

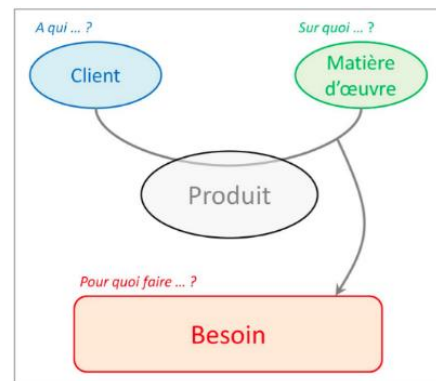
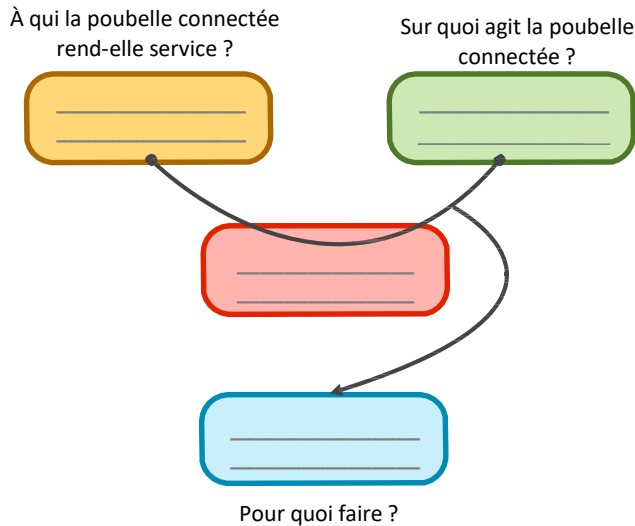


# Une poubelle connectée ?

NOM / Prénom : \_\_\_\_\_  
Classe : \_\_\_\_\_

## FORMULATION DU BESOIN

1. **Consigne** : Complétez le schéma appelé « bête à cornes » ci-dessous, en répondant aux questions posées. Utilisez comme ressources les **documents 1 et 2**.



Document 1

2. **Consigne** : Formulez le besoin sous la forme d'une phrase simple (*le besoin = fonction d'usage principale de l'objet*).

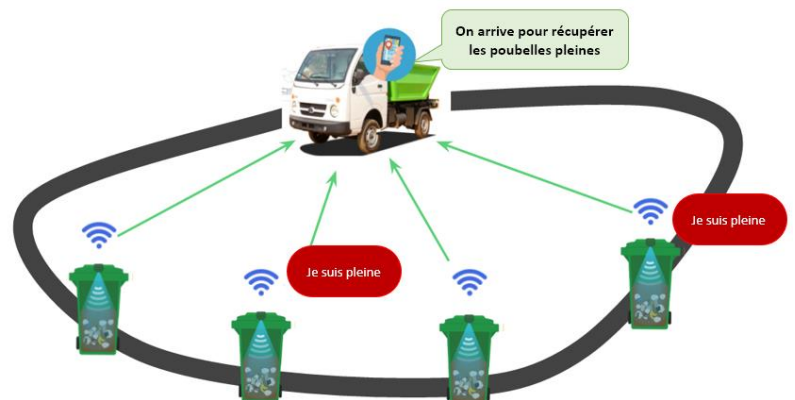
## Document 2 : Fonctionnement de la poubelle connectée.

La poubelle connectée e-cube permet de stocker, de compacter les déchets et d'alerter les éboueurs pour leur permettre de collecter les poubelles uniquement pleines.

Les déchets sont stockés dans le corps de la poubelle. La poubelle est alimentée par un ensemble de panneaux solaires et une batterie.

Le système équipé d'une carte de commande mesure et traite en permanence la hauteur de déchets stockée.

L'information de la hauteur de déchets est mesurée grâce à un capteur à ultrasons.



Quand cette hauteur est trop importante, la poubelle compacte les déchets à l'aide d'un ensemble composé d'un moteur et d'une vis sans fin. Le compactage s'effectue jusqu'à ce que la poubelle arrive à sa capacité maximale de stockage.

Une fois la capacité maximale atteinte, la poubelle signale d'abord par le réseau WIFI, ensuite à travers le réseau Internet son état à la société de récupération des déchets.

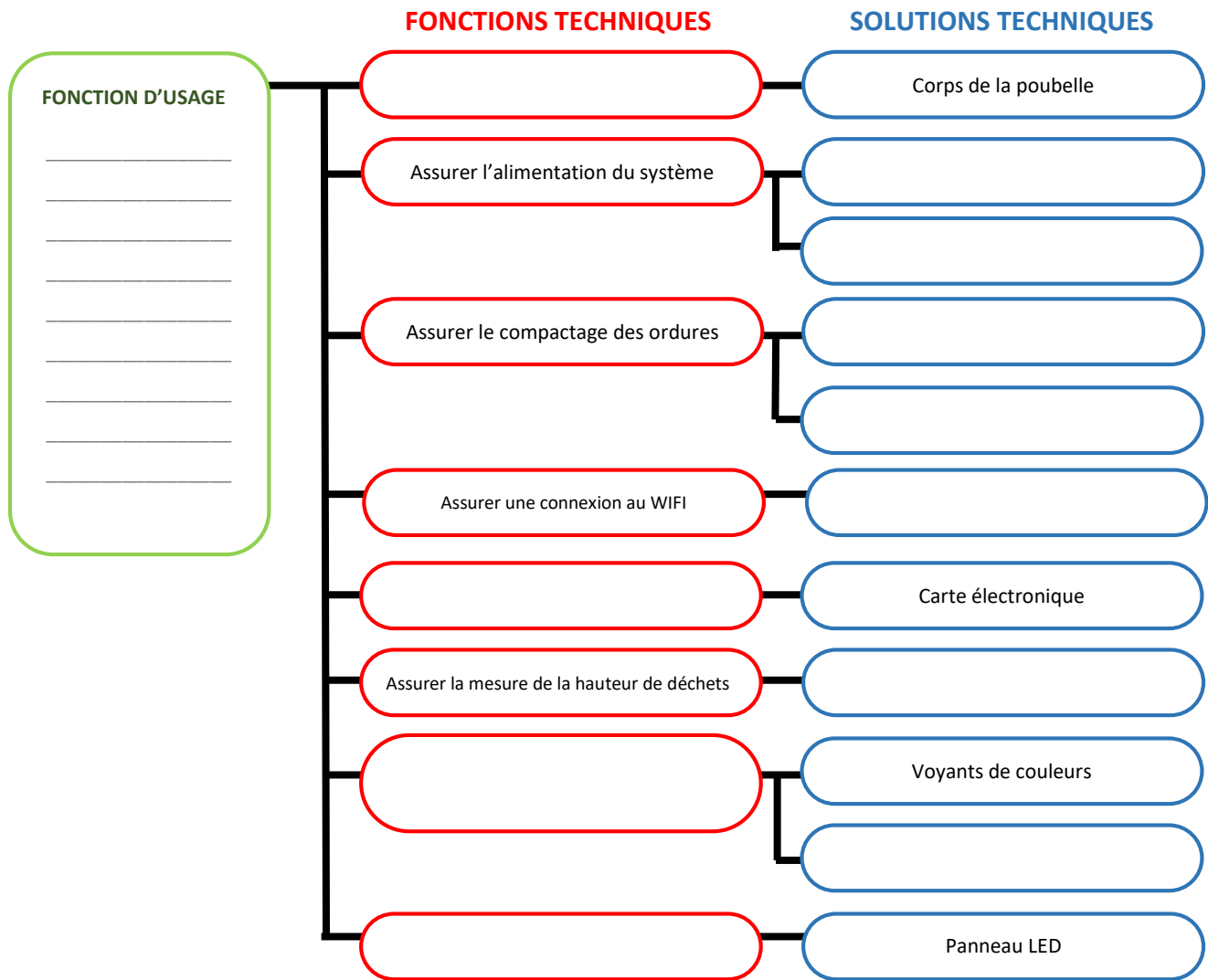
Une série de voyants de couleurs (vert, orange et rouge) permettent aussi de déterminer l'état de remplissage de la poubelle.

Comme autres fonctions, la poubelle peut servir aussi de borne WIFI pour les piétons ainsi que de panneaux publicitaires à l'aide d'un écran LCD à LED.

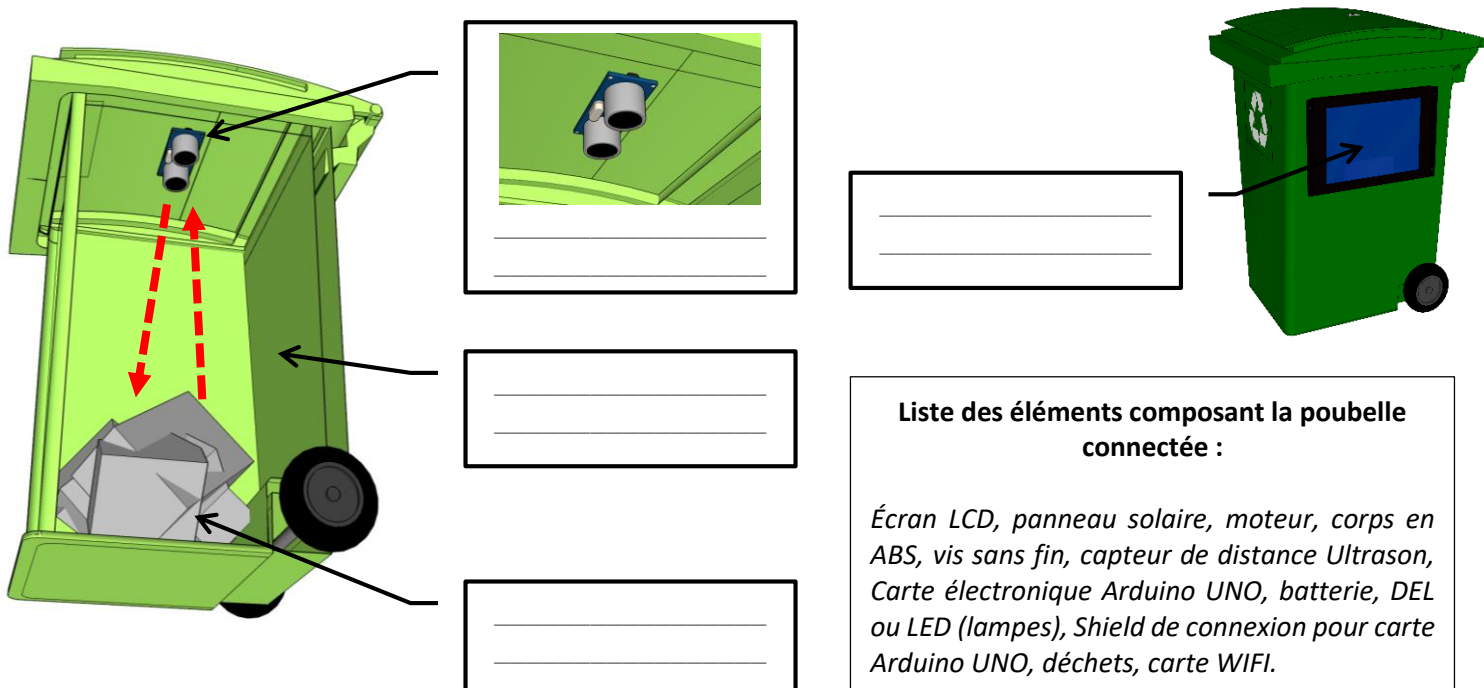


## ANALYSE FONCTIONNELLE : FONCTIONS TECHNIQUES ET SOLUTIONS ASSOCIÉES

1. **Consigne :** À l'aide du document 2, complétez le schéma ci-dessous en associant fonctions techniques et solutions techniques. N'oubliez pas la fonction d'usage.



2. **Consigne :** Compléter le schéma de la poubelle en associant les éléments à leurs noms respectifs.





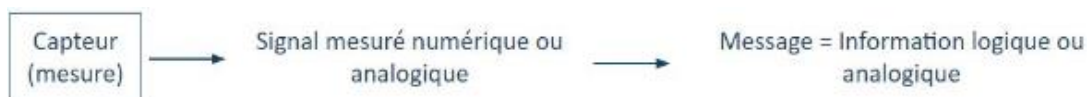
## ÉTUDE DU CAPTEUR DE DISTANCE : NATURE DE L'INFORMATION

**Consigne :** Utilisez le document ci-dessous ainsi que les repères (1), (2) et (3) sur la fiche pour compléter les 3 informations suivantes :

- 1** Nom de l'information : **hauteur de déchets.**  
Type d'information : ☐ logique ☐ analogique
- 2** Nom de l'information : **alerter les éboueurs par le réseau WIFI.**  
Type d'information : ☐ logique ☐ analogique  
Transport de l'information :
  - ☐ filaire/par conducteur [fil électrique, fibre optique, courant porteur de ligne CPL].
  - ☐ non-filaire [vibration, infra-rouge, onde radio : satellite, 4G, 5G, Bluetooth, wifi, Li-Fi].
- 3** Nom de l'information : **visualiser l'état de remplissage par des LED.**  
Type d'information : ☐ logique ☐ analogique  
Transport de l'information :
  - ☐ filaire/par conducteur [fil électrique, fibre optique, courant porteur de ligne CPL].
  - ☐ non-filaire [vibration, infra-rouge, onde radio : satellite, 4G, 5G, Bluetooth, wifi, Li-Fi].

### DOCUMENT DE TRAVAIL

Une **information** est un **message** que l'on souhaite transmettre. Pour **communiquer** une **information** sur un objet technique, on utilise un **signal** (mesuré par un capteur).



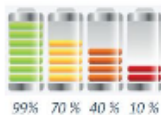
- L'information peut être **logique** ou **analogique**.

Une information logique ne prend que deux valeurs :  
présence/absence, ouvert/fermé, niveau atteint ou non...



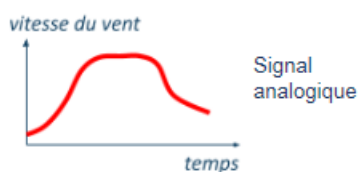
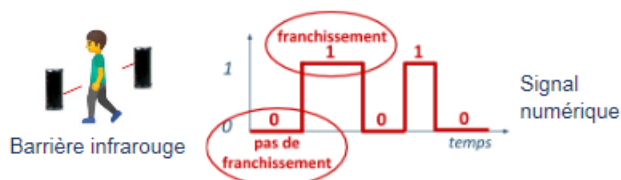
Aucune présence :  
**pas d'alarme.**  
Une présence :  
**alarme.**

Une information analogique prend une infinité de valeurs, qui varie de manière continue dans le temps : température, luminosité, distance...



La batterie est en cours de chargement, le pourcentage de charge **varie dans le temps.**

- Pour déterminer l'**information transmise**, on repère le **signal** mesuré par le **capteur**.



En résumé, deux types d'informations :

- Une **information logique** prend **deux valeurs** : 0 ou 1, ouvert/fermé, lumière/pas de lumière, présence/pas présence, ...
- Une **information analogique** prend une **infinité de valeurs** et **varie dans le temps de façon continue** (température, distance...).

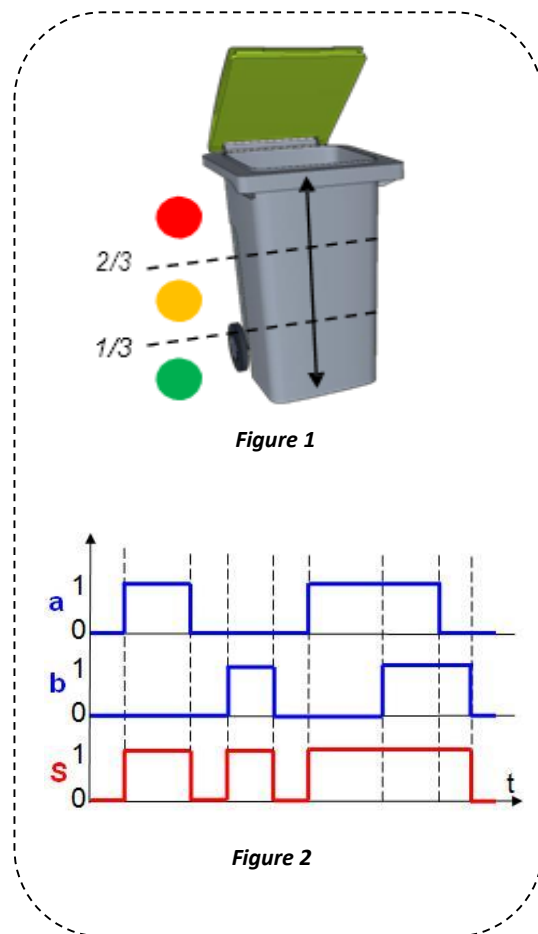
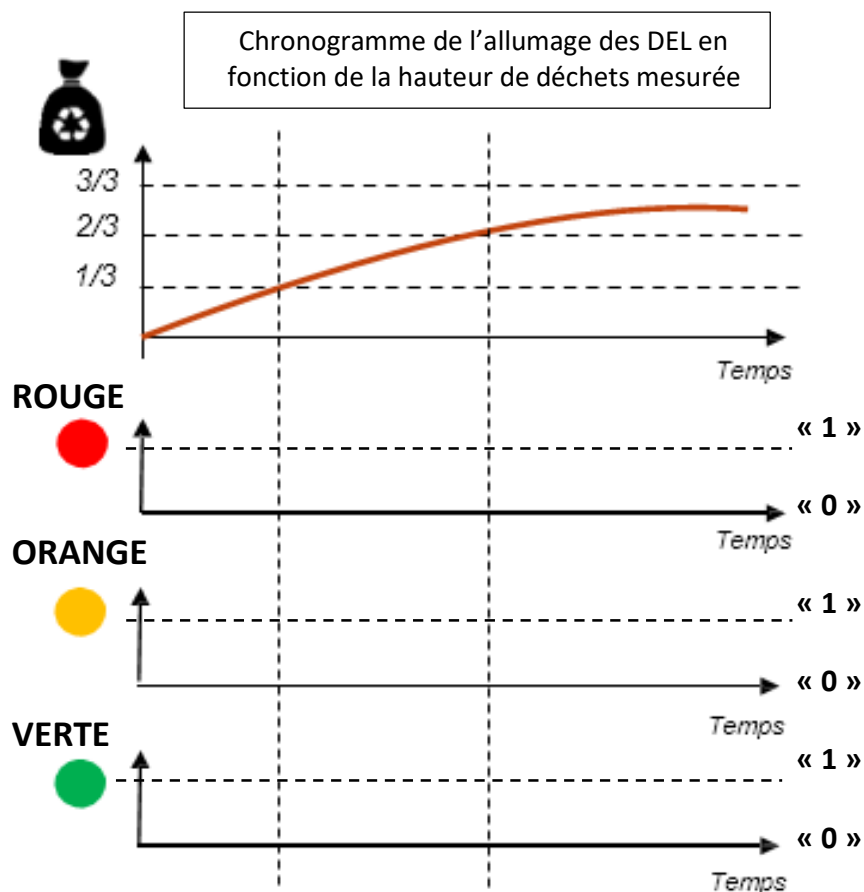
## DÉCRIRE LE FONCTIONNEMENT DE LA POUBELLE CONNECTÉE : LES ALGORITHMES

Seule la gestion de la communication est à traiter. La partie compactage des ordures n'est donc pas à gérer. Pour simplifier le fonctionnement, seuls 2 seuils de hauteur dans la poubelle sont en prendre en compte (comme indiqués sur la figure 1).

### Consigne :

Compléter le chronogramme ci-dessous en utilisant :

- les 2 seuils de hauteur des déchets (1/3 et 2/3) et,
- l'allumage des diodes en sachant que :  
« 0 » = voyant éteint (0 Volt aux bornes de la diode) – « 1 » = voyant allumé (5 Volts aux bornes de la diode) – voir l'exemple figure 2.



### Consigne :

Récapituler dans les deux colonnes la liste des évènements (2 au total) et des actions (5) qui vous servir à l'élaboration des algorithmes :

ÉVÈNEMENTS (2)*
.....
.....
.....

ACTIONS (5)**
« allumer le voyant rouge »
.....
.....
.....
.....

\* : un évènement correspond à une question à laquelle on peut répondre uniquement par : OUI ou NON

\*\* : une action est réalisée par un actionneur.



## FICHE DE PROGRAMMATION SOUS MBLOCK

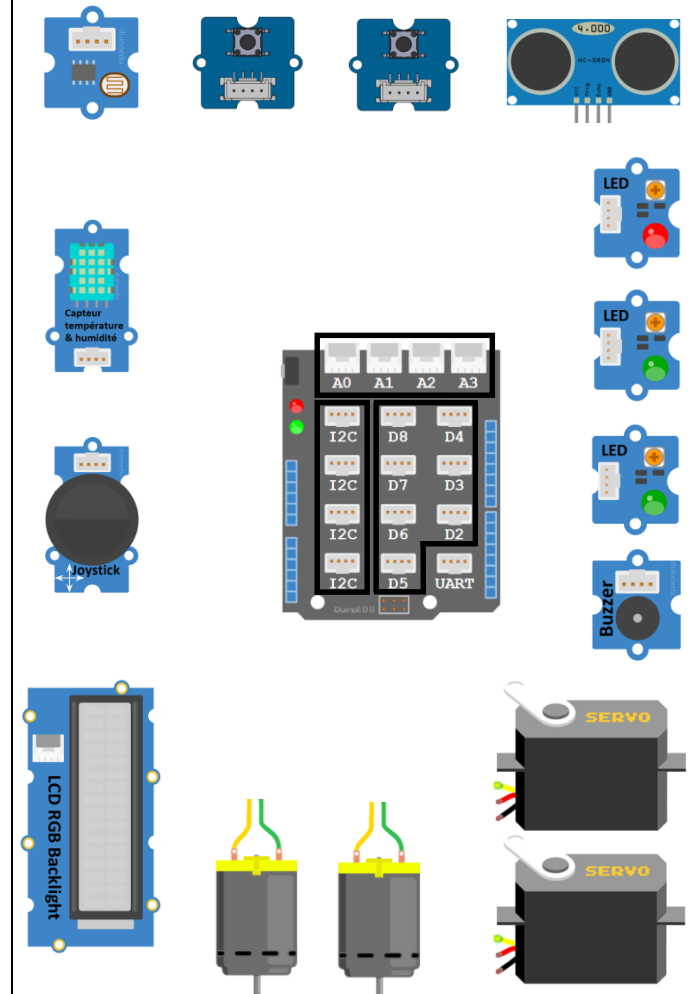
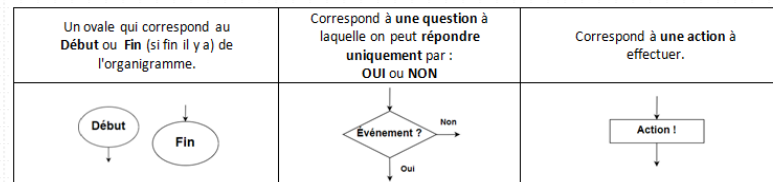


Algorithme sous la forme d'un texte



Algorithme sous la forme d'un organigramme de programmation

Schéma de câblage & programmation mBlock

- Si ... alors...
- Si ... alors... sinon... / répéter...jusqu'à / répéter indéfiniment
- Créer une variable / Stocker dans une variable / initialiser une variable.
- =, >, <, +, -, x, /, et, ou











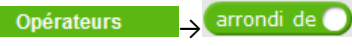


Entourer et relier à la carte Arduino **UNO** : en vert les actionneurs et en rouge les capteurs utilisés pour compléter l'objectif.

**Objectif(s) :** Avec les flèches du clavier  et , le sac se déplace verticalement en direction du capteur US. Les diodes de couleurs s'allument selon le taux de remplissage de la poubelle en suivant le tableau de concordance simplifié :


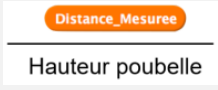
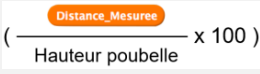
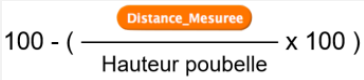

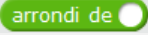







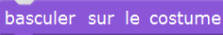
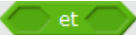
Taux de remplissage	Diode allumée	Diodes éteintes
< 33 %	Verte	Orange - Rouge
Entre 33% et 90 %	Orange	Verte - Rouge
> à 90 %	Rouge	Verte - Orange

### Consignes :

En utilisant l'algorithme de la page 6, réaliser la simulation du fonctionnement de la poubelle sur mBlock en suivant les étapes suivantes :

Étape 1	Télécharger la structure du programme et ouvrez le dans mBlock (ou Scratch)	
Étape 2	<p align="center"><b>Déplacement du sac poubelle</b></p> <p>On travaille sur le lutin/sprite du sac poubelle : .</p> <p><b>Le sac poubelle représentera la hauteur de déchets.</b></p> <p><b>Objectifs :</b> Faire se déplacer la poubelle et simuler la variation de la hauteur des déchets.</p> <p><b>Travail à faire :</b></p> <ol style="list-style-type: none"> <li>Quand les touches  et  du clavier, la poubelle doit se déplacer verticalement (axe y).</li> <li>Quand on presse la touche « espace », le sac poubelle revient à sa position d'origine.</li> </ol> <p><b>Aide :</b></p> <ul style="list-style-type: none"> <li>Position d'origine du sac poubelle : .</li> </ul> <p><b>Synthèse :</b></p> <p>Compléter les phrases suivantes :</p> <ul style="list-style-type: none"> <li>Quand j'appuie sur la <b>flèche</b> , la hauteur de déchets .....</li> <li>Quand j'appuie sur la <b>flèche</b> , la hauteur de déchets .....</li> <li>Quand j'appuie sur la touche « <b>espace</b> », la hauteur de déchets .....</li> </ul>	/ 3 points
Étape 3	<p align="center"><b>Calculer une distance en pixels</b></p> <p>On travaille sur le lutin/sprite du capteur : .</p> <p><b>Objectifs :</b> Calculer la distance (en pixels) entre le lutin du capteur et le lutin du sac poubelle.</p> <p><b>Travail à faire :</b></p> <ol style="list-style-type: none"> <li>Créer une variable « Distance_mesuree » pour tous les lutins.</li> <li>Dans cette « Distance_mesuree » placer : <ul style="list-style-type: none"> <li>un bloc pour mesurer la distance : ,</li> <li>un bloc pour arrondir le résultat .</li> </ul> </li> </ol> <p><b>Synthèse :</b></p> <p>Compléter les phrases suivantes :</p> <ul style="list-style-type: none"> <li>Quand j'appuie sur la <b>flèche</b> , la variable « Distance_Mesuree » .....</li> <li>Quand j'appuie sur la <b>flèche</b> , la variable « Distance_Mesuree » .....</li> <li>Quand j'appuie sur la touche « <b>espace</b> », la variable « Distance_Mesuree » affiche la valeur : .....</li> </ul>	/ 5 points



	<p style="text-align: center;"><b><u>Convertir la distance en pixels en taux</u></b></p> <p>On travaille sur le lutin/sprite du capteur : .</p> <p><b>Objectifs :</b> On souhaite convertir la distance mesurée à l'étape 2 en taux de remplissage exprimé en pourcentage (0 &lt; Taux &lt; 100%).</p> <p>Pour cela nous allons utiliser une formule :</p> <div style="border: 1px solid red; padding: 10px; margin: 10px auto; width: fit-content;"> <math display="block">\text{Taux\_Remplissage} = 100 - \left( \frac{\text{Distance\_Mesuree}}{\text{Hauteur poubelle}} \times 100 \right)</math> </div> <p><b>Travail à faire :</b></p> <ol style="list-style-type: none"> <li>Créer une variable « Taux_Remplissage ».</li> <li>Dans ce « Taux_Remplissage » : <ul style="list-style-type: none"> <li>Utiliser un bloc opérateur pour réaliser l'opération :  , la hauteur de la poubelle est de 140 pixels.</li> <li>Associer un nouveau bloc opérateur pour effectuer  . Attention à imbriquer les blocs dans le bon ordre.</li> <li>Associer un nouveau bloc opérateur pour finir la formule  . Attention à imbriquer les blocs dans le bon ordre encore une fois.</li> <li>Ajouter un bloc pour arrondir le résultat  → .</li> </ul> </li> </ol> <p><b>Synthèse :</b> Compléter les phrases suivantes :</p> <ul style="list-style-type: none"> <li>Quand j'appuie sur la <b>flèche</b> , la variable « Distance_Mesuree » ..... et le taux de remplissage .....</li> <li>Quand j'appuie sur la <b>flèche</b> , la variable « Distance_Mesuree » ..... et le taux de remplissage .....</li> <li>Quand j'appuie sur la touche « <b>espace</b> », la variable « Distance_Mesuree » affiche la valeur : ..... et le taux de remplissage affiche la valeur .....</li> </ul>	/ 5 points												
<b>Étape 4</b>	<p style="text-align: center;"><b><u>Allumer la diode en fonction du taux de remplissage</u></b></p> <p style="text-align: center;">  </p> <p>On travaille sur le lutin/sprite du capteur : Del_Vert... Del_Jau... Del_Ro...</p> <p><b>Objectifs :</b> Pour chaque diode, appliquer le tableau de concordance du système et vérifier la simulation.</p> <table border="1" style="margin: 10px auto; width: 80%;"> <thead> <tr> <th>Taux de remplissage</th><th>Diode allumée</th><th>Diodes éteintes</th></tr> </thead> <tbody> <tr> <td>&lt; 33%</td><td>Verte</td><td>Orange et Rouge</td></tr> <tr> <td>Entre 33% et 90%</td><td>Orange</td><td>Verte et Rouge</td></tr> <tr> <td>&gt; à 90%</td><td>Rouge</td><td>Verte et Orange</td></tr> </tbody> </table> <p>Pour la diode Verte et Rouge : Si...  est &lt;= à .....% alors ... sinon ...</p> <p>Pour allumer ou éteindre :  Del_Verte_ON ▼ ou  Del_Verte_OFF ▼</p> <p>Pour la diode Orange : Utiliser l'opérateur  pour gérer la fourchette « &gt; 33% <u>et</u> &lt;90% ».</p>	Taux de remplissage	Diode allumée	Diodes éteintes	< 33%	Verte	Orange et Rouge	Entre 33% et 90%	Orange	Verte et Rouge	> à 90%	Rouge	Verte et Orange	/ 7 points
Taux de remplissage	Diode allumée	Diodes éteintes												
< 33%	Verte	Orange et Rouge												
Entre 33% et 90%	Orange	Verte et Rouge												
> à 90%	Rouge	Verte et Orange												
<b>Étape 5</b>	<p>Tester votre programme et vérifier son bon fonctionnement. L'envoyer par la messagerie de l'ENT.</p>													