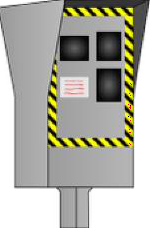
# EXERCICE 1

**Remplir** l’algorigramme permettant le fonctionnement connu d’un radar de vitesse automatique

: Un flash apparait dès la détection d’un véhicule à une vitesse plus grande que 90 km/h afin de  
prendre la photo de la plaque pour l’envoyer au central.

EVENEMENTS ACTIONS

|  |  |
| --- | --- |
| Détection véhicule  avec une vitesse > 90km/h ? | Lecture plaque |
|  | Activer Flash |
| Envoyer information au central |



Début

Détection véhicule avec vitesse > 90 km/h ?

Envoyer info. au central

Activer Flash

Lecture plaque

# EXERCICE 2

**Définir les évènements et les actions** puis **remplir** l’algorigramme permettant, dès la détection d'une personne la mise en marche d'un escalator de bas en haut.

Afin de limiter la consommation d'énergie, l'escalator est initialement (au début) à l'arrêt et ne fonctionne pas tant qu'une personne n'est pas détectée.

Il faut 1 minute à l'escalator pour monter une personne.



Bas

EVENEMENTS ACTIONS

|  |  |
| --- | --- |
| Détection bas ? | Activer Escalator |
|  | Désactiver  Escalator |
| Attendre 1  minute |

Détection

# Initialisation



Début

Désactiver Escalator

Détection bas ?

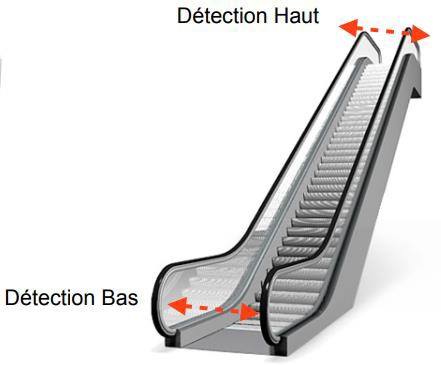
Désactiver Escalator

Attendre 1 minute

Activer Escalator

**EXERCICE 3**

Suite de l’exercice précédent.

En montant les marches, la personne qui utilise l’escalator mettra moins de temps pour arriver en haut. Inutile donc dans ce cas de faire fonctionner l’escalator durant toute une minute.

Pour des soucis d’économie d’énergie propose un autre algorigramme permettant le même principe, mais cette fois-ci, l'escalator devra s'arrêter uniquement lorsque la personne qui monte est détectée en haut de l'escalator.

EVENEMENTS ACTIONS

|  |  |
| --- | --- |
| Détection bas ? | Activer Escalator |
| Détection haut ? | Désactiver Escalator |



Début

Détection haut ?

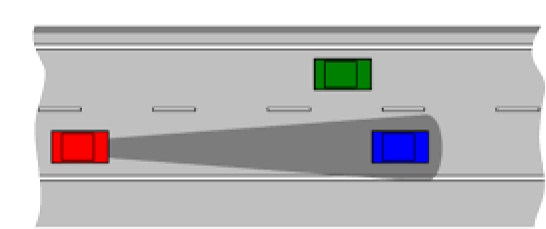
Détection bas ?

Désactiver Escalator

Activer Escalator



# EXERCICE 4

Imaginons un système embarqué dans une voiture capable d’évaluer si la distance entre la voiture et celle de devant est dangereuse. Si la distance devient dangereuse un voyant vert passe au rouge. La sécurité routière impose une distance de sécurité de 2 bandes blanches

EVENEMENTS ACTIONS

Début

|  |  |
| --- | --- |
| Variable\_distance < 2 bandes blanches ? | Enregistrer mesure dans « Variable\_distance  » |
|  | Mesure distance |
| Allumer voyant Vert |
| Allumer voyant Rouge |

# FACULTATIF

|  |  |
| --- | --- |
|  |  |
| Voyant Rouge | |
|  |  |
| Mesure distance | |
|  |  |
| Enregistrer mesure dans Variable\_distance | |
|  |  |

**Mais, permet de vérifier si le voyant fonctionne**

**correctement**

Variable\_distance < 2 bandes blanches ?

Voyant Rouge

Voyant vert

**EXERCICE 5**

Une télévision peut se mettre en veille depuis la télécommande. Mais les nouvelles télévisions se mettent en veille automatiquement après une utilisation de 3h sans intervention sur la télécommande. Propose la description de ce fonctionnement.

EVENEMENTS ACTIONS

|  |  |
| --- | --- |
| Bouton veille télécommande  activé ? | Mettre en veille la télévision |
| Aucune intervention  depuis 3h ? |



Début

Mettre en veille la télévision

Aucune intervention depuis 3h ?

télécommande

activé ?

veille

Bouton

Début

Compteur\_TPS = 0

Un chronomètre est ajouté à la télévision. Compléter les cases ajoutées avec des fonctions temporelles.



Mettre en veille la télévision

télécommande activé ?

veille

Bouton

Attendre 1 seconde

Compteur\_TPS =

Compteur\_TPS+1

Compteur\_TPS = 0

Aucune intervention depuis 3h ?

# EXERCICE 6

Propose une description définissant le fonctionnement

simple d’un portique de métro :

L’utilisateur glisse son ticket dans le lecteur, si le ticket est valide la porte s’ouvre et se referme que lorsque l’utilisateur est passé. Dans le cas contraire la porte reste fermée.

EVENEMENTS ACTIONS



|  |  |
| --- | --- |
| Présence de ticket  ? | Ouvrir portique |
| Détection présence après le portique ? | Fermer portique |
| Ticket valide ? | Rendre ticket |

Début

**Réalisez l’algorigramme**

Présence de ticket ?

Ticket valide ?

Ouvrir portique

Rendre ticket

Rendre ticket

Détection présence après le portique ?

Fermer portique

# EXERCICE6

Voici un distributeur de boisson : Le principe est simple, vous sélectionnez la boisson de votre choix, le prix s'affiche (ici 1€ pour toutes les boissons), vous insérez donc une pièce de 1 € (la machine vérifie la pièce) puis le distributeur vous donne la boisson. Dans le cas où la pièce n’est pas 1€ (0,10€, 0,20€, 0,50€, 2€ ou autre ...) le distributeur rejette la pièce.

**Réaliser les évènements et les actions** puis créer l’algorigramme

Merci de décrire exclusivement ce fonctionnement !

EVENEMENTS ACTIONS

|  |  |
| --- | --- |
| Pièce = 1 € ? | Distribuer la boisson |
| Boisson sélectionnée ? | Rejeter la pièce |
|  | Afficher le prix : 1 € |



Début

Boisson sélectionnée ?

=1€ ?

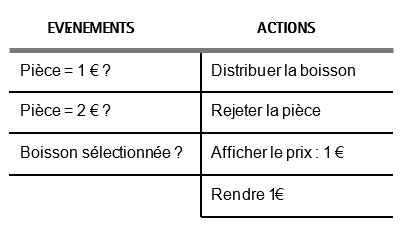
Rejeter la pièce

Distribuer la boisson

Pièce

Afficher le prix

Le problème avec le fonctionnement précédent, est l’impossibilité d’obtenir une boisson si nous n’avons pas de pièce de 1€. Modifier l’algorigramme pour avoir la possibilité d’obtenir une boisson avec une pièce de 1€ ou de 2€.



**Réaliser les évènements et les actions**

puis créer l’algorigramme

Début

Boisson sélectionnée ?

=1€?

Pièce=2€?

Distribuer la boisson

Rejeter la pièce

Pièce

Afficher le prix

Rendre 1€

TD rédigé par N. Tourreau / P. Pujades - Nov. 2017 - Correction exercices de logique Niv. 2