

Document Technique

Morpion Robot Niryo Ned2

1. Informations Générales.....	2
2. Résumé.....	2
3. Objectifs du programme.....	2
4. Description du programme.....	2
4.1 Architecture Générale.....	2
4.2 Fonctionnalités Implémentées.....	3
4.3 Déroulement Fonctionnel.....	3
5. Configuration du Robot.....	4
6. Aménagement de l'Environnement.....	5
6.1 Terrain.....	5
6.2 Conditions idéales.....	5
7. Installation et Exécution.....	5
8. Tests et Résultats.....	6
9. Evolutions Possibles.....	6
10. Annexe.....	6

1. Informations Générales

Nom du projet : Morpion

Date : 27/05/25

Version du document : 1.0

Auteurs : Escobar Paolo, Béranger Kylian

Organisation : Techlab

Robot concerne : Niryo Ned2

2. Résumé

Ce programme permet de faire s'affronter le robot Niryo Ned2 et un utilisateur humain au jeu du morpion (ou Tic-Tac-Toe).

C'est un jeu de réflexion se pratiquant à deux joueurs, tour par tour, chaque joueur choisit un type de jetons et les place sur un plateau de taille 3x3.

L'objectif est d'aligner 3 de ses pions en premier.

Le programme utilise la pince et la caméra dont peut être équipé le robot Niryo Ned2 afin de connaître l'état du plateau et jouer ses pièces.

3. Objectifs du programme

- Traduire les règles du morpion informatiquement
- Ne pas pouvoir perdre face au joueur
- Transmettre des informations au joueur pour suivre l'état de la partie

4. Description du programme

4.1 Architecture Générale

- Langage utilisé : Python
- Bibliothèques principales :
 - pyniryo : permet de piloter le bras robotique Niryo
 - OpenCV : utile pour le traitement d'image (détection de formes, de contour de barycentres)

- Structure du programme :
 - `morpion.py` : script principal contenant la logique de déroulement du jeu, l'interaction avec le robot, et la vision par ordinateur.
 - `minimax.py` : module dédié à la prise de décision, contenant l'implémentation de l'algorithme Minimax afin de réaliser le meilleur choix.

4.2 Fonctionnalités Implémentées

- Détection de formes via la caméra du robot et traitement morphologique de l'image
- Calibrage d'un repère dynamique nommé Morpion : défini à partir de trois points physiques sur la table de jeu, permettant de déterminer la position des 9 cases du plateau.
- Mise à jour automatique de la grille de jeu en détectant les objets posés par l'humain.
- Implémentation du tour de jeu : alternance entre l'utilisateur humain et le robot en détectant une modification du jeu lorsqu'un jeton a été placé par l'utilisateur.
- Détection des conditions de victoire, égalité, ou poursuite du jeu.
- Calcul du meilleur coup à jouer via un algorithme Minimax intégré, prenant en compte la profondeur de recherche pour simuler plusieurs coups à l'avance.

4.3 Déroulement Fonctionnel

1. Initialisation :
 - Connexion au robot
 - Calibration automatique
 - Définition du repère dynamique
 - Réglage de la caméra
 - Initialisation de la matrice de jeu (3x3)
2. Choix du premier joueur :
 - L'utilisateur entre 1 ou 2 dans la console, 1 fait commencer le robot, 2 l'utilisateur.
3. Boucle principale
 - Tour du joueur : le robot attend un mouvement (nouvelle forme détectée)
 - Tour du robot : le robot calcule le meilleur coup, prend un pion, et le place sur la case correspondante
 - À chaque tour, vérification des conditions de victoire ou égalité, affichage de l'état du jeu dans la console
 - Répétition de la boucle jusqu'à ce qu'un gagnant soit trouvé ou qu'il y ait égalité

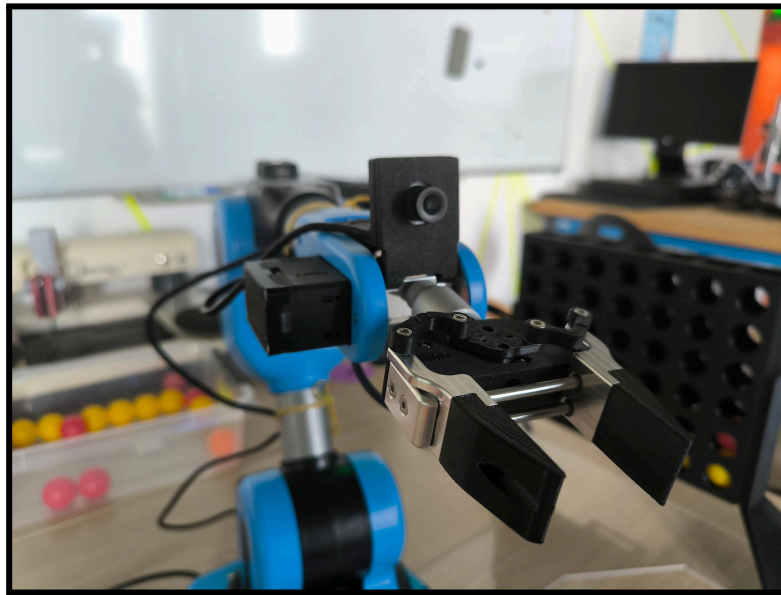
4. Fin du jeu

- Annonce du gagnant ou d'un match nul dans l'invite de commande
- Retour à la position "home" et fermeture propre de la connexion

5. Configuration du Robot

Matériel utilisé :

- Caméra (Niryo Vision Set)
- Pince (Niryo Ned2 Standard Gripper)



Vérifier que le robot Ned2 est bien alimenté, allumé (bouton presseur à l'arrière du robot) et que le **hotspot est activé** (Led bleue allumée à l'arrière du robot).



6. Aménagement de l'Environnement

6.1 Terrain



- Placer le robot comme indiqué sur l'image (bouton on/off du côté opposé au terrain)
- Placer le distributeur de jeton à 3 cm de la case centrale du terrain (à droite par rapport au robot)

6.2 Conditions idéales

S'assurer que la lumière de la pièce est allumée afin que la détection de formes soit bien réalisée.

7. Installation et Exécution

1. Télécharger Python

Suivez les instructions sur ce site en fonction de votre OS :

<https://kinsta.com/fr/base-de-connaissances/installer-python/#windows>

2. Ouvrez le **Terminal** (tapez cmd dans la barre de recherche pour Windows)

3. Installer les bibliothèques nécessaire avec les commandes :

- pip install pyniryo
- pip install opencv-python
- pip install numpy

4. Télécharger le dossier comprenant **morpion.py** et **minimax.py**
5. Ouvrir un terminal depuis ce dossier
6. Se connecter au réseau wi-fi Niryo Hotspot (mot de passe : niryorobot)
7. Mettre des pions carrés dans le distributeur de pions (l'utilisateur utilise les pions circulaire)
8. Lancer le programme **morpion.py** en rentrant dans le terminal :
 - `py .\morpion.py`

8. Tests et Résultats

Résultat majoritairement convaincant néanmoins quelque détails à prendre en compte :

- Possible que la détection de forme confonde cercle et carré si la luminosité n'est pas optimale (trop faible)
- Il faut éviter de réfléchir à son coups en plaçant sa main au niveau du terrain de jeu (peut reconnaître une forme qui n'a pas lieu d'être), il faut donc jouer rapidement (le robot détecte les changements toutes les 3 secondes, il faut donc jouer en moins de 3 secondes)

9. Evolutions Possibles

Améliorations possibles :

- Réglage automatique des paramètres de la caméra via IA afin d'être dans les meilleures conditions possible pour la détection de forme.
- Ajout d'un système de difficulté (facile, normal, difficile → difficulté actuelle)
- Optimisation possible de l'algorithme de minimax
- Afficher un état de la grille visuellement en temps réel
- Détection de triche

10. Annexe

Lien du github contenant les programmes : https://github.com/Dranelyh/Morpion_Ned2