur et hauteur) entre en collision avec un obstacle (également défini par ses coordonnées, largeur et hauteur).

#### **Fonctionnement:**

La fonction compare les positions et dimensions de la voiture et de l'obstacle pour déterminer si elles se chevauchent. Elle utilise les coordonnées des coins de chaque rectangle (coin supérieur gauche et coin inférieur droit) pour effectuer ces comparaisons.

Si les côtés droit et gauche de la voiture ne dépassent pas les côtés gauche et droit de l'obstacle, et si le bas et le haut de la voiture ne dépassent pas le haut et le bas de l'obstacle, alors il y a collision. Sinon, il n'y a pas de collision.

#### **la Fonction resize\_image\_3d:**

L'objectif principal de la fonction `resize_image_3d` est de redimensionner une image tridimensionnelle en utilisant un facteur d'échelle spécifié.

### Paramètres:

`image`: Image 3D à redimensionner (un tableau NumPy tridimensionnel).

`scale_factor`: Facteur d'échelle pour redimensionner l'image.

### Retour:

Une image redimensionner

### Fonctionnement :

La fonction prend en entrée une image 3D représentée par un tableau NumPy. Elle extrait la hauteur et la largeur de l'image, puis calcule la nouvelle hauteur et la nouvelle largeur en multipliant la hauteur et la largeur originales par le facteur d'échelle. Ensuite, elle crée un nouveau tableau avec les dimensions ajustées.

La fonction itère ensuite sur chaque pixel du nouveau tableau et assigne la valeur du pixel correspondant dans l'image originale à la position ajustée. Cela est fait pour chaque canal de couleur (composante RGB) de l'image

### Conclusion :

Le jeu avec détection d'objets offre une expérience immersive en combinant la vision par ordinateur et le jeu interactif. La détection d'objets permet une interaction en temps réel avec l'environnement virtuel, tandis que le score et la vitesse ajoutent des éléments de défi et de compétition. Ce projet démontre le potentiel de la vision par ordinateur dans le développement de jeux interactifs innovants.

## ANNEXE

### Guide d'utilisation de la GUI

Avant d'exécuter le fichier , veuillez installer le font Jolly Lodger en cliquant sur le fichier "JollyLodger-Regular.ttf" ,ensuite cliquer sur installer.

Ensuite, lancer l'interface graphique principale en exécutant le fichier "firstgui.py"

### Amélioration de la Fonction de Détection d'Objet par Couleur :

Pour remédier aux problèmes rencontrés avec la fonction, il est impératif de développer une stratégie d'amélioration. Dans la suite, nous présentons une analyse approfondie des composants et des améliorations apportées à la fonction de détection de couleur d'objet précédente

- **Propositions d'Amélioration :**

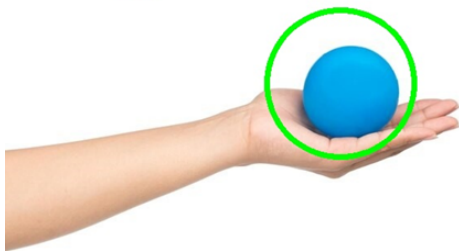
#### i.Redimensionnement de l'image :

L'image subit une opération de redimensionnement avant le lancement de la détection, permettant ainsi de travailler avec une résolution réduite. Cette étape a été cruciale pour minimiser la latence et améliorer la vitesse du processus. En effet, cette amélioration majeure s'est avérée déterminante pour rendre la détection de l'objet réalisable dans un laps de temps raisonnable

**ii.Application des filtres sur l'image :** Cette amélioration vise à réduire le bruit et à renforcer la robustesse de la détection

#### Détection de Contours Optimisée & Seuillage Paramétré et amélioré

x: 410, y: 150



Ces modifications visent à optimiser la détection d'objet par couleur, réduisant le bruit et améliorant la précision tout en préservant la simplicité d'utilisation

---

### Guide d'utilisation de jeu - Brick Racing Game :

Le Brick Racing Game, un jeu où votre voiture doit éviter les obstacles en se déplaçant de gauche à droite à l'aide de la détection d'objet de la caméra. Voici un guide pour vous aider à profiter de l'expérience de jeu.

### 1. Prérequis:

- Placez les images de la voiture ('car.png') et du contour ('contour.jpg') dans le dossier 'Object color detection/Images'.
- Assurez-vous d'avoir installé les bibliothèques nécessaires (OpenCV, NumPy).

### 3. Interface du Jeu:

- Deux fenêtres apparaîtront à l'écran : la fenêtre de la caméra et la fenêtre du jeu.

### 4. Contrôles:

Déplacement de la Voiture :

- L'objectif est d'éviter les obstacles en déplaçant la voiture de gauche à droite.
- La voiture suit le mouvement de l'objet détecté par la caméra.

Quitter le Jeu :

- Appuyez sur la touche 'q' pour quitter le jeu à tout moment.

### 5. Détection d'Objet et Mouvement de la Voiture:

- L'objet détecté par la caméra est représenté par des cercles et des points dans la fenêtre de la caméra.
- La voiture suit automatiquement le mouvement de cet objet.
- Déplacez l'objet horizontalement pour déplacer la voiture à gauche ou à droite.

### 6. Obstacles et Score:

- Des obstacles rouges descendent de la partie supérieure de la fenêtre du jeu.
- Évitez les obstacles pour maintenir votre score.
- Le score et la vitesse actuelle sont affichés en haut à droite de la fenêtre du jeu.

### 7. Game Over:

- Le jeu se termine si la voiture entre en collision avec un obstacle.
- "Game Over sera affiché, suivi d'une pause de 2.5 secondes avant de fermer automatiquement la fenêtre.

### 8. Déplacements Précis de la Voiture:

- La voiture se déplace vers la gauche si l'objet est à gauche de l'écran.
- La voiture se déplace vers la droite si l'objet est à droite de l'écran.