

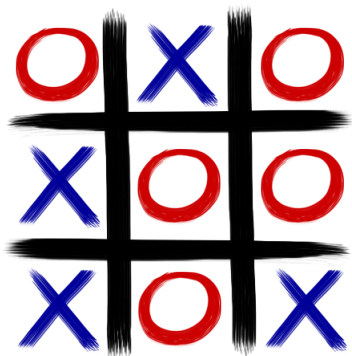
# Jouer au Tic Tac Toe avec Poppy Ergo Jr

---

March 1, 2016

- 1 Le projet Tic-Tac-Toe en ISN
  - Objectifs pédagogiques du projet
  - Matériels et logiciels
    - Processing : un outil java performant
  - Travail à réaliser
    - Déplacement du robot sur l'aire de jeu
    - Initiation à la vision par ordinateur
    - Communiquer avec Snap! depuis Processing
    - Jouer contre une intelligence artificielle
  - Conclusion

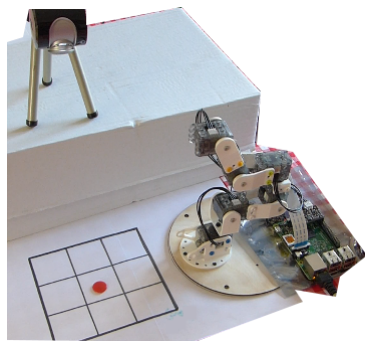
# Objectifs pédagogiques du projet (12 semaines)



- Théorie des jeux
- Introduction à la robotique
- Résolution et programmation d'un problème ouvert : Jouer au tic-tac-toe contre un robot (Ergo Jr)

# Matériels et logiciels

- Une créature Poppy Ergo Jr avec sa carte Raspberry
- Snap!
- Une webcam connectée à l'ordinateur
- Processing avec les modules OpenCV, net, video et Arrays



# Un outil java performant : Processing

Processing est un outil basé sur le langage Java et conçu par Benjamin Fry et Casey Reas du MIT (~2001) pour :

- la création dans le domaine des arts numériques,
- le multimédia : vidéo, son, électronique.



Processing est distribué avec une Creative Commons License offrant la possibilité d'être modifié mais à un usage non commercial.

# Travail à réaliser

- Construction d'un plateau de jeu en fonction des dimensions de l'Ergo Jr
- Implantation Snap! des déplacements du robot (par démonstration ou codage)
- Analyse de l'aire de jeu à l'aide de la webcam et Processing (Initiation à la vision par ordinateur)
- Transfert des données de Processing vers Snap!
- Programmation de la réponse du robot.

# Déplacement du robot sur l'aire de jeu

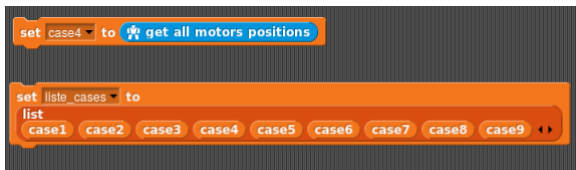
## Objectif:

Le robot doit être capable d'atteindre la case dans laquelle il veut jouer

# Déplacement du robot sur l'aire de jeu

## Prérequis :

- Savoir déclarer et affecter une variable
  - Savoir récupérer la position des moteurs
  - Savoir utiliser le bloc list
- 👉 Tous ses concepts ont été abordés dans l'activité : *Prise en main de Snap! Unit 1 : Contrôler Poppy avec les touches de son clavier*



The image shows two Snap! code blocks on a dark grey background. The first block is an orange 'set' block with 'case4' in the dropdown menu, followed by 'to' and a blue 'get all motors positions' block. The second block is an orange 'set' block with 'liste\_cases' in the dropdown menu, followed by 'to' and a 'list' block. The 'list' block contains ten orange buttons labeled 'case1' through 'case9', followed by a double arrow button.

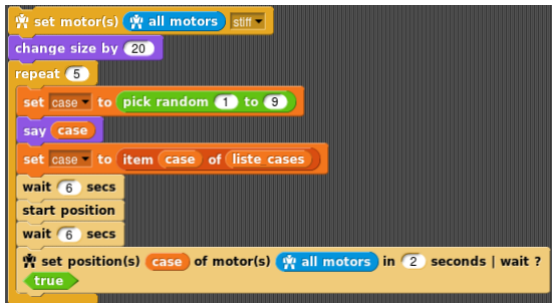


# Tester les différentes positions acquises

## Prérequis :

- Savoir contrôler le robot et faire bouger les moteurs
- Savoir utiliser les blocs **pick random** et **repeat**
- Savoir créer un bloc snap!

👉 Tous ces concepts ont été abordés dans l'activité : *Prise en main de Snap! Unit 1 : Contrôler Poppy avec les touches de son clavier*



```
set motor(s) all motors stiff
change size by 20
repeat 5
  set case to pick random 1 to 9
  say case
  set case to item case of liste cases
  wait 6 secs
  start position
  wait 6 secs
  set position(s) case of motor(s) all motors in 2 seconds | wait ?
  true
```

# Initiation à la vision par ordinateur

## Objectif:

Connecté à une caméra le robot doit être capable de repérer les positions des pions sur la grille de jeu.

# Initiation à la vision par ordinateur

La vision par ordinateur correspond à la partie du projet intitulée *analyse de l'aire de jeu (grille + pions)*.

L'analyse du flux vidéo en temps réel est réalisée avec une webcam, Processing et les modules :

- **video**
- **OpenCV** (OpenCV for Processing)

# Analyse de l'aire de jeu

## Traitements préliminaires de l'image

- Capture de l'image à analyser à partir du flux vidéo
- Conversion d'une image couleur en niveau de gris
- Binarisation de l'image avec seuillage à valeur fixe

## Prérequis :

- Utiliser l'API de Processing, video et OpenCV
- 👉 Pratique déjà abordée longuement en ISN

# Analyse de l'aire de jeu

Résolution d'un problème complexe : comment analyser la grille et la position des pions ?

- Détection des contours :
  - grille de jeu,
  - cases,
  - pions.
- Repérage sur la grille de jeu des positions des pions.
- Stocker l'information pour la transmettre à Snap!

# Communiquer avec Snap!

Réalisation d'un serveur web :



# Implantation Serveur HTTP avec Processing

```
1 void serverHTTP(String val) {
2   Client thisClient = myServer.available();
3   if (thisClient !=null) {
4     String whatClientSaid = thisClient.readString();
5     if (whatClientSaid != null) {
6       println(thisClient.ip() + "t" + whatClientSaid);
7       myServer.write("HTTP/1.0 200 OK\n");
8       myServer.write("Content-Type : text/html\n");
9       myServer.write("Access-Control-Allow-Origin: *\n\n");
10      myServer.write(val);
11      myServer.disconnect(thisClient);
12    }
13  }
14 }
```

Merci à Théo et Damien pour leurs explications sur les requêtes *cross domain*

# Tic-Tac-Toe et l'algorithme minimax

Tic-Tac-Toe est un jeu déterministe à information complète (comme les dames, les échecs, ...)

L'algorithme **minimax** permet de calculer le meilleur coup possible à jouer par l'ordinateur mais implantation complexe pour les élèves en raison des notions :

- d'arbre
- de récursivité



# Programmer l'aléatoire

Une première approche plus simple pour les élèves est de programmer une réponse aléatoire.

**Problématique** : Comment l'ordinateur peut-il produire des nombres censés constituer des nombres aléatoires, alors que son fonctionnement est en principe complètement déterministe?

- Méthode du carré médian
- Congruence linéaire

$$x_{n+1} = (a \times x_n + c) \% m$$

# Conclusion

Un projet très riche qui monopolise de nombreuses compétences :

- Introduction à la robotique
  - Montage d'un système de développement robotique (action et vision) : Poppy + Raspberry + Snap! + webcam
  - Interaction du robot avec l'environnement
- Programmation
  - Implantation des fonctionnalités avec Snap! et Processing (proche de Java)
  - Communication inter-application (ou inter-langage) à l'aide d'un serveur web
  - Initiation à l'analyse d'image et à la vision par ordinateur

Au final les élèves élaborent une nouvelle plate-forme (du robot qui voit clair) plus riche que celle proposée par Poppy Ergo Jr.