Projet Deadpool

Éléments utilisés

* Raspberry Pi EU & UK Power Supply 2.5A, 5.1V, Micro USB B, 1.5m Cable
* Raspberry Pi 3 Model B, Bluetooth 4.1, 1GB RAM
* Raspberry Pi Camera V2, 8MP
* Raspberry Pi Camera Rev 1.3
* Cercle 12 LED
* Projecteur Medion

Objectifs

* Linux
* Python
* Programme d’entraînement en temps réel
* Open Source
* YOLO V3 (Traitement d’images)
* ArUco (calibration)

Problèmes de l’ancien prototype

* Puissance projecteur
* Vitesse de traitement d’images (puissance Raspberry Pi 3)
* Calibration

Recherche traitement d’images avec python :

<https://www.youtube.com/watch?v=RaCwLrKuS1w>

Permet de détecter des objets circulaires (style balles) sur une image envoyée par une caméra et à l’air de fonctionner super bien.

<https://github.com/sgrieve/PoolTable>

Projet pratiquement identique.

OpenCV :

<https://fr.acervolima.com/tutoriel-opencv-python/#images>

<https://fr.acervolima.com/detection-de-cercle-avec-opencv-python/>

<https://www.youtube.com/watch?v=t-f9Z1Wztoo> (Test2CircleDetection)

Détection d’objet/de fonctionnalité (objdetect, features2d, nonfree)

1. Importation de l’image
2. Analyser et manipuler l’image
3. Sortie dans laquelle le résultat peut être modifié image ou rapport basé sur l’analyse d’image

Détection de couleurs :

<https://www.youtube.com/watch?v=cMJwqxskyek>

Étapes du projet :

1. Détecter les billes (si possible en fonction des couleurs)
2. Comment incrémenter YOLO V3 dans son code (ou alternatives)
3. Délimiter la surface de jeu avec code ArUco automatiquement (ou alternatives [par rapport aux trous])
4. Menu ordre des opérations
5. Mini jeux simples et scénarios avancés
6. Support caméra/projecteur
7. Temps réel