

# MotorDaemon Map Editor

## Cahier des Charges

### Fonctionnel v1.0

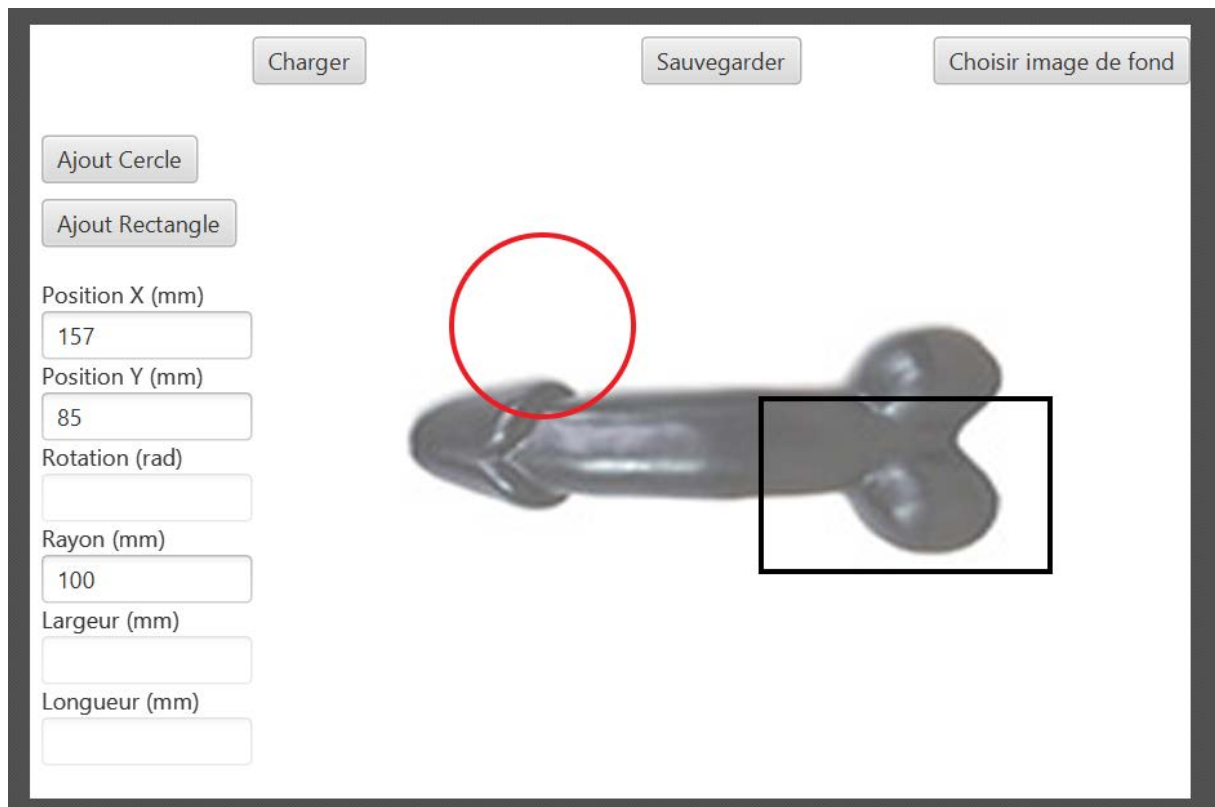


Figure 1 - Mockup de l'interface

## I- Fonctionnalités de l'interface

L'interface doit ressembler à la figure 1, pas dans le détail bien sûr, mais lorsque l'on sélectionne un obstacle, il doit changer de couleur et les informations à gauche doivent se mettre à jour. Les options inutiles, comme le rayon pour un rectangle doivent être rendues grisées (disable le textfield).

La map doit pouvoir être zoomée/dézoomée, et l'on doit avoir des sliders, exactement comme ARJL.

La position des obstacles dans l'interface doivent représenter leur position réelle. Il faudra pouvoir les drag&drop, ainsi que configurer leur position manuellement avec les champs à gauche.

Nous devons pouvoir afficher une image de fond pour aider au placement des obstacles. Il faudra faire une popup au moment de charger l'image pour pouvoir indiquer les dimensions réelles de la carte, afin de bien aligner les obstacles.

On suppose que le (0,0) se situe toujours au centre-bas de la carte.

## II- Sauvegarder/charger

La map doit être sauvegardée en JSON pur dans un fichier avec pour extension « .mdmap ». Il s'agira d'une liste d'objets JSON représentés comme ceci :

```
{
  "type" : "rectangle",
  "x" : 100,
  "y" : 150,
  "angle" : 0,
  "rayon" : null,
  "height" : 100,
  "width" : 100
}

{
  "type" : "circle",
  "x" : 100,
  "y" : 150,
  "angle" : null,
  "rayon" : 100,
  "height" : null,
  "width" : null
}
```

Pour plus d'infos sur les objets JSON : [https://www.w3schools.com/js/js\\_json\\_objects.asp](https://www.w3schools.com/js/js_json_objects.asp)

### III- Fonctionnalités à prévoir

Ceci traite des futures fonctionnalités si la première partie fonctionne correctement :

- Pouvoir ajouter un « home point » qui représente le point de départ du robot et celui vers lequel il retourne en cas de perte de connexion. A sauvegarder dans le JSON
- Pouvoir orienter les rectangles « à la main »
- Pouvoir spécifier des limites physiques de la map, c'est-à-dire spécifier le terrain de jeu du robot sous forme d'un polygone plus ou moins complexe (temporairement nous pouvons mettre des obstacles rectangulaires pour faire les côtés du polygone). A sauvegarder dans le JSON
- Positionnement drag&drop magnétique quand aligné avec un autre obstacle, exactement comme dans n'importe quel éditeur WYSIWYG.