CYCLE 4 TECHNOLOGIE

L'informatique et la programmation

Algorithmique et codage au collège

Un algorithme est une suite finie et non-ambiguë d'instructions permettant de donner la réponse à un problème.

Le *langage de programmation* est l'intermédiaire entre l'humain et la machine, il permet d'écrire dans un langage proche de la machine, mais intelligible par l'humain, les opérations que l'ordinateur ou le système doit effectuer.

L'algorithme est codé dans un **langage de programmation graphique** spécifique, il s'agit de la phase de **programmation** au collège

Scratch : un environnement de programmation très complet 1/3

Notions de base

Un langage de programmation permet de décrire d'une part les structures des données qui seront manipulées par l'appareil informatique et d'autre part quelles seront les manipulations. Il offre un ensemble de notions telles que les instructions, les variables, les types, et les procédures, qui peuvent être utilisées pour traduire des algorithmes.

Une *instruction*

Un ordre donné à un ordinateur.

Une *variable*

Un nom utilisé dans un programme pour faire référence à une donnée manipulée par programme.

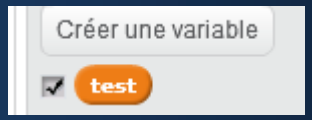
Une *constante*

Un nom utilisé pour faire référence à une valeur permanente.

Une expression *littérale*

Une valeur mentionnée en toutes lettres dans le programme. **Fonction : y=2x+3**

Dans Scratch 2.0



x est une variable à créer.

Scratch : un environnement de programmation très complet 2/3

Notions de base

Un *type*

Les types de données primitifs courants sont les nombres entiers, les nombres réels, le booléen, les chaînes de caractères et les pointeurs.

Valeur de type booléenne

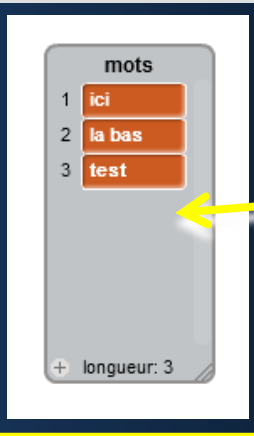
Plus précisément, le type booléen est un type qui n'a que deux valeurs, *vrai* et *faux*, tandis que le type pointeur : une **référence** à une donnée, qui se trouve quelque part en mémoire.

Pointeurs possibles en définissant des variables

Une structure de données

Pointeur possible en définissant une variable

Une manière caractéristique d'organiser un ensemble de données en mémoire, qui influe sur les algorithmes utilisés pour les manipuler. Les structures courantes sont les tableaux, les enregistrements, les listes, les piles et les arbres.



Dans Scratch 2 on ne peut que créer des listes.

Scratch : un environnement de programmation très complet 3/3

Notions de base

Une *déclaration*

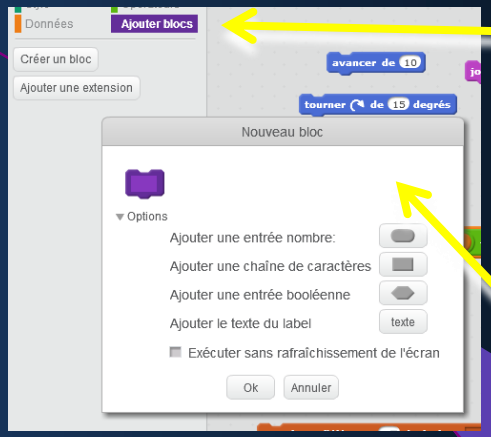
Une phrase de programme qui sert à renseigner au traducteur (compilateur, interpréteur, ...) les noms et les caractéristiques des éléments du programme tels que des variables, des procédures, des types..

Les *procédures*, *fonctions*, *méthodes*

Divers langages de programmation offrent la possibilité d'isoler un fragment de programme, et d'en faire une opération générale, paramétrable, susceptible d'être utilisée de façon répétée. Ces fragments sont appelés *procédures*, *fonctions* ou *méthodes*.

Les *modules*

Les langages de programmation peuvent également offrir la possibilité de découper un programme en plusieurs pièces appelées *modules*, chacune ayant un rôle déterminé, puis de combiner les pièces. Les notions de *procédure* et de *module* sont destinées à faciliter la création de programmes complexes et volumineux en assistant la prise en charge de cette complexité. Ces fonctions permettent en particulier la modularité et l'abstraction.

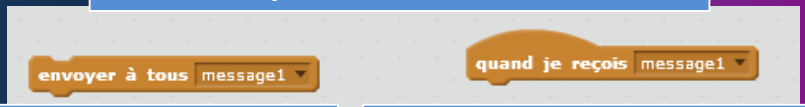


Les modules sont paramétrables par des données entrées et utilisées par la fonction.

Dans scratch 2.0 il n'ya pas de distinction entre procédures, fonctions et modules.

Le module test est défini puis utilisé comme une simple instruction.

Une autre façon de définir des modules



Programme principal

Module ou sous programme

Tutoriel de prise en main du logiciel Scratch 2

Mathématiques

Connaissances et compétences associées	Exemples de situations, d'activités et de ressources pour l'élève
Décomposer un problème en sous-problèmes afin de structurer un programme ; reconnaître des schémas. Écrire, mettre au point (tester, corriger) et exécuter un programme en réponse à un problème donné. Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs. Programmer des scripts se déroulant en parallèle. » Notions d'algorithme et de programme. » Notion de variable informatique. » Déclenchement d'une action par un évènement, séquences d'instructions, boucles, instructions conditionnelles.	Jeux dans un labyrinthe, jeu de Pong, bataille navale, jeu de nim, tic tac toe. Réalisation de figure à l'aide d'un logiciel de programmation pour consolider les notions de longueur et d'angle. Initiation au chiffrement (Morse, chiffre de César, code ASCII…). Construction de tables de conjugaison, de pluriels, jeu du cadavre exquis… Calculs simples de calendrier. Calculs de répertoire (recherche, recherche inversée...). Calculs de fréquences d'apparition de chaque lettre dans un texte pour distinguer sa langue d'origine : français, anglais, italien, etc.

Thème E - Algorithmique et programmation

Technologie

▶ CYCLE 4 TECHNOLOGIE

Écrire, mettre au point et exécuter un programme

Analyser le comportement attendu d'un système réel et décomposer le problème posé en sous-problèmes afin de structurer un programme de commande. Écrire, mettre au point (tester, corriger) et exécuter un programme commandant un système réel et vérifier le comportement attendu. Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs. » Notions d'algorithme et de programme. » Notion de variable informatique. » Déclenchement d'une action par un évènement, séquences d'instructions, boucles, instructions conditionnelles. » Systèmes embarqués. » Forme et transmission du signal. » Capteur, actionneur, interface.	Concevoir, paramétrer, programmer des applications informatiques pour des appareils nomades. Observer et décrire le comportement d'un robot ou d'un système embarqué. En décrire les éléments de sa programmation Agencer un robot (capteurs, actionneurs) pour répondre à une activité et un programme donnés. Écrire, à partir d'un cahier des charges de fonctionnement, un programme afin de commander un système ou un système programmable de la vie courante, identifier les variables d'entrée et de sortie. Modifier un programme existant dans un système technique, afin d'améliorer son comportement, ses performances pour mieux répondre à une problématique donnée. Les moyens utilisés sont des systèmes pluri-technologiques réels didactisés ou non, dont la programmation est pilotée par ordinateur ou une tablette numérique. Ils peuvent être complétés par l'usage de modélisation numérique permettant des simulations et des modifications du comportement.

▶ CYCLE 4 TECHNOLOGIE

L'informatique et la programmation

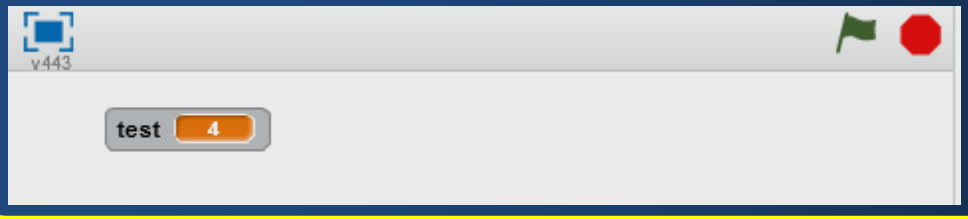
« La notion de variable informatique »

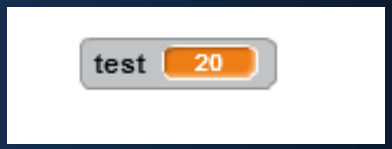
On peut considérer qu'une variable se comporte comme une **mémoire** capable de stocker une donnée : un nombre, une chaîne de caractères.

Des commandes ou opérations sont possibles autour de la variable définie pour en modifier son contenu.

Glisser le calcul dans la variable

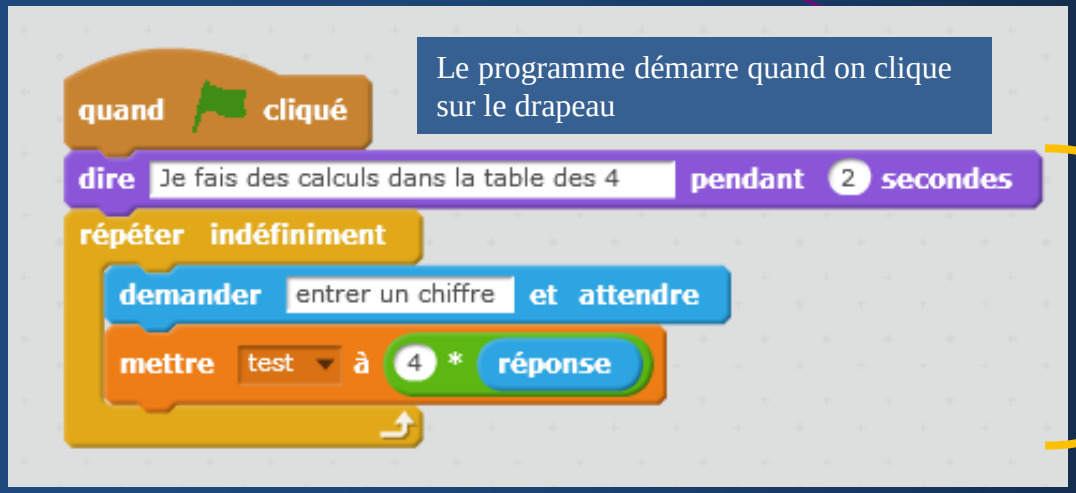
Compléter les opérandes de la multiplication





Cliquer sur la variable pour effectuer le calcul directement.

« Séquence d'instructions, boucles »



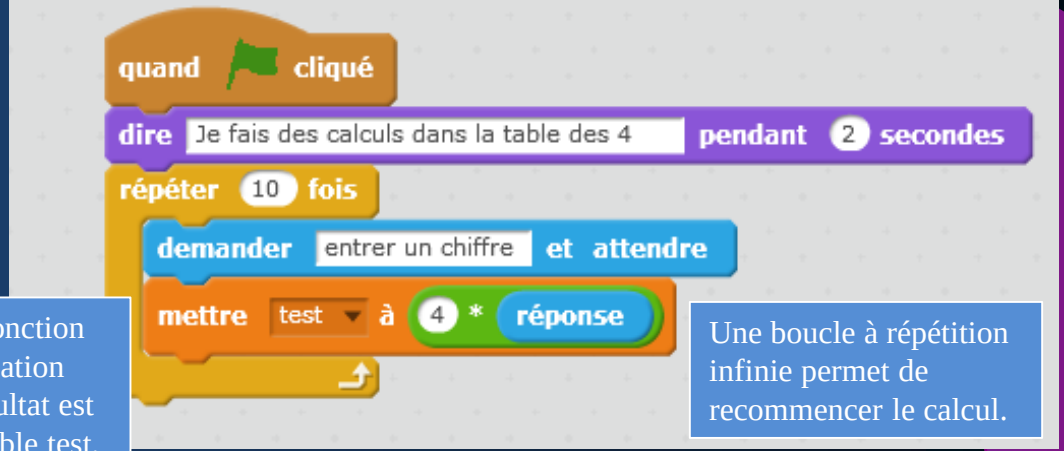
Le programme démarre quand on clique sur le drapeau

La séquence d'instruction comporte une boucle à répétition infinie.

Le chat « parle » et délivre son message

Le nombre saisi est affecté dans la variable réponse

La formule ou fonction est une multiplication par 4 dont le résultat est affecté à la variable test.



Une boucle à répétition infinie permet de recommencer le calcul.

En « détachant » les instructions, on remplace la boucle à répétition infinie

Variante : Le programme répète le calcul un nombre fini de fois et s'arrête.

« *Séquence d'instructions, boucles, instruction conditionnelle* »

L'objet chat précise qu'il s'agit d'un jeu.

La variable coups est mise à 0 avant de commencer la partie.

A la variable nombre, on affecte un nombre aléatoire compris entre 1 et 100.

L'utilisateur du jeu doit saisir un nombre

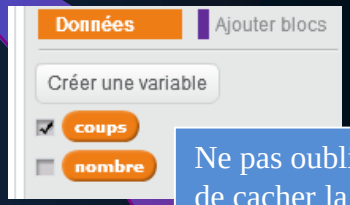
Le programme s'arrête, sort de la boucle infinie si le bon nombre est trouvé.

L'opération « regroupe » permet de mettre les informations les unes à la suite des autres, on dit aussi concaténées.

Scratch nous permet de créer des jeux simples et mathématiques.

On commence par créer deux variables : nombre et coups.

On commence par créer deux variables : nombre et coups



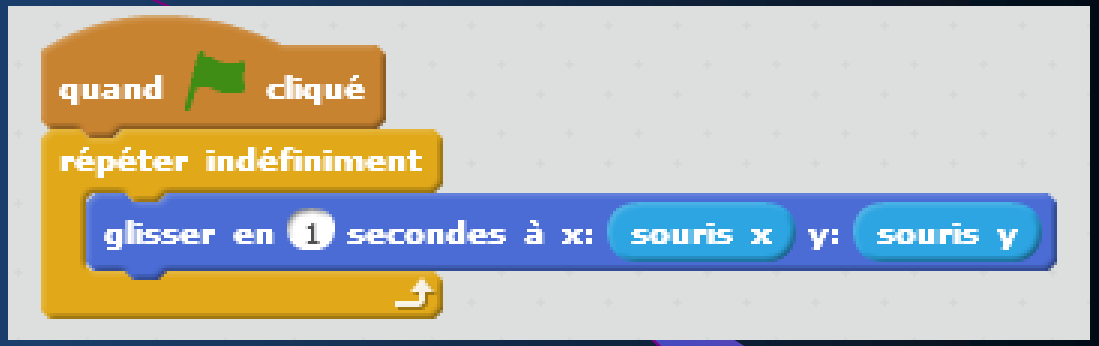
Ne pas oublier de cacher la variable nombre.

Trois instructions conditionnelles ou tests permettent de renseigner le joueur pour qu'il affine sa sélection.

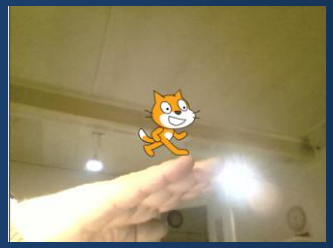
Jeu du nombre mystère

«Ecrire un programme dans lequel des actions sont déclenchées par des évènements extérieurs. »

Ce programme simple va permettre de déplacer l'objet chat à l'écran en fonction des appuis sur les touches de direction du clavier.



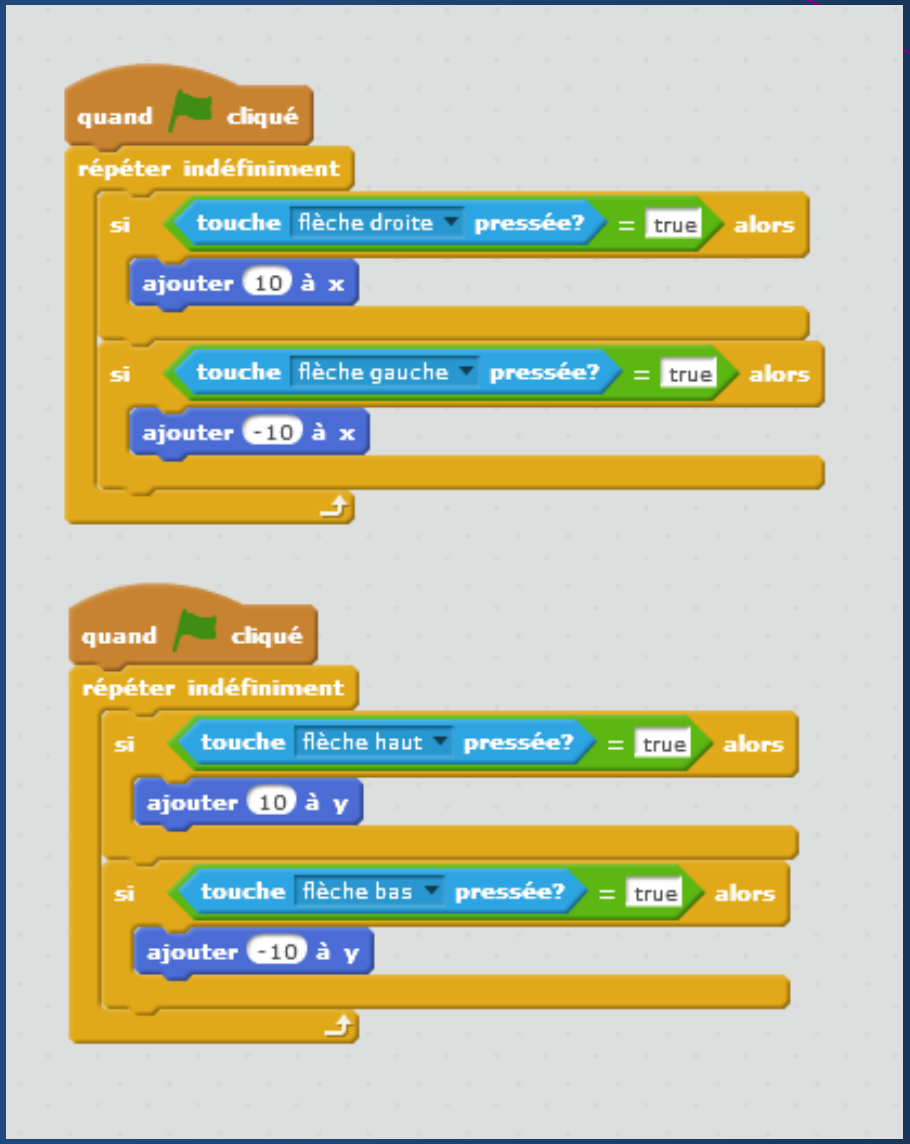
Cet autre programme permet à l'objet chat de se rapprocher du pointeur de la souris avec un délai d'attente de 1 seconde.



Celui-ci va permettre de faire bondir le chat en fonction des mouvements de la main devant la camera.

C'est le pourcentage de vitesse qui est détecté : mouvement rapide, 100 %

«Programmer des scripts se déroulants en parallèle»



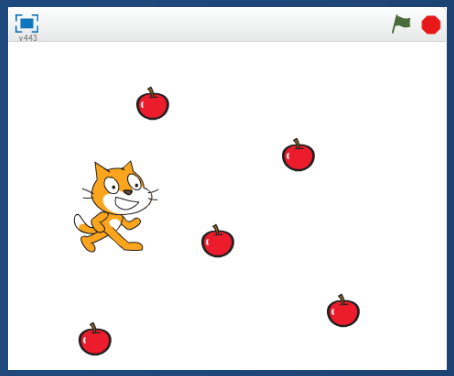
Dans ce programme, deux boucles ont été créees. Elles démarrent en même temps en fonction du clic sur le drapeau vert qui représente le lancement de l'ensemble des instructions. Les instructions visent à contrôler les mouvements de l'objet chat à partir d'appuis sur les touches de direction.

Les deux boucles fonctionnent indépendamment l'une de l'autre, les instructions s'exécutent en parallèle.

Scratch 2 dispose de capacités multitâches réelles. On peut les exploiter dans différents contextes.

Remarque : on peut tester chaque boucle de façon indépendante, ce qui peut être un avantage pour débugger un programme plus complexe.

«Déclenchement d'une action par un évènement.»



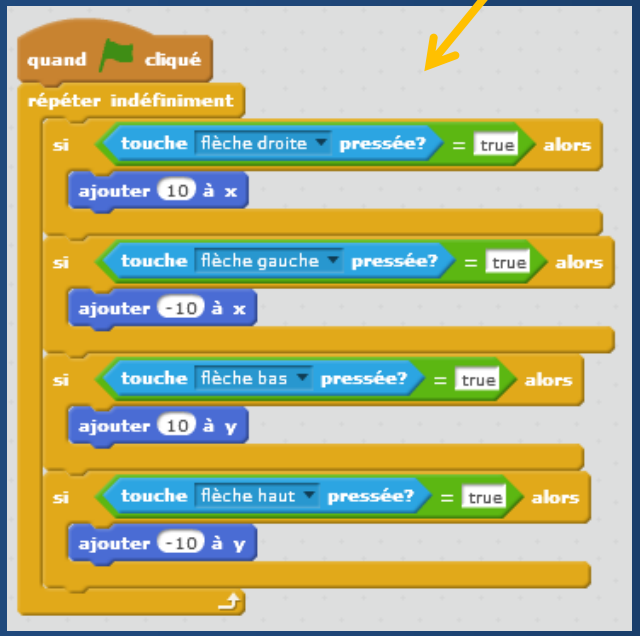
Partons du problème simple de l'objet chat qui doit ramasser les pommes.

Contraintes :
- le chat doit pouvoir se déplacer dans toutes les directions.
- Quand l'objet chat rencontre un objet pomme, celui-ci doit s'effacer et être comptabilisé.

La fonction « dupliquer un objet » va nous permettre rapidement de créer une cohorte de pommes identiques.

Cette instruction, que l'on peut ajouter au code d'une pomme pour « cloner » une pomme, et incrémenter la variable nombre de 2.

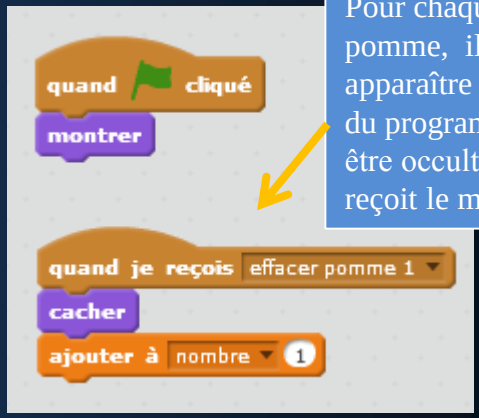
Boucle de gestion des mouvements de l'objet chat.



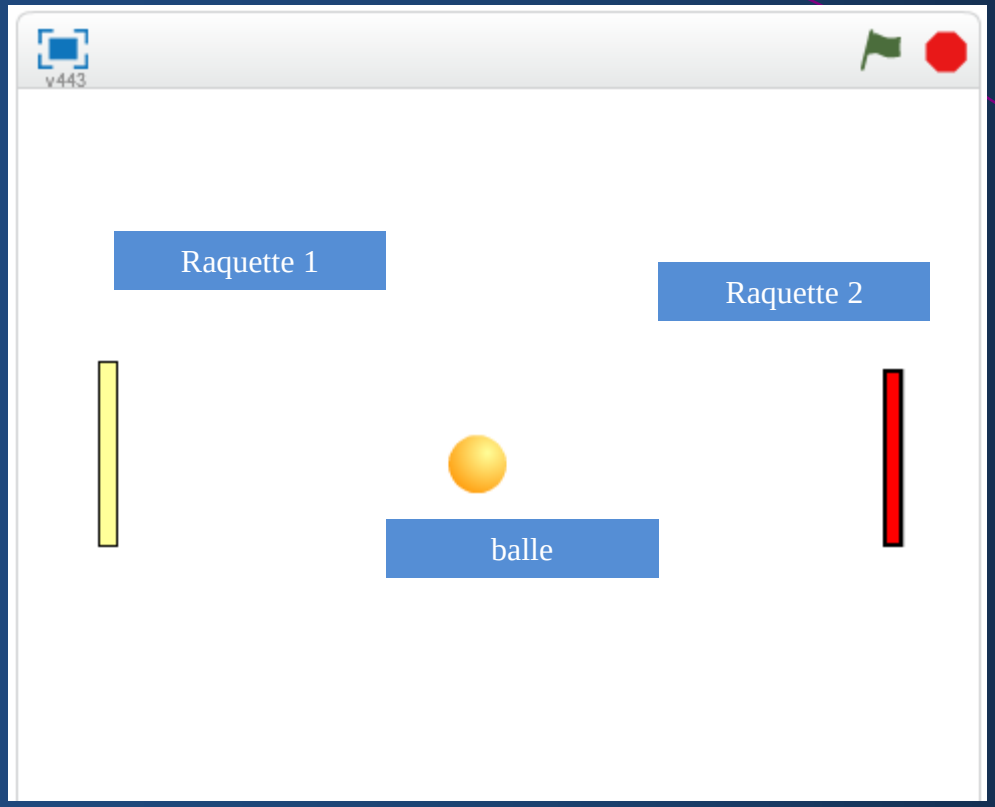
Boucle parallèle visant à envoyer aux objets pommes l'ordre de disparaître si elles sont touchées par l'objet chat.

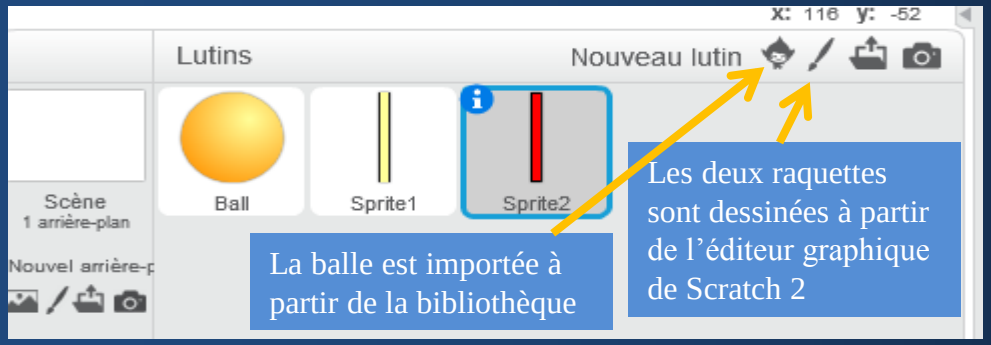
La variable nombre est initialisée à 0 au début du programme.

Pour chaque objet pomme, il doit apparaître au début du programme puis être occulté s' il reçoit le message.



«Décomposer un problème en sous-problèmes, afin de structurer un programme. 1/2»





La réalisation d'un jeu de « pong » encore nommé ping pong peut conduire à décomposer l'algorithme en plusieurs sous-problèmes :

Sous-problème 1 :

La balle doit rebondir sur les bords de l'écran.
La direction de la balle doit changer de 90°.

Sous-problème 2 :

La balle doit rebondir sur la raquette 1.
La direction de la balle doit changer de 90°.

Sous-problème 3 :

La balle doit rebondir sur la raquette 2.
La direction de la balle doit changer de 90°.

Sous-problème 4 :

La raquette 1 doit se déplacer verticalement en fonction des appuis sur les touches q et a.

Sous-problème 5 :

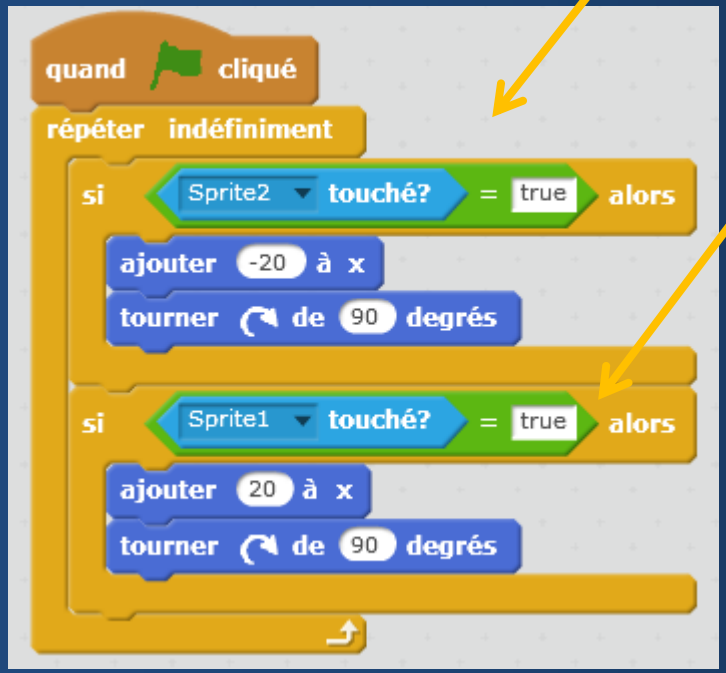
La raquette 2 doit se déplacer verticalement en fonction des appuis sur les touches p et l.

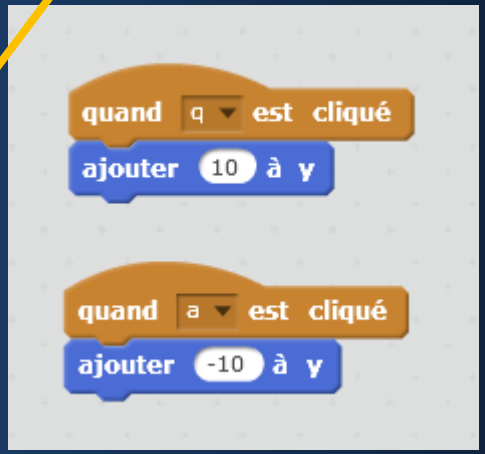
«Décomposer un problème en sous-problèmes, afin de structurer un programme. 2/2»

Sous-problème 1 : rebondir si le bord est atteint.

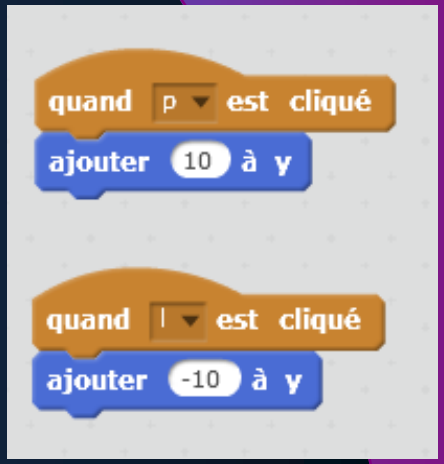
Sous-problème 2 : La balle doit rebondir sur la raquette 1 (sprite 2).

Sous-problème 3 : La balle doit rebondir sur la raquette 2 (sprite 1).



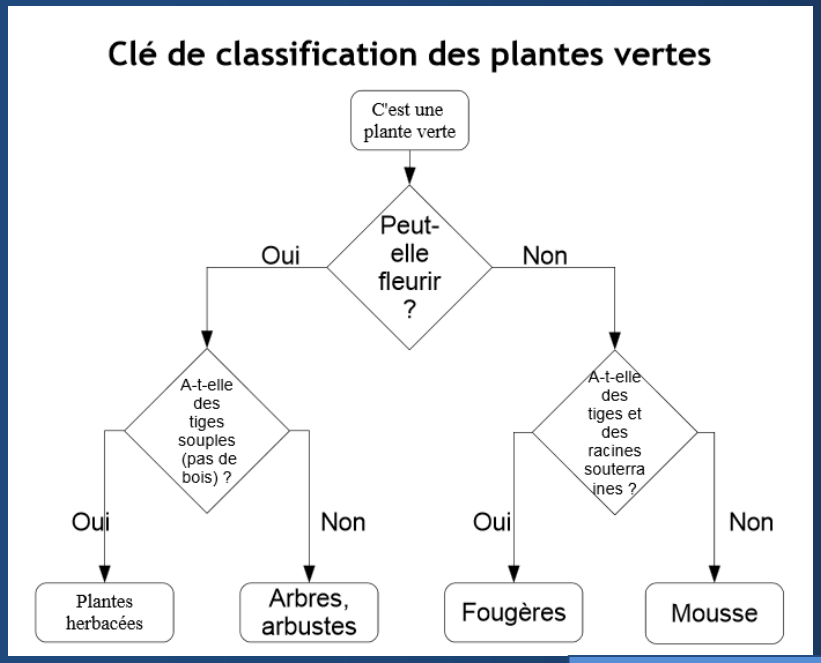


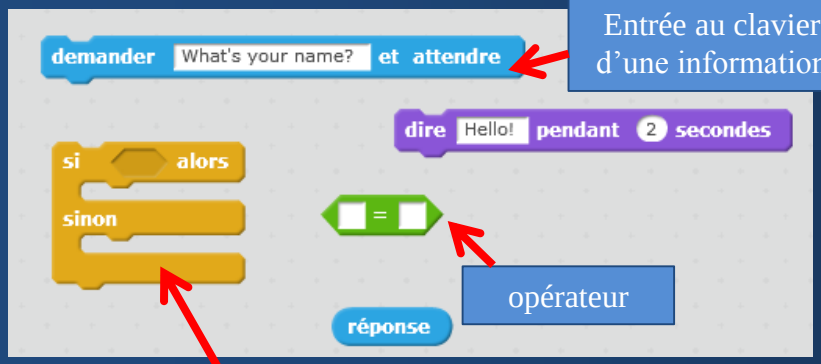
Sous-problème 4 : la raquette 1 doit se déplacer en appuyant sur les touches a ou q.



Sous-problème 5 : La raquette 2 doit se déplacer en appuyant sur les touches p ou l.

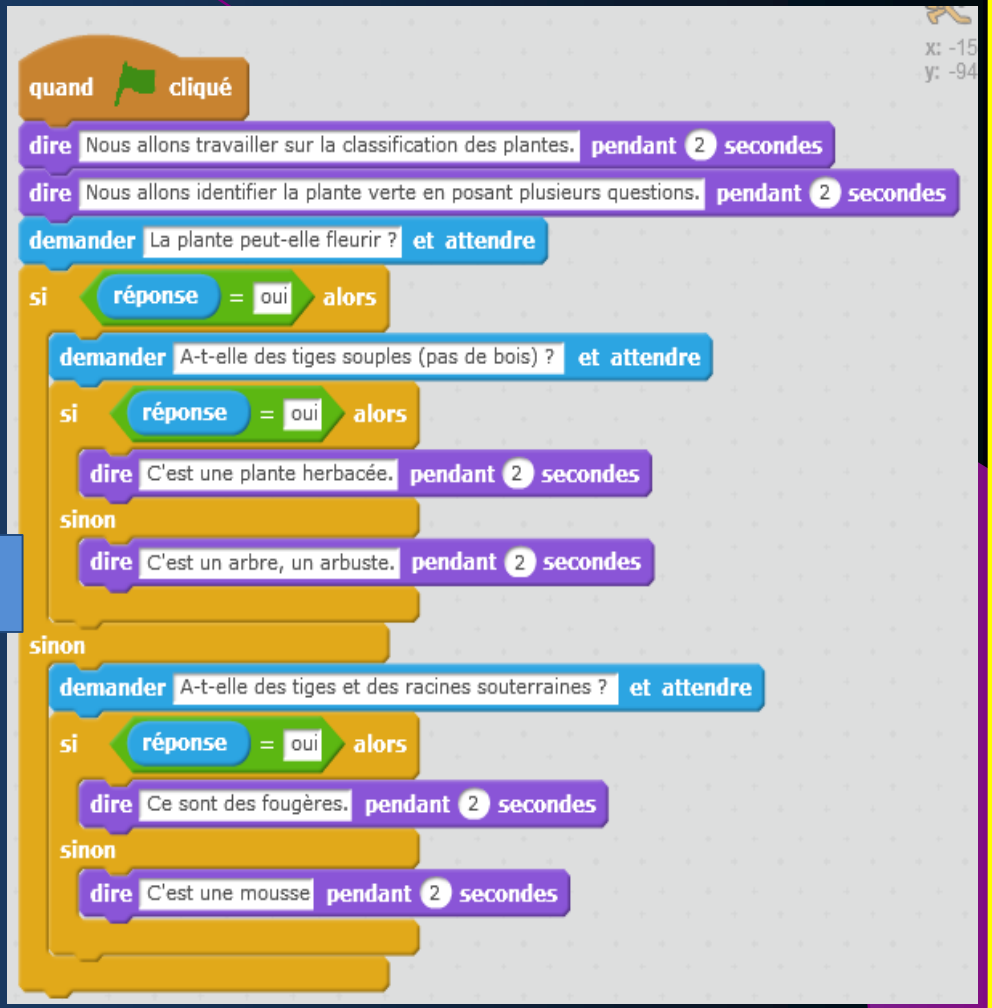
«Reconnaître des schémas.»





L'instruction conditionnelle « si, alors, sinon » permet de traduire l'algorithme en un programme.

A partir d'un organigramme, on peut être amené à réaliser le programme correspondant.



Proposition de projets et de supports pédagogiques 1/2

CYCLE 4 TECHNOLOGIE

Écrire, mettre au point et exécuter un programme

Connaissances et compétences associées	Exemples de situations, d'activités et de ressources pour l'élève
Analyser le comportement attendu d'un système réel et décomposer le problème posé en sous-problèmes afin de structurer un programme de commande. Écrire, mettre au point (tester, corriger) et exécuter un programme commandant un système réel et vérifier le comportement attendu. Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs. » Notions d'algorithme et de programme. » Notion de variable informatique. » Déclenchement d'une action par un évènement, séquences d'instructions, boucles, instructions conditionnelles. » Systèmes embarqués. » Forme et transmission du signal. » Capteur, actionneur, interface.	Concevoir, paramétrer, programmer des applications informatiques pour des appareils nomades. Observer et décrire le comportement d'un robot ou d'un système embarqué. En décrire les éléments de sa programmation Agencer un robot (capteurs, actionneurs) pour répondre à une activité et un programme donné. Écrire, à partir d'un cahier des charges de fonctionnement, un programme afin de commander un système ou un système programmable de la vie courante, identifier les variables d'entrée et de sortie. Modifier un programme existant dans un système technique, afin d'améliorer son comportement, ses performances pour mieux répondre à une problématique donnée. Les moyens utilisés sont des systèmes pluri-technologiques réels didactisés ou non, dont la programmation est pilotée par ordinateur ou une tablette numérique. Ils peuvent être complétés par l'usage de modélisation numérique permettant des simulations et des modifications du comportement.

Architecture et risques majeurs

Réalisation d'une table vibrante

en 5e

Simulateur de conduite routière

en 3e

Cette proposition peut donner lieu à un EPI

La maison domotique idéale

Système d'aide au stationnement dans un garage

en 4e

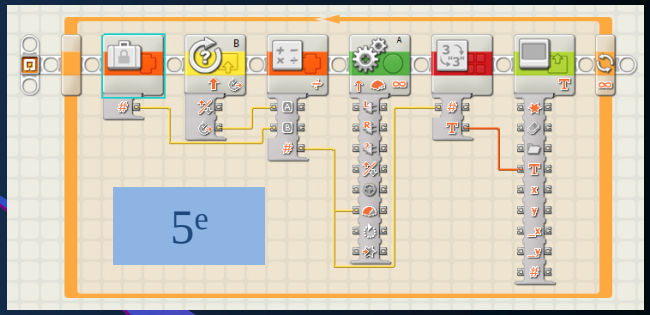
Proposition de projets et de supports pédagogiques 2/2

Repères de progressivité

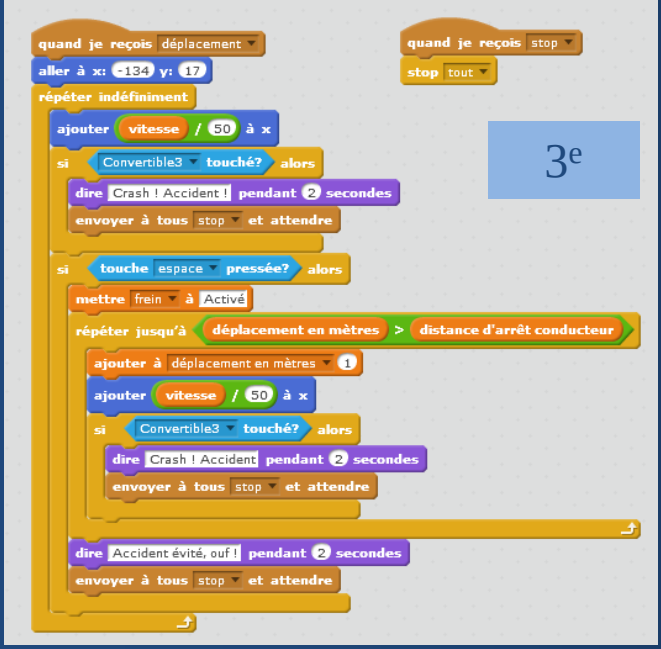
En 5e : traitement, mise au point et exécution de programme simple avec un nombre limité de variables d'entrée et de sortie, développement de programmes avec des boucles itératives.

En 4e : traitement, mise au point et exécution de programme avec introduction de plusieurs variables d'entrée et de sortie

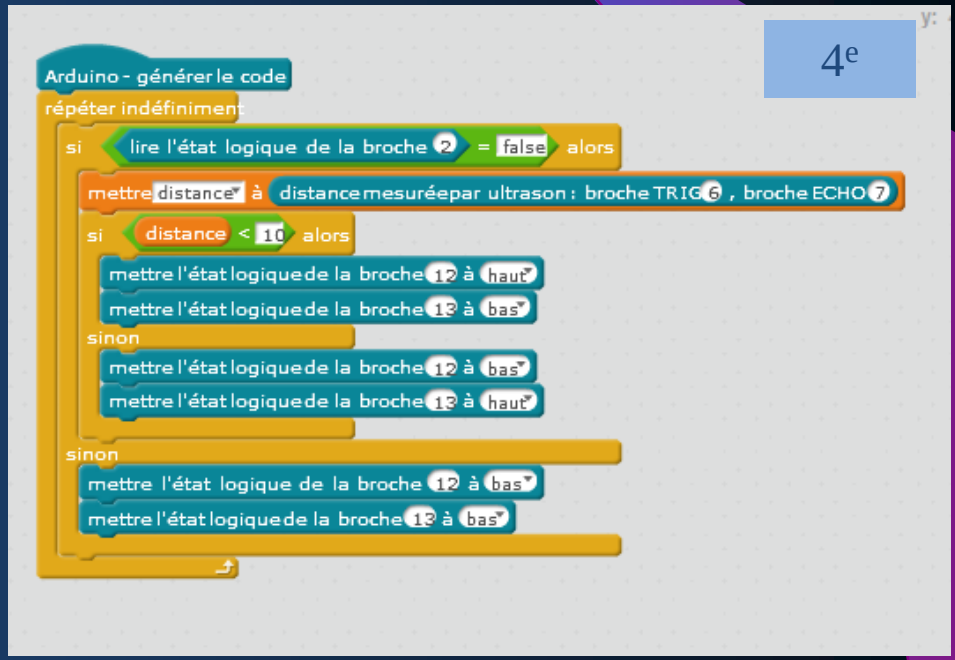
En 3e : introduction du comptage et de plusieurs boucles conditionnels imbriqués, décomposition en plusieurs sous-problèmes



Une boucle, une variable d'entrée, une variable de sortie.



Deux boucles imbriquées, un comptage, une variable de sortie.



Une boucle, deux variables d'entrée, une variable de sortie.

Qu’est ce qu’un système embarqué ? 1/2

Notions de base

Un **système embarqué** est défini comme un système électronique et informatique autonome, souvent temps réel, spécialisé dans une tâche bien précise. Le terme désigne aussi bien le matériel informatique que le logiciel utilisé. Ses ressources sont généralement limitées. Cette limitation est généralement d'ordre spatial (encombrement réduit) et énergétique (consommation restreinte).

C’est aussi un système informatique destiné à fonctionner dans un véhicule ou dans un appareil, comme un avion, un drône, un four ou encore un Smartphone.

Il existe différents types de systèmes embarqués :

Calcul général : Jeu vidéo.

”

Contrôle de systèmes en Temps Réel : Système de navigation aérien.

”

Traitement du signal : radar, sonar.

”

Transmission d’information et commutation : téléphone, internet.

Arduino est un système d'informatique embarqué grand public

Qu’est ce qu’un système embarqué ? 2/2

Notions de base

Carte Arduino Uno

Un prix dérisoire étant donné l'étendue des applications possibles. On comptera 20 euros pour la carte. Le logiciel est fourni gratuitement !

Une compatibilité sous toutes les plateformes, à savoir : Windows, Linux et Mac OS.

Une communauté ultra développée ! Des milliers de forums d'entr-aide, de présentations de projets, de propositions de programmes et de bibliothèques, …

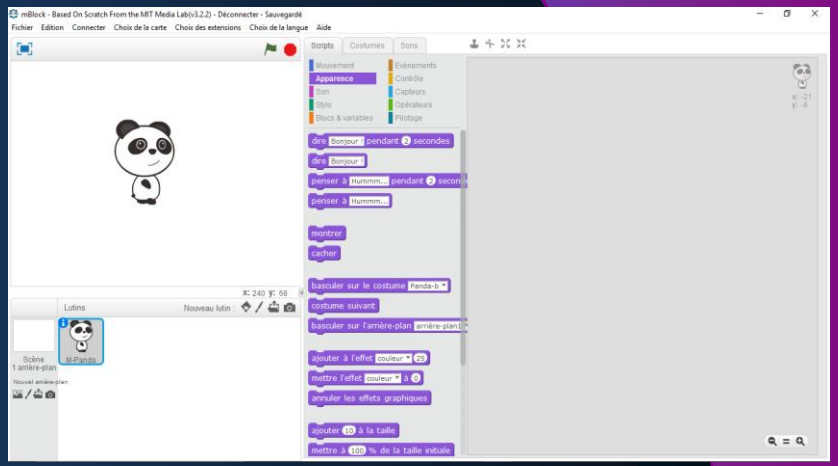
Un site en français arduino.cc où vous trouverez tout de la référence Arduino, le matériel, des exemples d'utilisations, de l'aide pour débuter, des explications sur le logiciel et le matériel, etc.

Une liberté quasi absolue. Elle constitue en elle-même deux choses :

- Le logiciel : gratuit et open source, développé en Java, dont la simplicité d'utilisation relève du savoir cliquer sur la souris.
- Le matériel : cartes électroniques dont les schémas sont en libre circulation sur internet.

Carte compatible Arduino Me Orion

De nombreux capteurs et actionneurs peuvent être connectés facilement à cette carte électronique.



Le logiciel mBblock est une extension de Scratch2, il permet le contrôle des cartes de base Arduino. On peut le télécharger gratuitement ici.

Forme et transmission du signal 1/2

Notions de base

Lien avec les Sciences Physiques

▶ CYCLE 4 PHYSIQUE-CHIMIE

Des signaux pour observer et communiquer

Attendus de fin de cycle

- Caractériser différents types de signaux (lumineux, sonores, radio…).
- Utiliser les propriétés de ces signaux.

Connaissances et compétences associées	Exemples de situations, d'activités et d'outils pour l'élève
Signaux lumineux Distinguer une source primaire (objet lumineux) d'un objet diffusant. Exploiter expérimentalement la propagation rectiligne de la lumière dans le vide et le modèle du rayon lumineux. Utiliser l'unité « année lumière » comme unité de distance. » Lumière : sources, propagation, vitesse de propagation, année lumière. » Modèle du rayon lumineux.	L'exploitation de la propagation rectiligne de la lumière dans le vide et le modèle du rayon lumineux peut conduire à travailler sur les ombres, la réflexion et des mesures de distance. Les activités proposées permettent de sensibiliser les élèves aux risques d'emploi des sources lumineuses (laser par exemple). Les élèves découvrent différents types de rayonnements (lumière visible, ondes radio, rayonsX…)
Signaux sonores Décrire les conditions de propagation d'un son. Relier la distance parcourue par un son à la durée de propagation. » Vitesse de propagation. » Notion de fréquence : sons audibles,	Les exemples abordés privilégient les phénomènes naturels et les dispositifs concrets : tonnerre, sonar... Les activités proposées permettent de sensibiliser les élèves aux risques auditifs.
Signal et information » Comprendre que l'utilisation du son et de la lumière permet d'émettre, de transporter un signal donc une information.	

La modélisation et la simulation des objets et systèmes techniques

Connaissances et compétences associées	Exemples de situations, d'activités et de ressources pour l'élève
Analyser le fonctionnement et la structure d'un objet	
Mesurer des grandeurs de manière directe ou indirecte. » Instruments de mesure usuels. » Principe de fonctionnement d'un capteur, d'un codeur, d'un détecteur » Nature du signal : analogique ou numérique. » Nature d'une information : logique ou analogique.	

Les différents types de signaux utilisés pour transmettre des informations

Les impulsions électriques permettent de transmettre des informations dans des câbles électriques.

Les impulsions lumineuses permettent de transmettre des informations dans l'air ou dans le vide avec des infrarouges ou dans le verre d'une fibre optique

Les vibrations mécaniques permettent de transmettre des informations dans l'air ou dans l'eau en utilisant des ultrasons.

Les ondes électromagnétiques permettent de transmettre des informations dans l'air ou dans le vide.

Forme et transmission du signal 2/2

Un signal correspond à une variable ou une source d'information évoluant au cours du temps.

Dès qu'on parle de communication de donnée, les termes : signal analogique et numérique reviennent constamment.

Dans un signal analogique, tel que celui en usage pour la diffusion radio et TV, les informations voyagent sous forme d'onde continûment variable. Comme le montre l'illustration suivante.

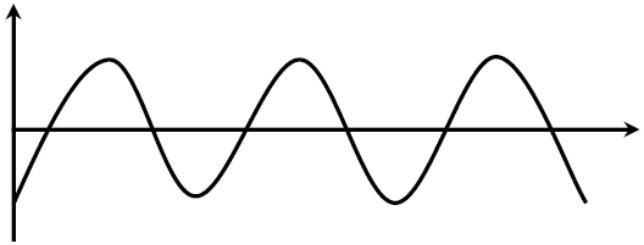


Figure 1.1 Exemple de signal analogique

Numérique : lorsqu' on a affaire à des signaux numériques, on est en face de signaux plus simples, dans la mesure où les informations circulent au moyen d'impulsions binaires (avec seulement deux états).

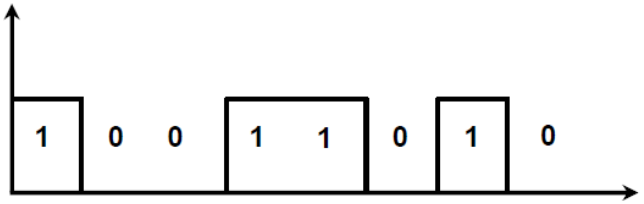


Figure 1.2 Exemple de signal numérique

Par exemple, du courant peut être envoyé sur le fil électrique pour transmettre un '1' binaire, et l'absence de courant équivaut à un '0' binaire.

Capteurs, actionneurs, interfaces.

Notions de base

Un **capteur** est un dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable, telle qu'une tension électrique, une hauteur de mercure, une intensité ou la déviation d'une aiguille.

Capteurs analogiques
La sortie est une grandeur électrique dont la valeur est une fonction de la grandeur physique mesurée par le capteur. La sortie peut prendre une infinité de valeurs continues. Le signal des capteurs analogiques peut être du type : tension ou courant.

Capteurs logiques
Ou *capteurs TOR.* La sortie est un état logique que l'on note 1 ou 0. La sortie peut prendre ces deux valeurs. Le signal des capteurs logiques peut être du type : courant présent/absent dans un circuit ; potentiel, souvent 5 V/0 V ;

Capteurs numériques
La sortie est une séquence d'états logiques qui, en se suivant, forment un nombre. La sortie peut prendre une infinité de valeurs discrètes. Le signal des capteurs numériques peut être du type : train d'impulsions, avec un nombre précis d'impulsions ou avec une fréquence précise ; code numérique binaire.

Un actionneur est un organe qui transforme l'énergie qui lui est fournie en un phénomène physique utilisable.
Le phénomène physique fournit un travail qui modifie le comportement ou l'état du système.

Exemples d'actionneurs
un vérin pneumatique ou hydraulique génère un mouvement à partir d'une énergie mécanique transmise par un fluide gazeux ou liquide ou à partir de l'énergie électrique.
il est possible d'avoir :

- un mouvement grâce à un électroaimant, ou un moteur électrique.
- de la chaleur grâce à une résistance électrique.
- de la lumière grâce à une lampe, une DEL.
- un champ magnétique grâce un électroaimant.
- un son grâce à une enceinte acoustique.

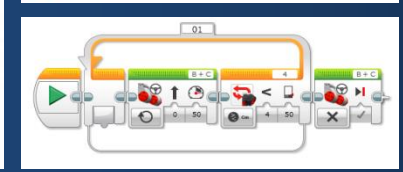
Interface :

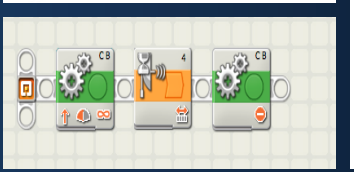
- Dispositif permettant la liaison de deux circuits électroniques ne devant pas avoir de répercussion l'un sur l'autre.
- En informatique, jonction entre deux matériels ou logiciels leur permettant d'échanger des informations par l'adoption de règles communes ; module matériel ou logiciel permettant la communication d'un système avec l'extérieur.

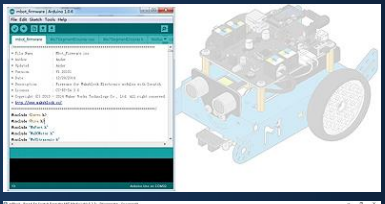
Des propositions de matériels et logiciels en technologie 1/2

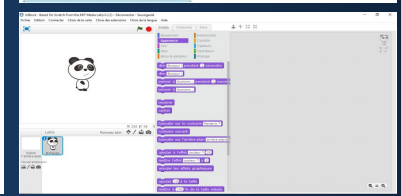
Robotique

http://www.lego.com/fr-fr/mindstorms/downloads









Ces robots intègrent la carte Me Orion compatible Arduino. Elle se programme avec les outils Arduino ou le logiciel mBlock.

http://www.lextronic.fr/P35755-robot-mbot-blue-bluetooth-version.html

http://www.makeblock.cc/

Des propositions de matériels et logiciels en technologie 2/2

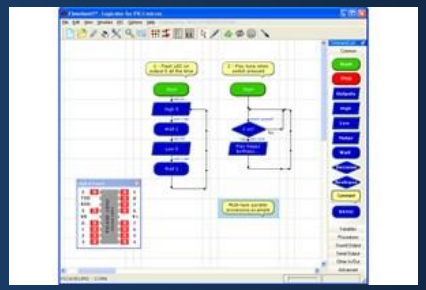
http://tic.technologiescollege.fr/portail/portal/index.php#tab/3

Robotique (suite)

Ces robots sont constitués de pièces réalisées avec une imprimante 3D, ils se programment avec l'application BlockyDuino (gratuite), dans un langage graphique proche de celui de Scratch 2.0

Interfaces programmables pouvant être utilisées comme systèmes embarqués

Boîtiers Autoprog



http://www.picaxe.com/Software/PICAXE/Logicator-for-PICAXE/

Logicator for PIC and PICAXE

Interfaces

Lego Wedo

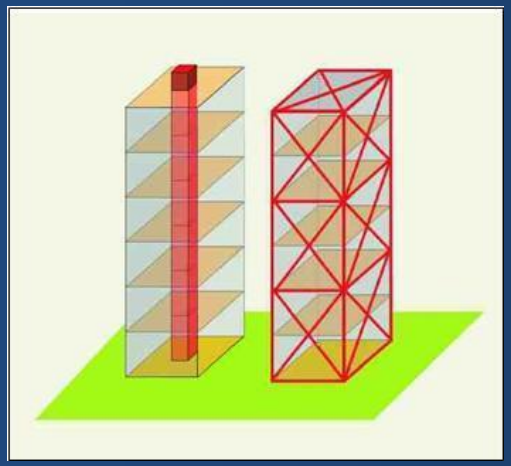
Interface Picoboard

Webcam utilisée comme capteur

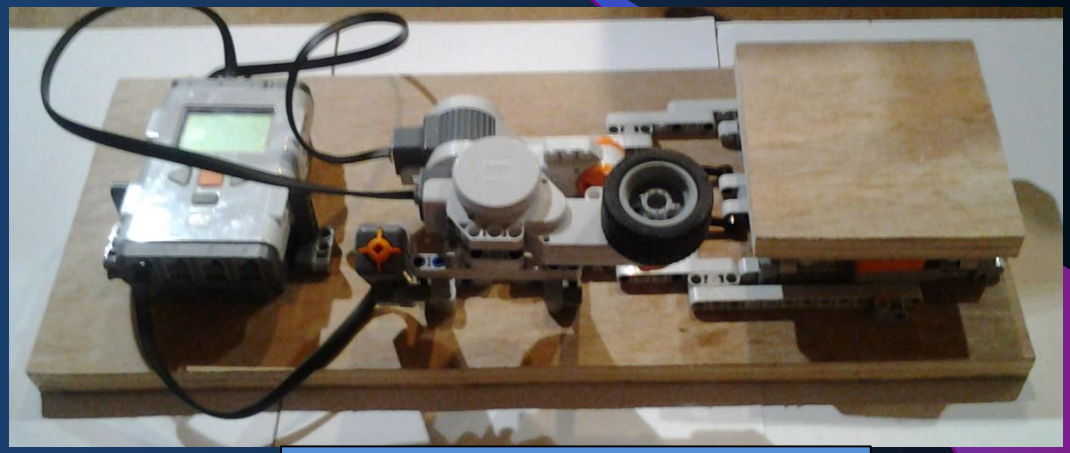
Scratch 2

Table vibrante de simulation des effets d'un séisme en 5e

Il existe actuellement une technique expérimentale pour simuler les effets d'un séisme sur une structure : elle est réalisée au moyen d'une table vibrante. La table vibrante représente le sol qui sous l'effet du séisme vibre dans une, deux ou trois directions suivant le type de table. Le bâtiment, modélisé le plus souvent à échelle réduite, est fixé sur la table. On peut alors enregistrer les effets. A quel moment la structure s'écroule-t-elle ? Y a-t-il des fissures dans les murs ? Comment survient l'endommagement ? Pendant combien de temps la structure reste-t-elle debout ?



Pour permettre les essais avec différentes structures



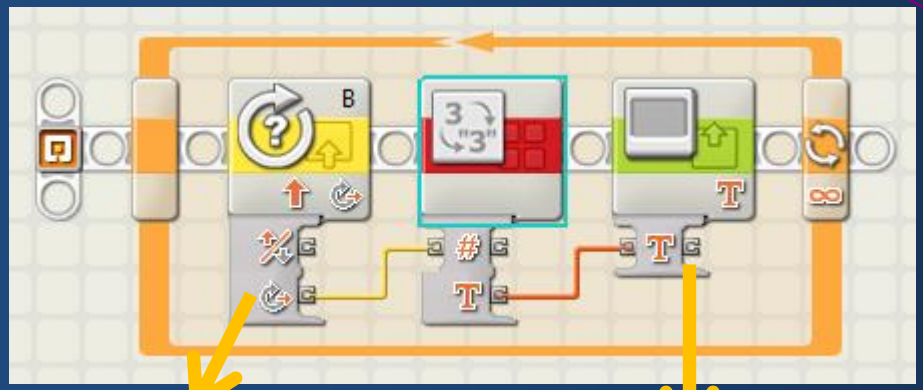
Prototype de table vibrante

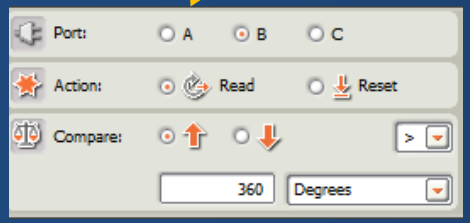
Problème posé : on voudrait modifier le système pour permettre de faire varier la vitesse de la table.

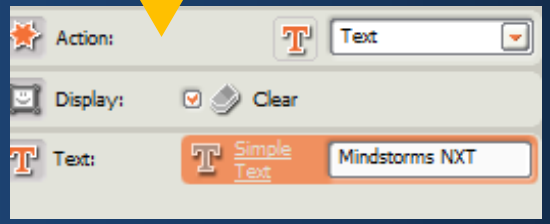
La rotation du capteur d'angle du servomoteur va nous permettre de faire varier la vitesse de la table.

Le programme initial

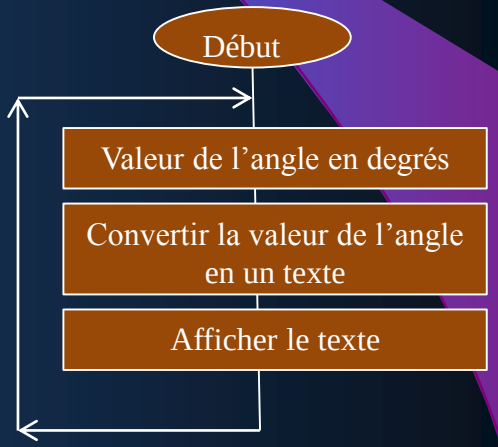
Ressources pour réaliser le programme de la table vibrante 1/2

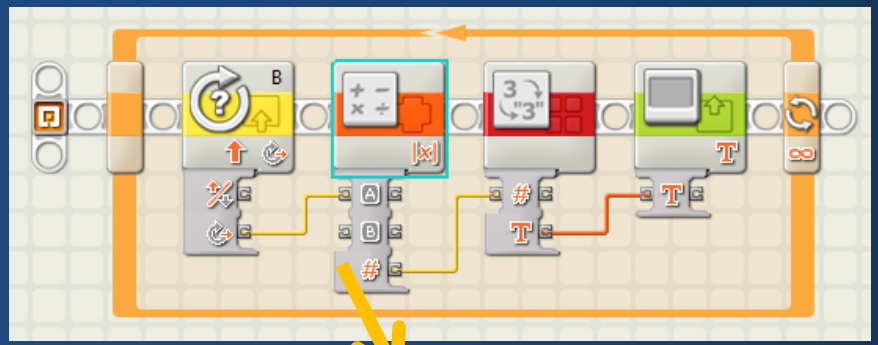


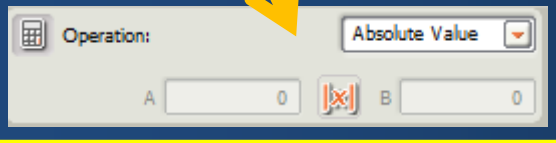




Ce programme affiche sur l'interface programmable la valeur de l'angle en degrés correspondant à la rotation du servomoteur connecté sur B.







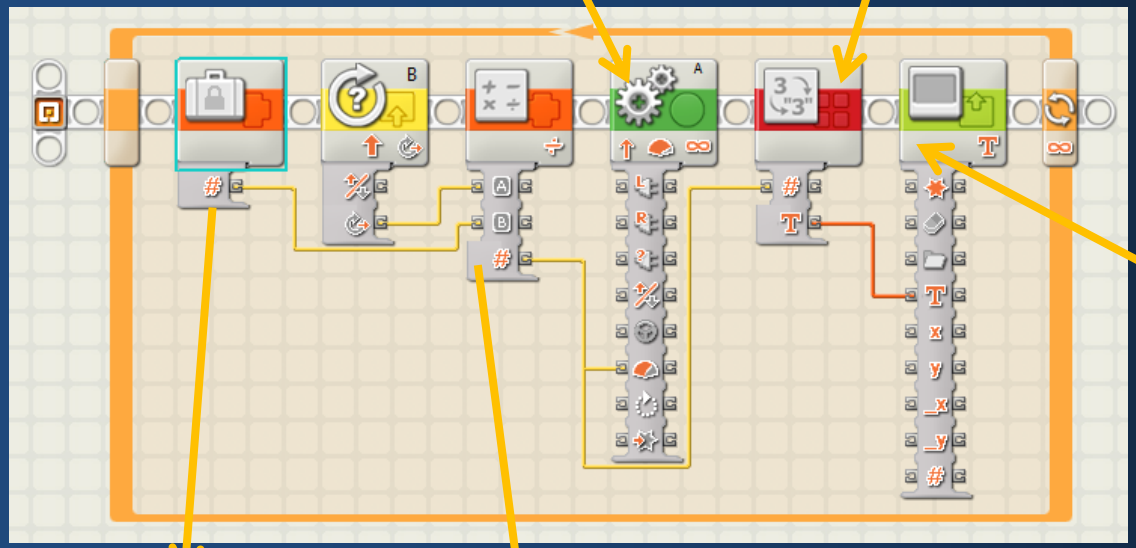
En ajoutant la fonction mathématique valeur absolue, la valeur affichée est positive

Ressources pour réaliser le programme de la table vibrante 2/2

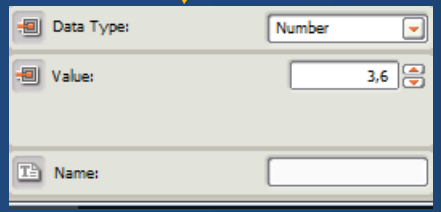
Piloter le moteur A avec une variation de puissance de 0 à 100.

Cette valeur est convertie en une donnée de type texte

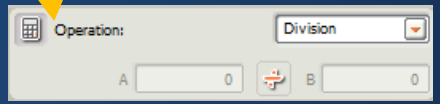
Ce programme affiche sur l'interface programmable une valeur de 0 à 100 correspondant à la puissance à fournir au moteur A



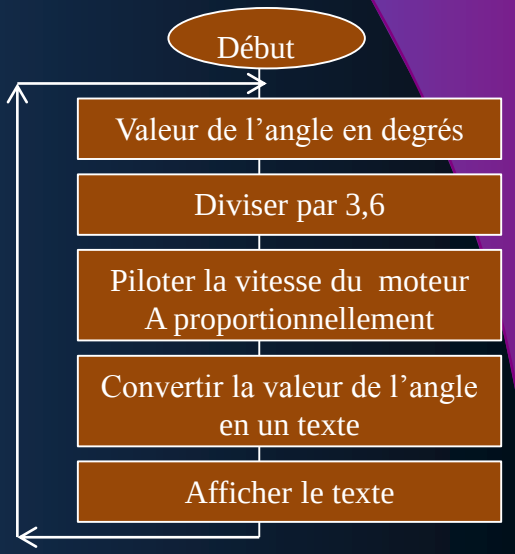
Ce texte est affiché sur la brique programmable



Constante fixée à 3,6

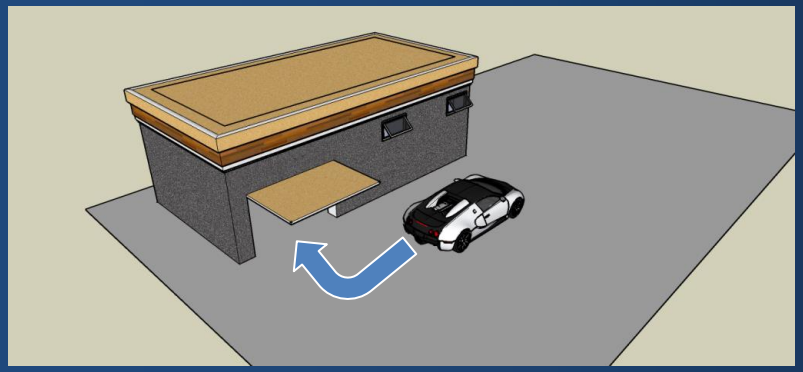


Diviser la variation de 0 à 360° par 3,6 pour récupérer une variation de 0 à 100.



La maison domotique idéale un système d'aide au stationnement dans un garage en 4e

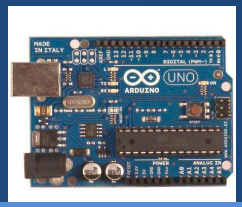
Le problème posé :



Le véhicule familial ne dispose pas de radar de recule ni de système d'aide au stationnement assisté.

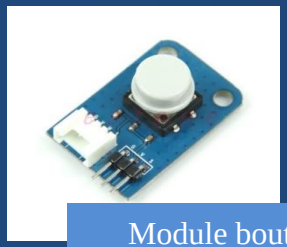
Comment faciliter le stationnement en marche arrière du véhicule dans ce garage ?

La solution réalisée par Monsieur Domotic :

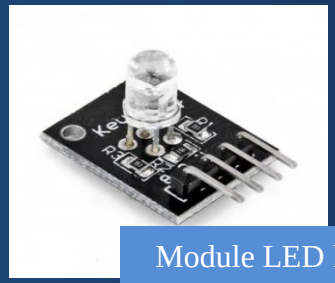


Carte électronique Arduino Uno

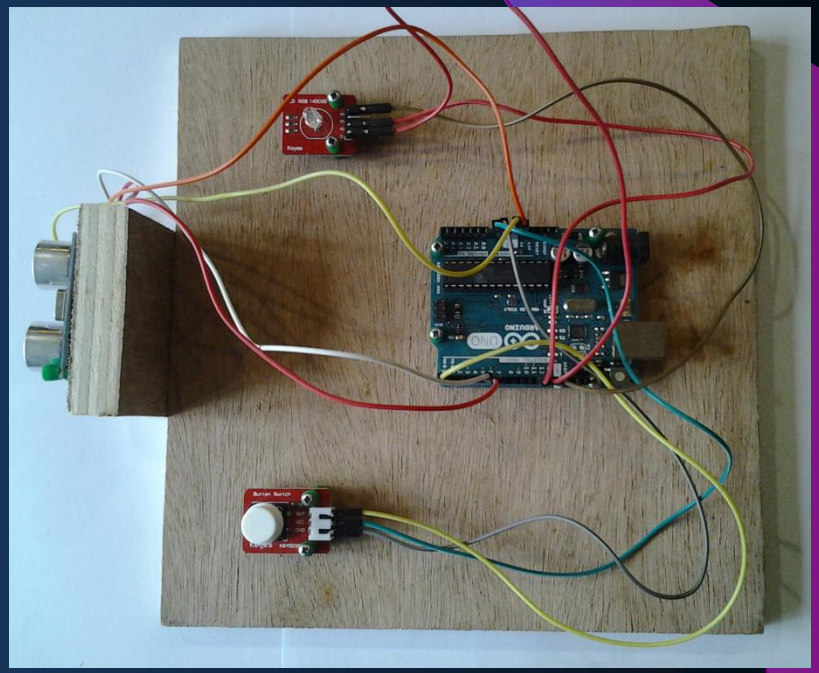
Module émetteur récepteur à ultrasons



Module bouton poussoir

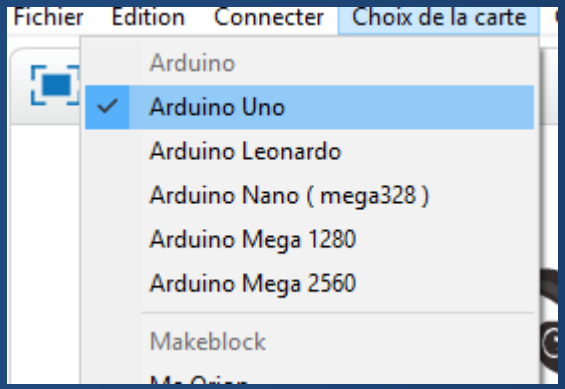


Module LED haute luminosité 3 couleurs

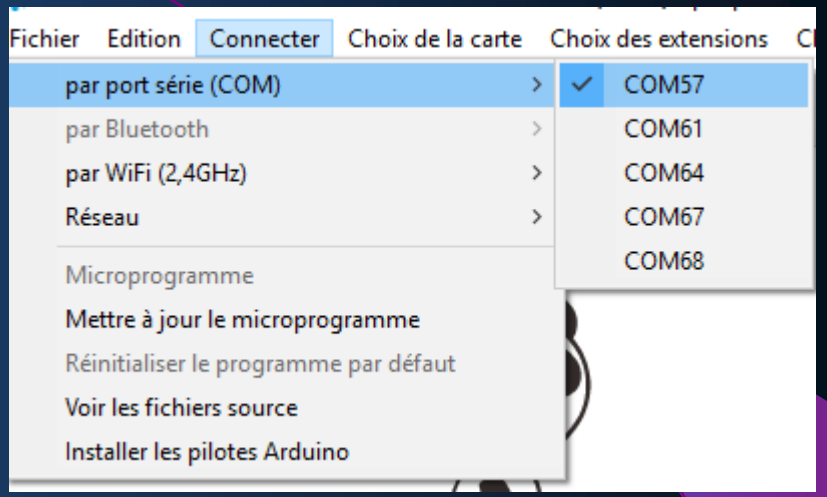


Tutoriel de prise en main du logiciel mBlock

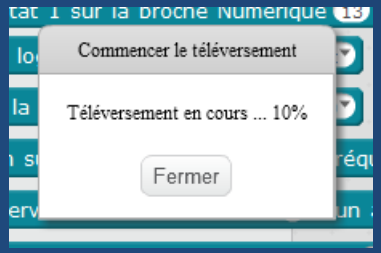
Réalisation d'un radar de recul :



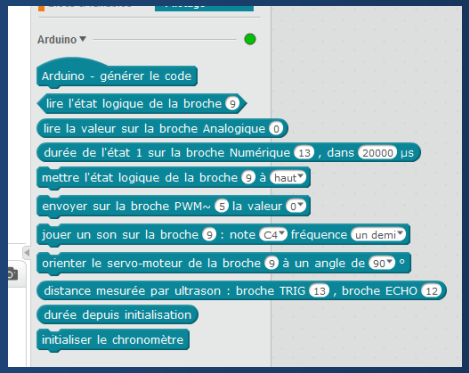
1 Choisir le type de matériel connecté



2 Sélectionner le port série sur lequel la carte est connectée



3 Dans le menu « Connecter », choisir l'option « Mettre à jour le micro programme » Pour effacer le contenu de la mémoire de la carte.

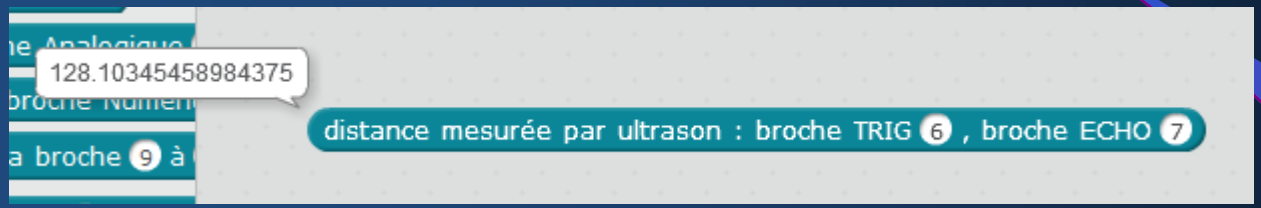


4 Cliquer dans le menu « Pilotage », un nouveau jeu d'instruction est présent. Le voyant vert indique que la carte est connecté

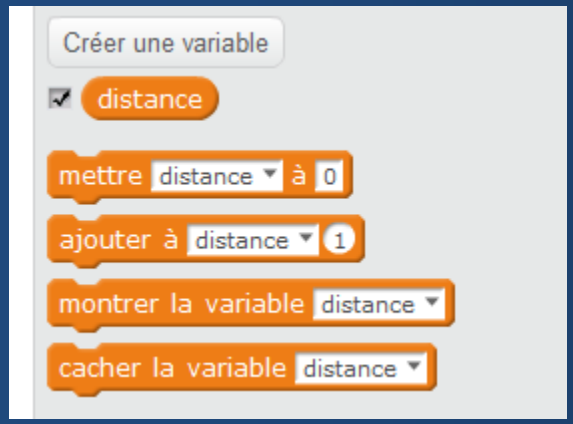
5 Glisser l'instruction « distance mesurée par ultrason ». Connecter :
- la broche GND du module sur GND de la carte.
- la broche VCC du module sur +5 Volts de la carte.
- la broche trig sur la borne 6 de la carte.
- la broche echo sur la borne 7 de la carte.

Tutoriel de prise en main du logiciel mBlock

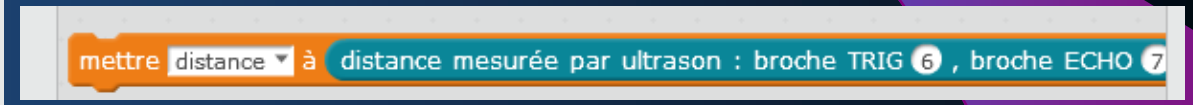
6 Paramétrer l'instruction correctement.



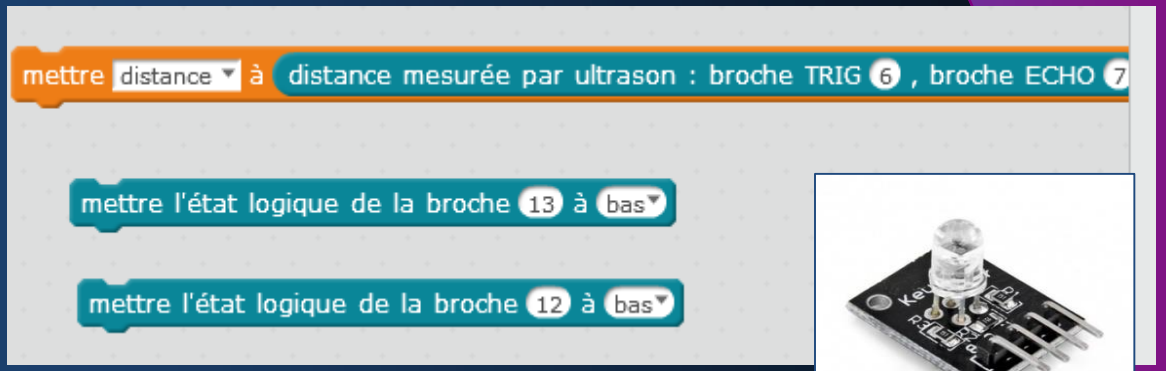
7 Cliquer sur l'instruction, la valeur numérique acquise correspond à la distance en centimètres entre le capteur et le premier obstacle devant lui.



8 Créer la variable « distance ».



9 Affecter la variable distance de la valeur retournée par le capteur.



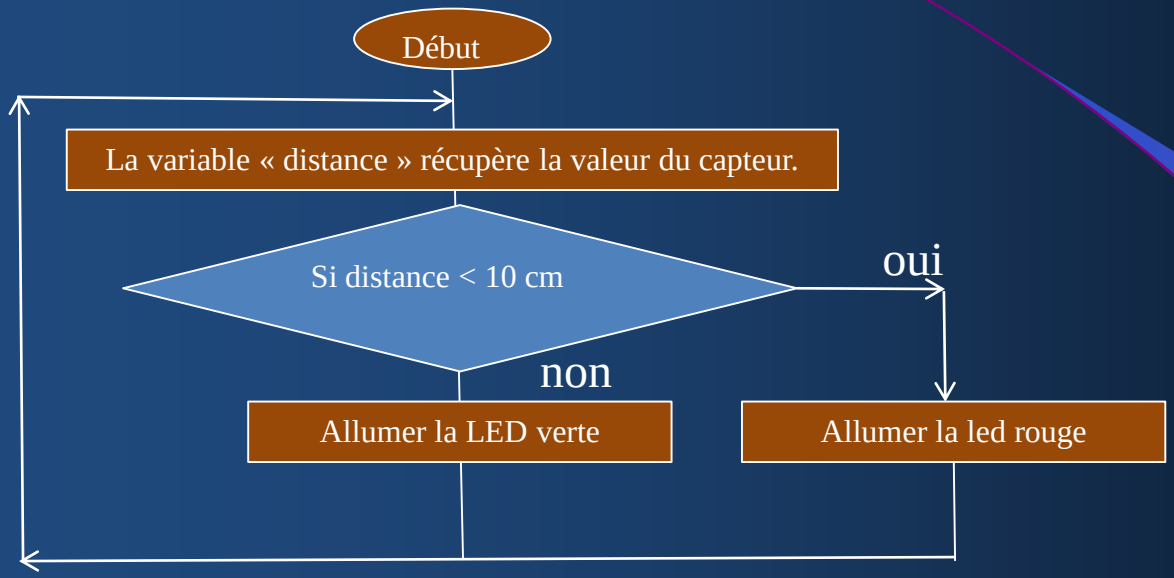
10 Connecter le module LED :

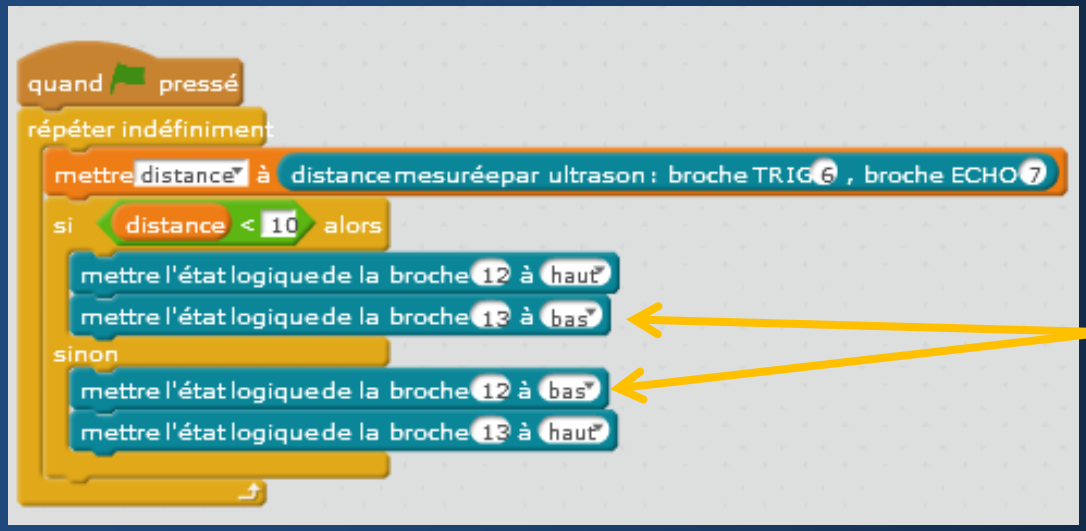
- borne GND sur GND de la carte.
- borne G du module (green) sur la borne 13 de la carte.
- borne R du module (red) sur la borne 12 de la carte.

Allumer et éteindre la led RGB en vert ou rouge en choisissant les états haut et bas et en cliquant sur les instructions.

Tutoriel de prise en main du logiciel mBlock

11 Traduisons cet algorithme en une suite d'instructions.

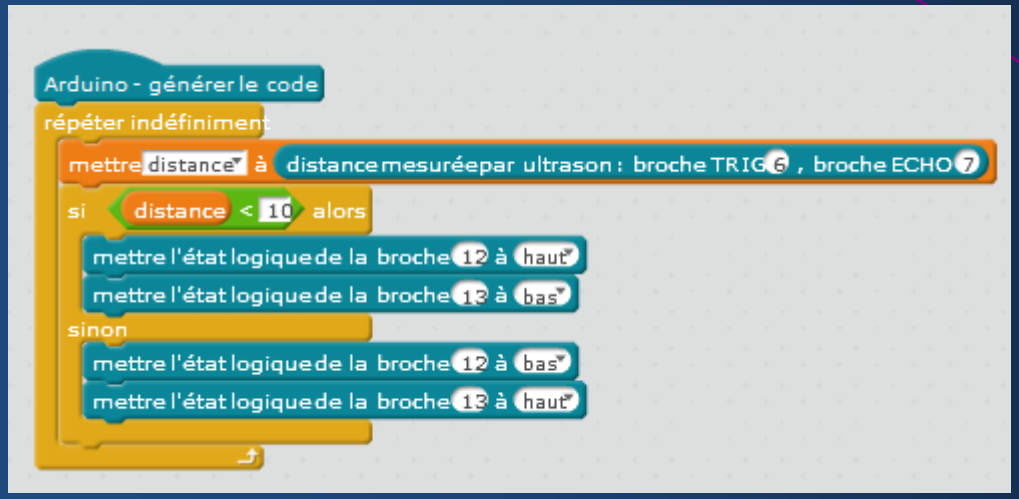




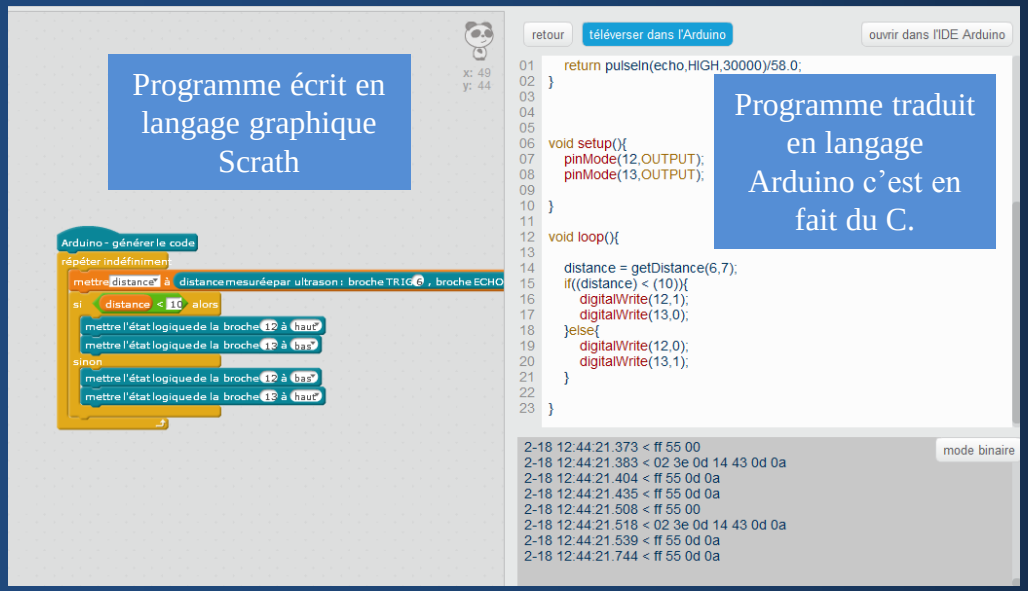
Il faudra penser à éteindre les composantes rouge et verte de la LED.

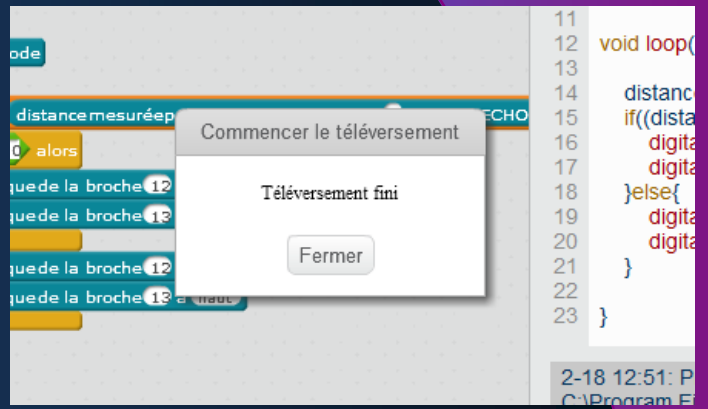
Tutoriel de prise en main du logiciel mBlock

12 En liaison avec le logiciel Arduino, téléversons le programme dans la carte électronique pour la rendre autonome.



Nos essais étant satisfaisants, il faut maintenant convertir le programme en langage compatible Arduino. On remplace le début du programme par « Arduino – générer le code » et cliquer dessus.





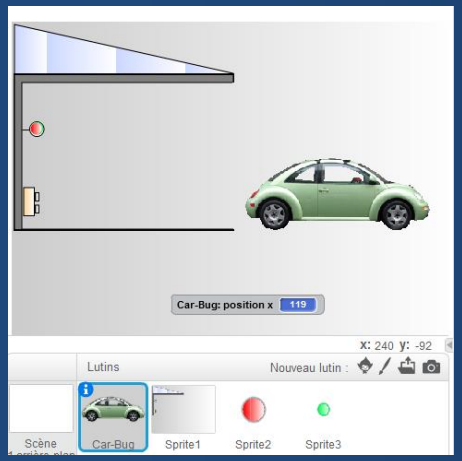
Une fois le programme traduit, il est téléchargé dans la mémoire du micro contrôleur. La carte est maintenant autonome, plus besoin de mBlock.

Tutoriel de prise en main du logiciel mBlock

Modifier un programme existant dans un système technique, afin d'améliorer son comportement, ses performances pour mieux répondre à une problématique donnée.

Problème posé : à partir du programme précédent, Monsieur Domotic souhaiterait que son système d'aide au stationnement dans le garage démarre si la porte de garage est ouverte.

Réalisons en premier lieu une simulation avec le logiciel mBlock.

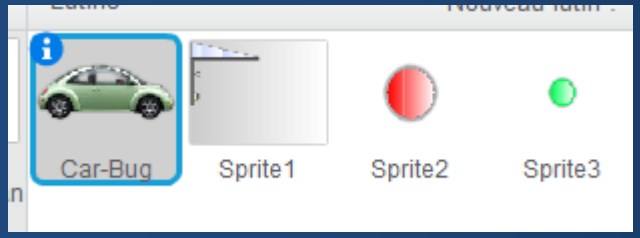


Importer le dessin de voiture Car Bug depuis la bibliothèque

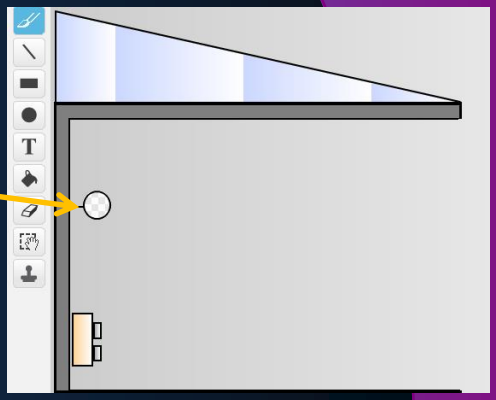
Créer un nouveau « lutin » ou objet et dessiner le garage.

Créer un nouveau « lutin » en forme de disque rouge.

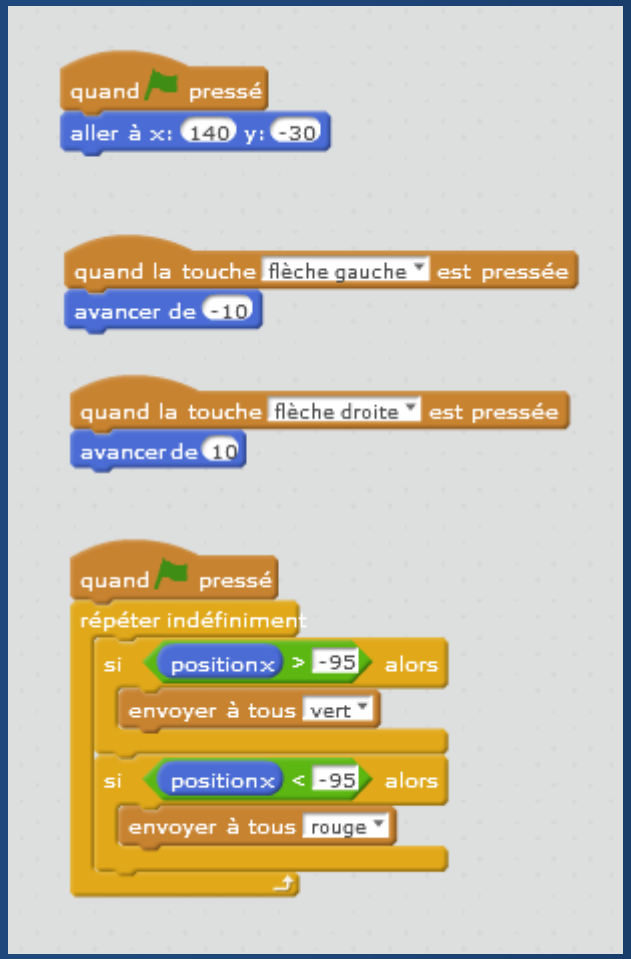
Créer un nouveau « lutin » en forme de disque vert.

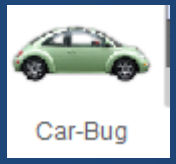


Placer les disques vert et rouge sur le cercle dans le dessin de garage.



Tutoriel de prise en main du logiciel mBlock

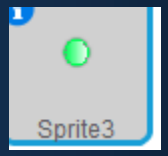


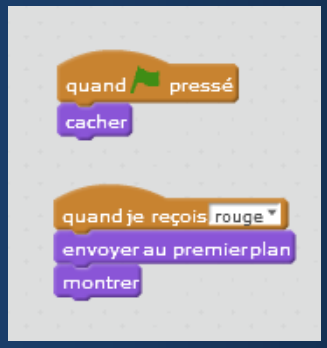


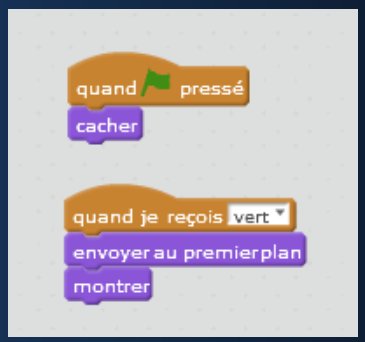
Pour le code, il faut pouvoir positionner la voiture à droite quand le programme commence.

Il faut gérer les déplacements de la voiture en fonction des appuis sur les touches du clavier : flèches gauche et droite.

Si la position en x de la voiture > -95 alors le message « vert » est envoyé, sinon, c'est le message « rouge » qui est envoyé.





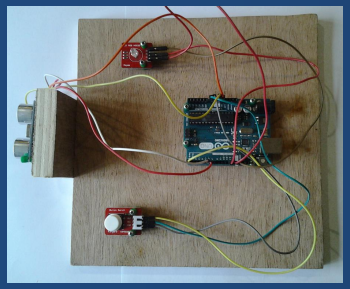


En fonction de la position de la voiture, les disques vert et rouge sont affichés au premier plan.

Tutoriel de prise en main du logiciel mBlock

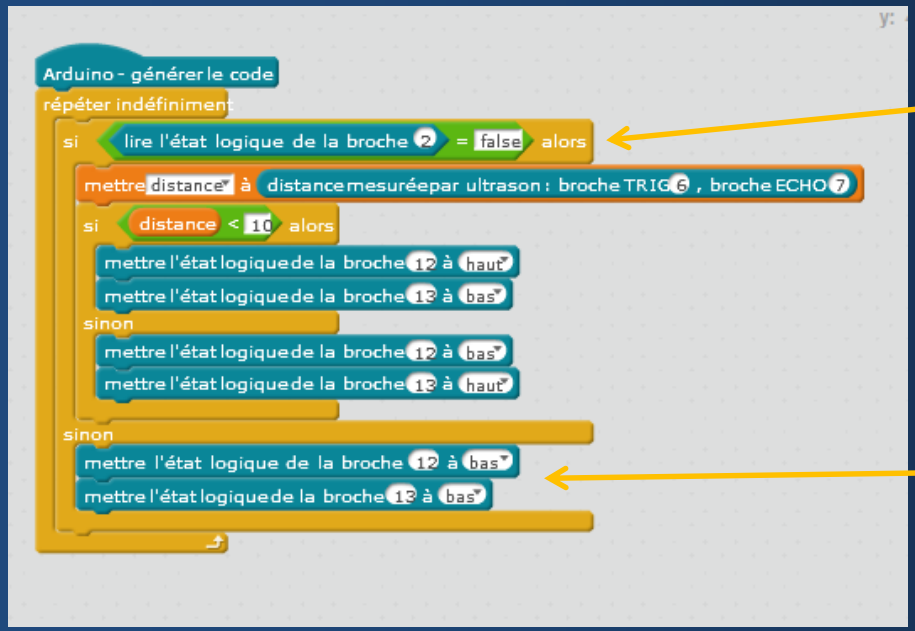
Modifier un programme existant dans un système technique, afin d'améliorer son comportement, ses performances pour mieux répondre à une problématique donnée.

Connectons un bouton poussoir simulant l'ouverture complète de la porte



Le module bouton poussoir comporte 3 connexions :

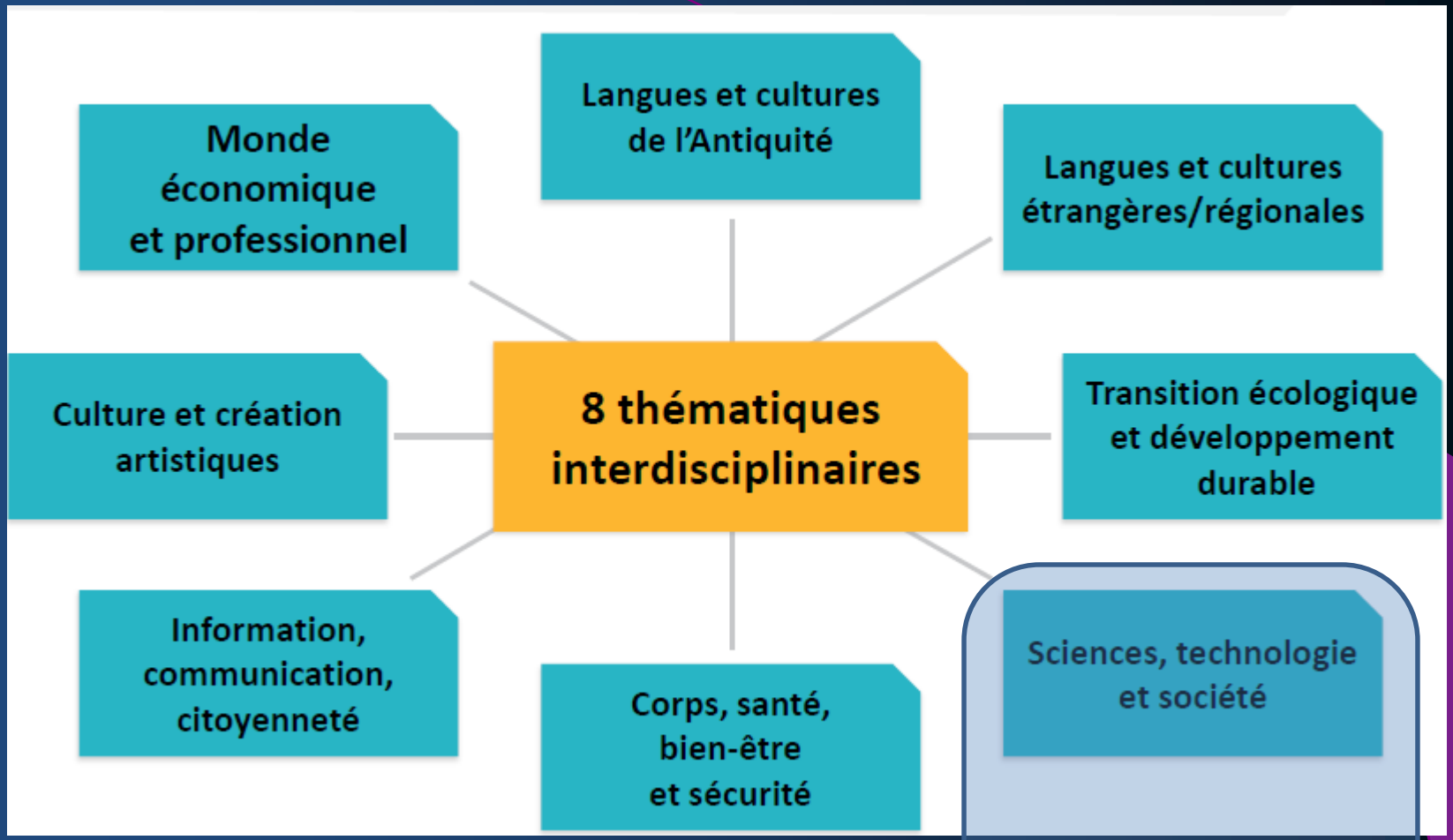
- +Vcc à connecter sur le +5V de la carte.
- GN D à connecter sur GND de la carte (0 Volt).
- OUT (sortie) à connecter sur la borne 2 de la carte.



Si la porte de garage est ouverte, l'état du bouton poussoir change, il passe de true à false. Dans ce cas le programme exécute la détection de distance.

Si la porte de garage est fermée alors les composantes verte et rouge de la LED sont éteintes.

Les enseignements complémentaires : les enseignements pratiques interdisciplinaires



Simulateur de conduite routière en 3e proposition d'un EPI en Mathématiques et Technologie

EPI Mathématiques et Technologie

Simulateur de conduite routière en 3e proposition d'un EPI en Mathématiques et Technologie

Connaissances et compétences associées	Exemples de situations, d'activités et de ressources pour l'élève
Décomposer un problème en sous-problèmes afin de structurer un programme ; reconnaître des schémas. Écrire, mettre au point (tester, corriger) et exécuter un programme en réponse à un problème donné. Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs. Programmer des scripts se déroulant en parallèle. » Notions d'algorithme et de programme. » Notion de variable informatique. » Déclenchement d'une action par un évènement, séquences d'instructions, boucles, instructions conditionnelles.	Jeux dans un labyrinthe, jeu de Pong, bataille navale, jeu de nim, tic tac toe. Réalisation de figure à l'aide d'un logiciel de programmation pour consolider les notions de longueur et d'angle. Initiation au chiffrement (Morse, chiffre de César, code ASCII…). Construction de tables de conjugaison, de pluriels, jeu du cadavre exquis… Calculs simples de calendrier. Calculs de répertoire (recherche, recherche inversée...). Calculs de fréquences d'apparition de chaque lettre dans un texte pour distinguer sa langue d'origine : français, anglais, italien, etc.

Mathématiques

▶ CYCLE 4 TECHNOLOGIE

Écrire, mettre au point et exécuter un programme	
Analyser le comportement attendu d'un système réel et décomposer le problème posé en sous-problèmes afin de structurer un programme de commande. Écrire, mettre au point (tester, corriger) et exécuter un programme commandant un système réel et vérifier le comportement attendu. Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs. » Notions d'algorithme et de programme. » Notion de variable informatique. » Déclenchement d'une action par un évènement, séquences d'instructions, boucles, instructions conditionnelles. » Systèmes embarqués. » Forme et transmission du signal. » Capteur, actionneur, interface.	Concevoir, paramétrer, programmer des applications informatiques pour des appareils nomades. Observer et décrire le comportement d'un robot ou d'un système embarqué. En décrire les éléments de sa programmation Agencer un robot (capteurs, actionneurs) pour répondre à une activité et un programme donnés. Écrire, à partir d'un cahier des charges de fonctionnement, un programme afin de commander un système ou un système programmable de la vie courante, identifier les variables d'entrée et de sortie. Modifier un programme existant dans un système technique, afin d'améliorer son comportement, ses performances pour mieux répondre à une problématique donnée. Les moyens utilisés sont des systèmes pluri-technologiques réels didactisés ou non, dont la programmation est pilotée par ordinateur ou une tablette numérique. Ils peuvent être complétés par l'usage de modélisation numérique permettant des simulations et des modifications du comportement.

Technologie

En mathématiques

« Une place importante doit être accordée à la résolution de problèmes internes aux mathématiques, liés à des situations issues de la vie courante ou d'autres disciplines. »

Un projet, le Simulateur de conduite

En technologie

« Un axe des sciences industrielles de l'ingénieur pour comprendre, simuler, concevoir les systèmes contemporains, en relation avec les sciences expérimentales dans des démarches d'investigation et de résolution de problème. »

Les EPI sont adossés aux nouveaux programmes

Un projet, le simulateur de conduite

Proposition d'organisation : 1 h hebdomadaire en Mathématiques et Technologie en co-animation si possible.

- Les EPI sont **adossés aux nouveaux programmes**
- Une **démarche de projet…**
- … conduisant à une **réalisation concrète, individuelle ou collective**
- Incluant l'usage des **outils numériques** et la pratique des **langues vivantes**
- Contribuant à la **mise en œuvre des parcours éducatifs**

L'EPI consiste donc à rendre plus accessible à l'élève les connaissances, les compétences, les procédures mises en œuvre dans plusieurs disciplines avec comme fil conducteur et support un projet et une production.

Algorithmique et programmation en Mathématiques et Technologie

En technologie, la démarche de projet est inspirée de la démarche de conception des objets techniques utilisée en sciences de l'ingénieur outil SysmL et APTE

La réalisation du simulateur de conduite routière inclue :
- l'interfaçage à un ordinateur
- La mise en œuvre de l'imprimante 3D
- La sélection, la programmation et tests des situations de sécurité routière avec ou sans interfaçage.

Logiciel de programmation Scratch 2, développé au MIT universel et gratuit.
Logiciel de Tableur Grapheur
Logiciel mBlock (une version Scratch 2 pour l'interfaçage avec une carte électronique)
Logiciel de modelage volumique 3D Google Sketchup 8 gratuit.

Ateliers découvertes des métiers :
- un policier (sécurité routière)
- un ingénieur(programmation)

Classes de 4e ou 3e

Simulateur de conduite routière en 3 proposition d'un EPI en Mathématiques et Technologie

En Mathématiques les élèves travaillent sur le problème de la distance d'arrêt : C'est la distance que parcourt un automobiliste entre le moment où il voit un obstacle puis freine et le moment où son véhicule s'arrête.

Cette situation peut être travaillée dans le cadre de l'ASSR. Des documents sont disponibles sur les sites académiques. On peut consulter notamment les pages suivantes :

http://www.pedagogie.ac-nantes.fr/1166481205218/0/fiche___ressourcepedagogique/&RH=1160730502671
http://www4b.ac-lille.fr/~convergence/lille/securite/freinage d un véhicule en 3eme Maths.pdf
http://sciences-physiques.ac-dijon.fr/documents/college/SecuriteRoutiere/SecuriteRoutiere.htm

Sous certaines conditions, les formules ci-dessous donnent une valeur approximative de la distance d'arrêt D d'une voiture, exprimée en m, en fonction de sa vitesse V, exprimée en km/h.

• Voici la formule que l'on peut utiliser pour calculer la distance d'arrêt si le conducteur est lucide :

$$D = \frac{V}{6} + 0,007 \times V^2$$

• Voici maintenant la formule que l'on pourrait utiliser pour calculer la distance d'arrêt si le conducteur est peu lucide (fatigué, distrait par un appel téléphonique, ayant absorbé de l'alcool,…) :

$$D = \frac{V}{2} + 0,007 \times V^2$$

1) Compléter le tableau (arrondir les résultats au mètre près) :

Vitesse en km/h	50	90	100	110	130
Distance d'arrêt en m pour un conducteur lucide					
Distance d'arrêt en m pour un conducteur peu lucide					

2) Pour un conducteur lucide, la distance d'arrêt est-elle proportionnelle à la vitesse ?

3) Un conducteur roule à 100 km/h. Surgit un obstacle à 100 m de lui. Pourra-t-il s'arrêter à temps ?

4) Un conducteur lucide veut pouvoir s'arrêter en 10 mètres au maximum. Déterminer, à 1km/h près, la vitesse qu'il ne doit pas dépasser.

Commentaires :

À la question 1), le fait que le tableau comporte trois lignes peut empêcher certains élèves de distinguer les grandeurs pour lesquelles la question de proportionnalité se pose. Aide possible : faire construire deux tableaux.

La présence de la variable V à la puissance 2 n'est pas ici gênante pour le socle puisqu'il s'agit seulement de remplacer V par une valeur.

Pour la dernière question, il ne s'agit évidemment pas de résoudre une équation du second degré mais de trouver un encadrement par essais successifs.

Il est possible de compléter l'énoncé par les deux questions suivantes :

6) Comparer les distances d'arrêt pour un conducteur lucide avec celles relatives à un conducteur peu lucide.

7) En tant que futur automobiliste, que doit-on retenir de cette comparaison ?

Le problème fait partie de la banque de problèmes publiée par la DEGESCO
Outil numérique proposé : Tableur Grapheur

Avec un autre outil numérique le logiciel de programmation Scratch 2.0

Mathématiques

Mode Turbo

Cette activité est conduite en Mathématiques

vitesse : 120

donne moi une vitesse en km/h

$$D = \frac{V}{6} + 0,007 \times V^2$$

distance d'arrêt pour conducteur lucide 120.8

$$D = \frac{V}{2} + 0,007 \times V^2$$

Pour répondre à la question 1
Le logiciel de programmation est utilisé comme un tableur

distance d'arrêt pour conducteur peu lucide 160.8

50

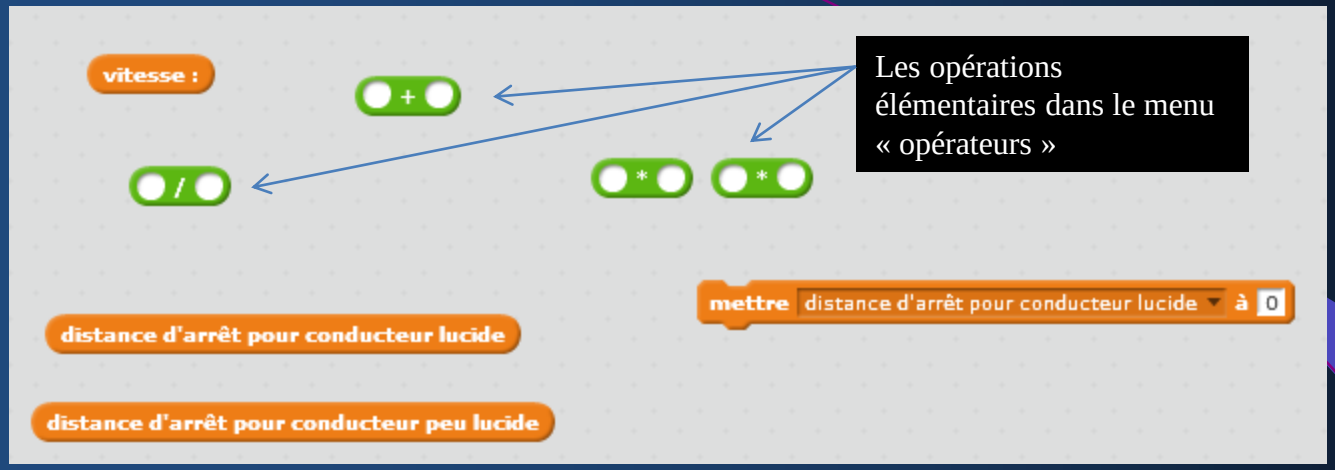
L'élève programme cette application qui va lui servir d'outil pour répondre à la question 1.

Il faut créer 3 variables :

Vitesse
Distance d'arrêt pour conducteur lucide
Distance d'arrêt pour conducteur peu lucide.

Les élèves construisent le programme

Mathématiques



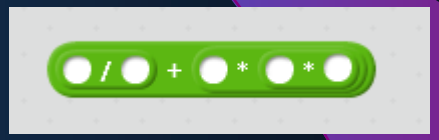
Il faut programmer les formules :

$$D = \frac{V}{6} + 0,007 \times V^2$$

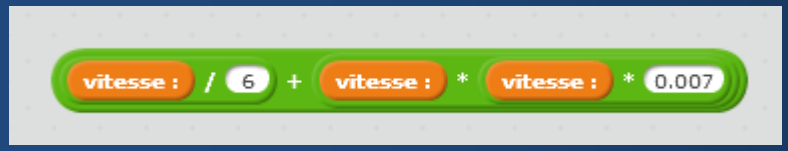
- Deux multiplications
- Une division
- Une addition

On glisse une multiplication dans une autre

On glisse la division dans le premier membre de l'addition



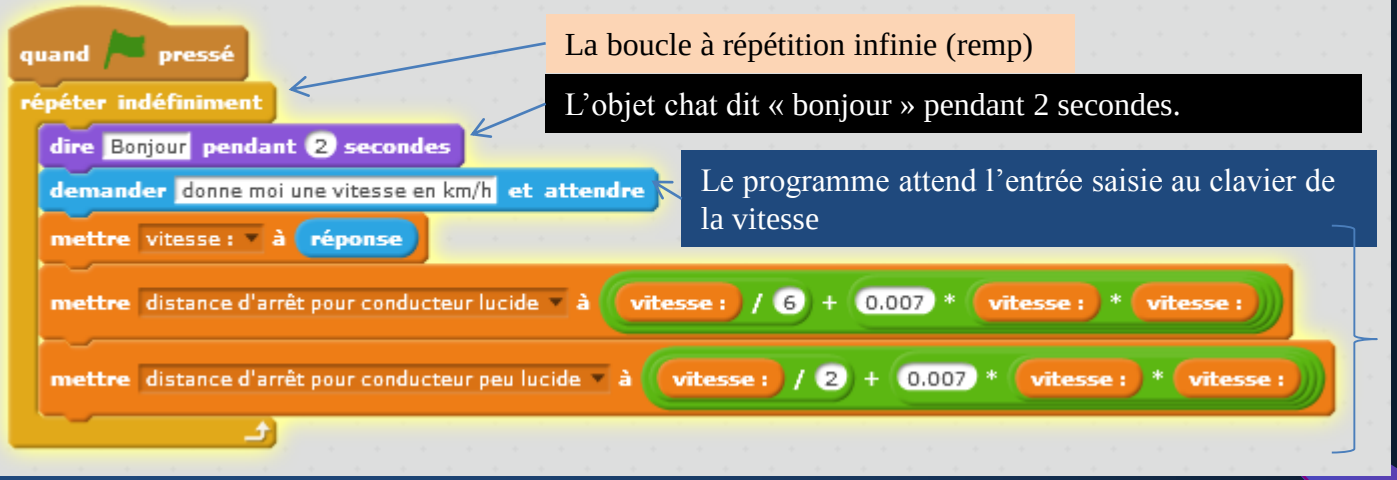
On glisse les multiplications dans le second membre de l'addition



On complète la formule avec les autres variables et les constantes.

On affecte de cette formule la variable distance d'arrêt pour conducteur lucide en la glissant dedans.

Même démarche pour l'autre variable.



La boucle à répétition infinie (remp)

L'objet chat dit « bonjour » pendant 2 secondes.

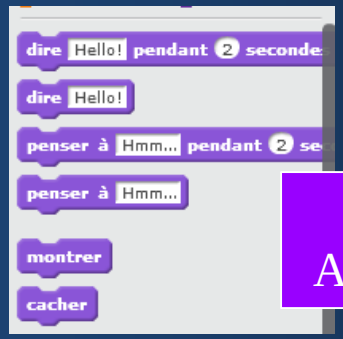
Le programme attend l'entrée saisie au clavier de la vitesse

Mathématiques

On construit un programme permettant de faire des calculs à partir d'une vitesse saisie.

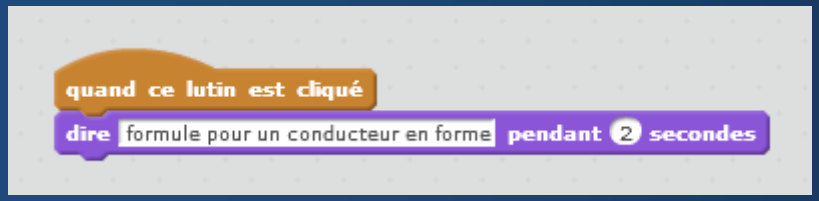
La séquence d'instructions suivante effectue les deux calculs.

Outils Contrôles

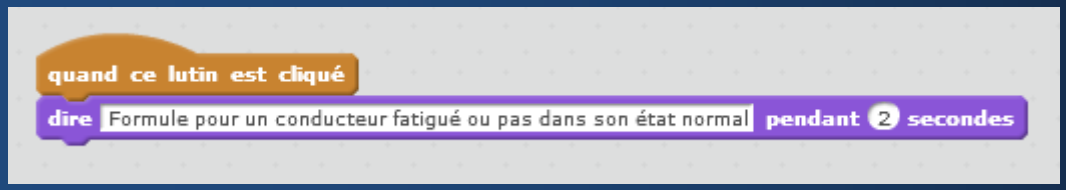


Outils Apparence

Outils Opérateurs



$$D = \frac{V}{6} + 0,007 \times V^2$$



$$D = \frac{V}{2} + 0,007 \times V^2$$

On peut compléter le programme en faisant « parler » les formules.

En technologie les élèves proposent et modélisent des situations de sécurité routière faisant intervenir la distance de freinage

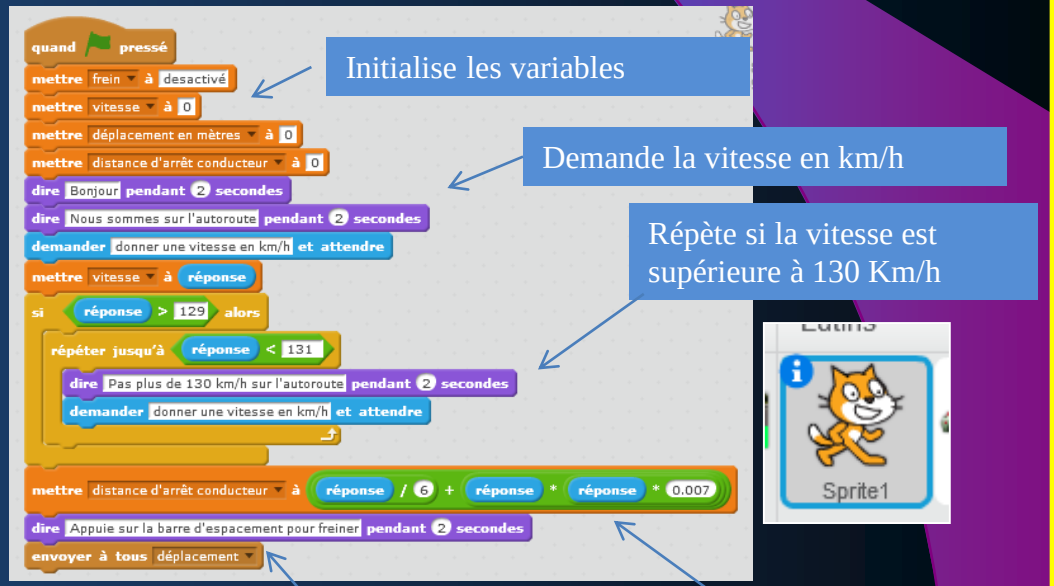
Technologie

La scène est composée de 2 images.

3 variables sont à définir.

Il faut éviter l'accident

Script pour simuler l'arrêt d'urgence du véhicule devant.



Initialise les variables

Demande la vitesse en km/h

Répète si la vitesse est supérieure à 130 Km/h

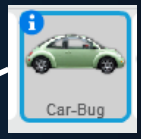
Envoie un message pour exécuter un sous-programme

Calcule la distance d'arrêt

3 programmes ou scripts sont associés aux objets.

Algorithme détaillé de gestion du déplacement du véhicule

Technologie



Positionne le véhicule à gauche de l'écran

Boucle à répétition infinie

Adapte à l'écran le déplacement du véhicule

Teste si le véhicule entre en contact avec le véhicule à l'arrêt

Si la condition du test est vrai, affiche le message d'accident.

Teste si la touche espace est pressée pour simuler le freinage

Change l'état du frein

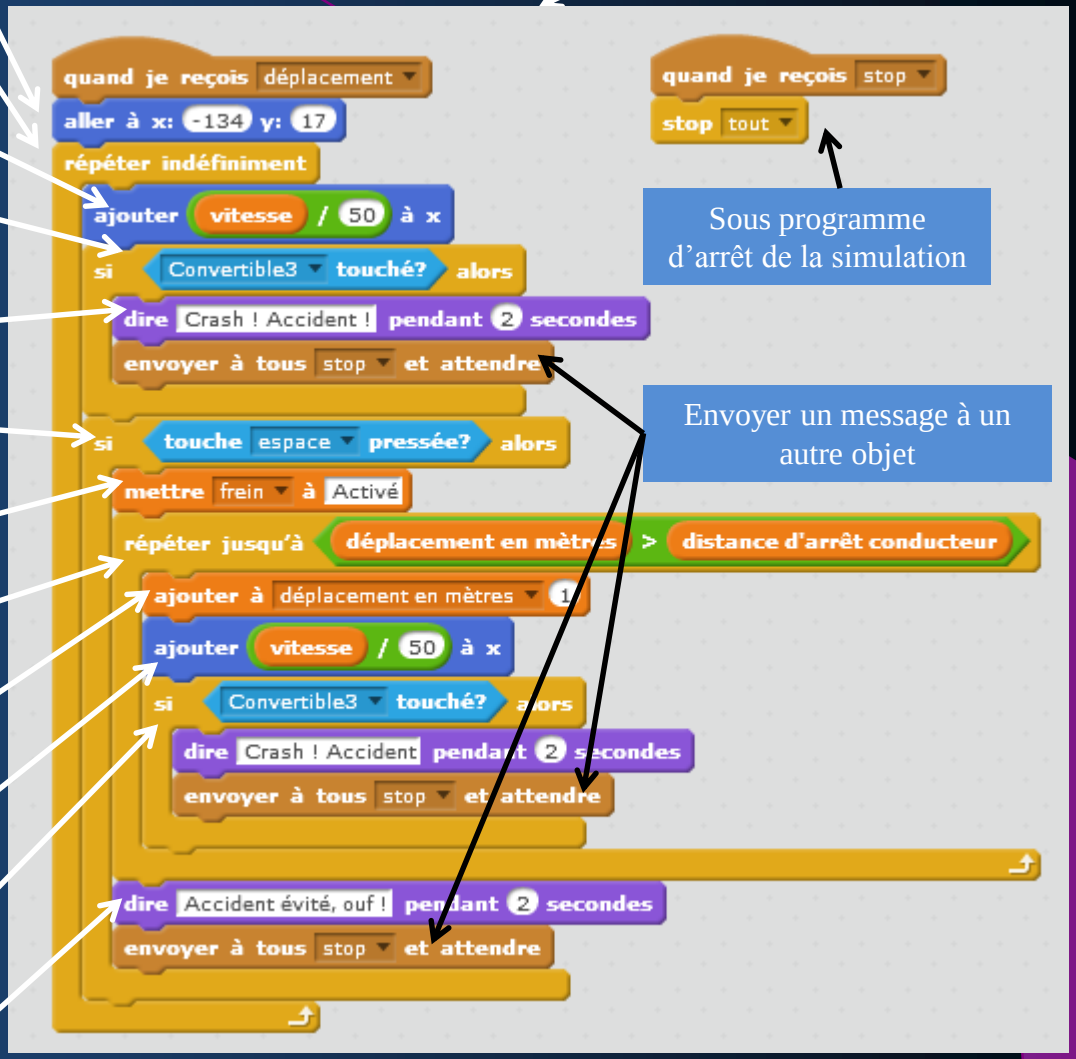
Continue de faire avancer le véhicule de la distance de freinage calculée

Montre et incrémente la distance après freinage

Adapte à l'écran le déplacement du véhicule pendant le freinage

Teste si le véhicule entre en contact avec le véhicule à l'arrêt pendant le freinage

Si il n'y a pas de contact avec le véhicule, affiche que l'accident est évité



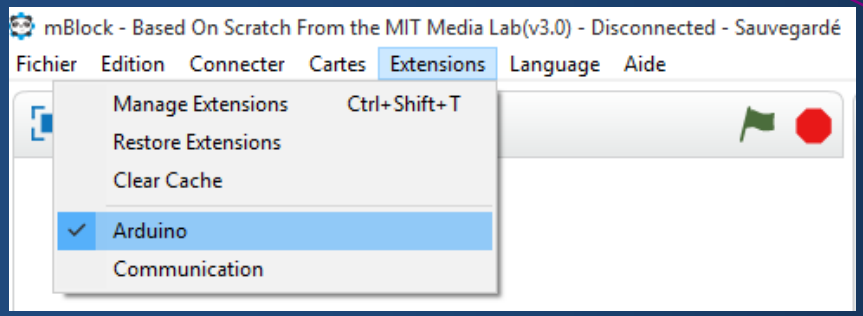
Sous programme d'arrêt de la simulation

Envoyer un message à un autre objet

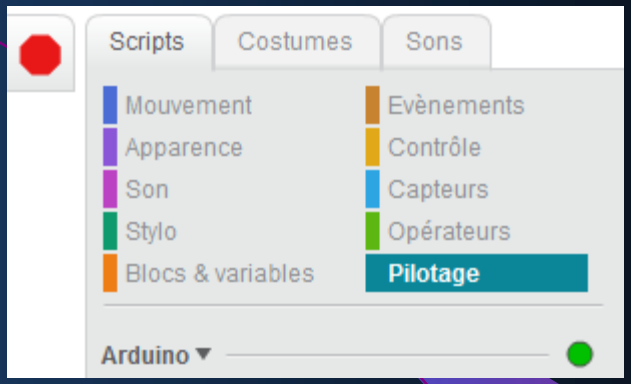
Connexion d'une carte Arduino de type « Uno » dans le logiciel mBlock

Technologie

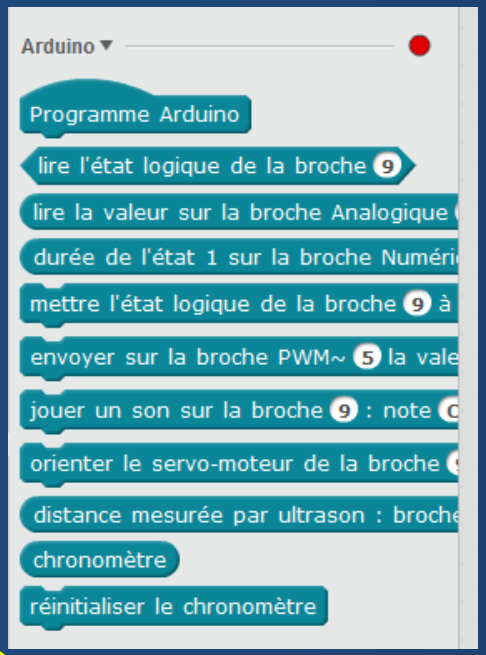
Un clic sur le menu « Extensions »

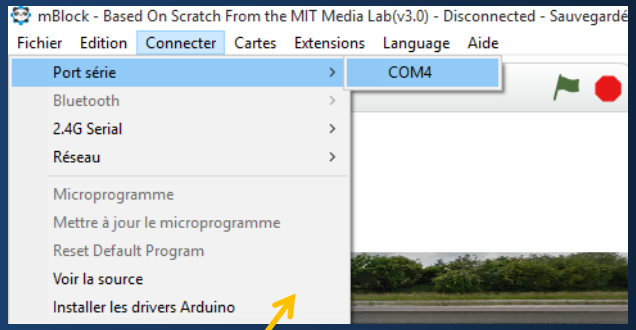


Un autre clic sur le menu « Robotique »

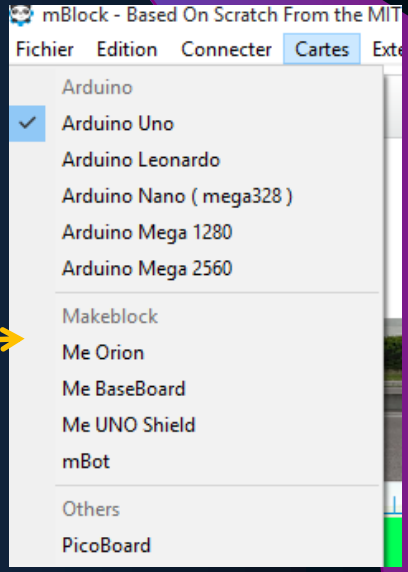


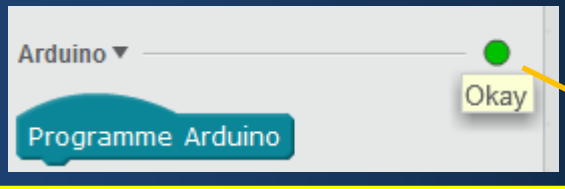
Les instructions relatives au matériel Arduino sont utilisables comme les blocs et objets de Scratch 2.





Pour connecter la carte, préciser le modèle et le port de communication utilisé

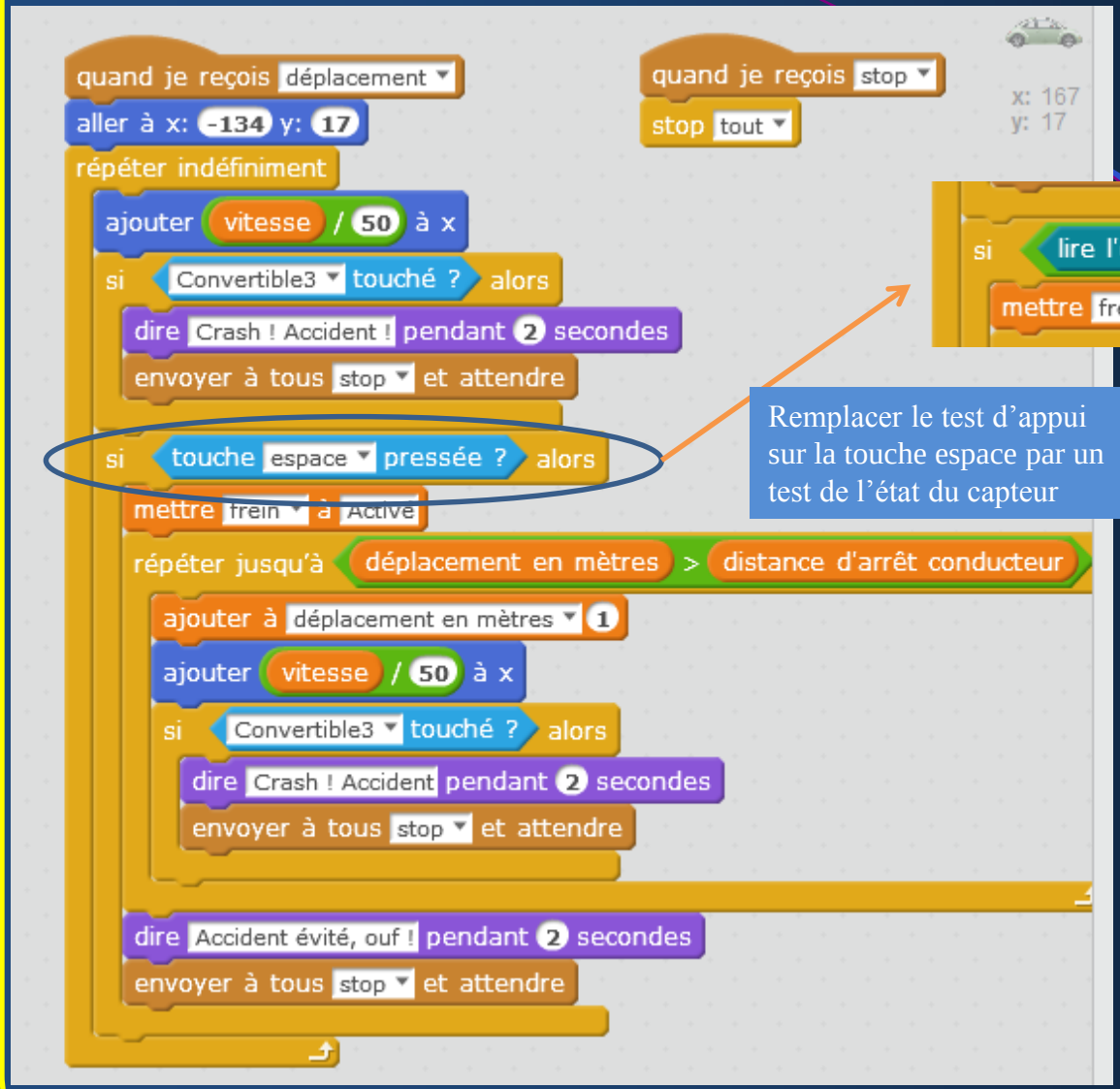




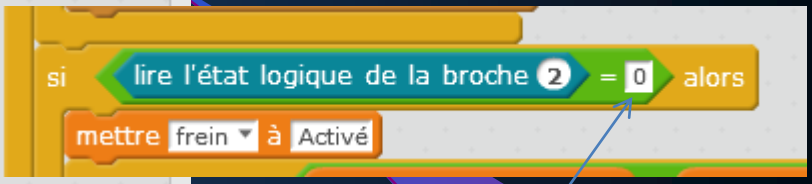
Le voyant vert indique que la connexion est établie.

Technologie

Modifier le déclenchement du freinage en appuyant sur la touche espace par une pression sur un capteur : un bouton poussoir pour commencer

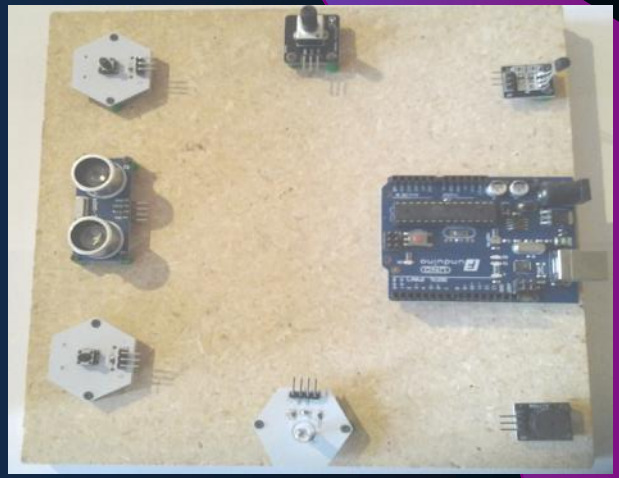


Remplacer le test d'appui sur la touche espace par un test de l'état du capteur



Le bouton poussoir doit être connecté sur la broche 2 de la carte

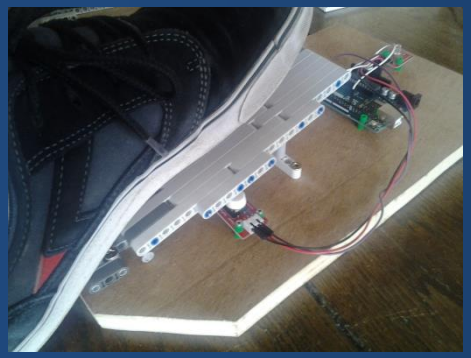
Si l'état du bouton poussoir passe de 1 à 0 alors la condition est vraie

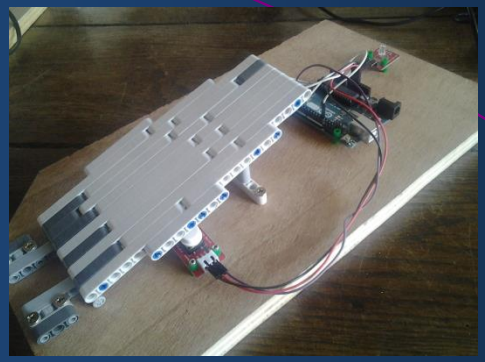


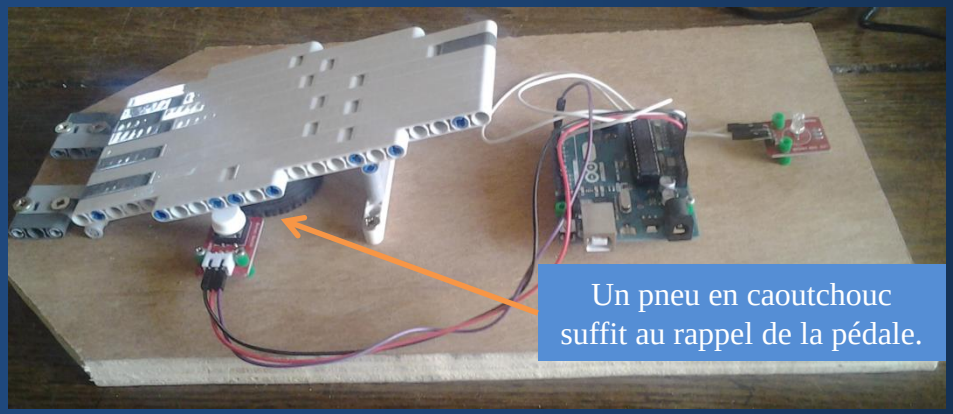
Un module Arduino va permettre de faire les essais avec différents capteurs

Prototype de pédale de frein et programme de test de la pédale.

Technologie



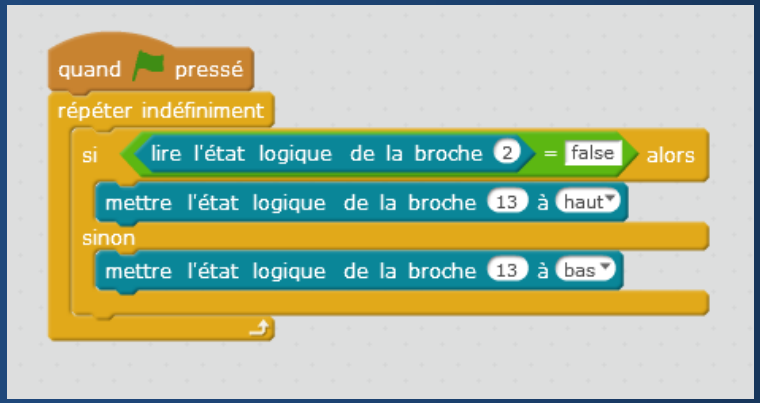




Un pneu en caoutchouc suffit au rappel de la pédale.

La pédale est réalisée à l'aide de quelques pièces de Lego, c'est un prototypage rapide. Pour la réalisation d'un simulateur de conduite plus complet (avec un volant) on peut envisager d'autres solutions techniques.

Un bouton poussoir est utilisé comme capteur de contact.
Une LED permet de faire des essais avec la carte électronique.



Le programme d'essais est composé d'une boucle à répétition infinie. Si le bouton poussoir change d'état (sa sortie est reliée sur l'entée 2) alors la LED s'allume, sinon elle reste éteinte.

La LED est reliée sur la sortie 13.