

Lever les doutes avec la barrière de parking

CHRISTOPHE VIÉ DAVIDAS*, BENOÎT PASCAL**

Faire des exercices de programmation n'est pas motivant pour les élèves s'ils n'ont pas un problème concret à résoudre. Le problème de la barrière de parking est simple, facile à comprendre et présente des difficultés progressives. Mais alors comment intéresser tous les élèves ? Quelles sont les étapes de programmation ? Et comment la fabriquer ?

Dans le numéro précédent de *Technologie* (n° 204), nous vous présentions comment débiter avec Arduino. Nos élèves avaient pu tester le principe de boucle, de valeur logique ou analogique, de condition, etc. Il faut maintenant leur donner un projet concret afin de mettre en application les notions abordées. Avec peu de budget, et pour tous les niveaux d'élèves, le projet de la barrière connectée non seulement ne démotivera pas les plus lents, mais il donnera aussi envie aux plus rapides d'explorer de nouvelles pistes.

Le constat de départ

Au collège, les élèves n'ont pas le choix de faire ou non de la technologie, contrairement au lycée. La technologie n'est donc pas source de motivation pour tous. Pour favoriser l'apprentissage et peut-être donner l'envie de poursuivre dans les filières technologiques, les élèves doivent être motivés. Au collège César-Franck de Paris, la très grande hétérogénéité de niveau des élèves nécessite de proposer des activités avec un objectif minimum à atteindre et d'autres permettant d'aller plus loin en accomplissant des tâches plus complexes. Dans le cas de la programmation, il est essentiel que les élèves soient acteurs et puissent travailler en très petit groupe sur un support technique simple et évolutif. Pour mettre en place cette stratégie, le choix a été de réaliser des boîtiers de programmation (carte Uno + base *grove*) indépendants et de multiplier le nombre de supports à moindre coût, comme cette barrière de parking, pour que chacun puisse avancer et progresser à son rythme. La rapidité et la facilité de mise en œuvre de la maquette permettent à travers une série de scénarios de plus en plus complexes d'acquérir des bases de programmation de façon ludique (voir encadré La maquette).

MOTS-CLÉS

programmation,
équipement didactique

* Professeur de
technologie au collège
César-Franck, Paris (75)

** Professeur de
technologie au collège
Eugène-Chevreul (94)

Progressivité d'apprentissage

Qui ne connaît pas la barrière de parking ? Tous les élèves ont pu la voir en situation au péage autoroutier ou à la caisse d'une pompe à essence, aux parkings souterrains, etc. **1** C'est un objet technique du quotidien de l'élève qui a un fonctionnement des plus basiques. La barrière se lève pour faire passer une voiture, puis se baisse. Mais la particularité de cette barrière est qu'elle peut avoir un nombre incalculable d'options, comme un clignotement lors du passage, un fonctionnement grâce à un code d'accès ou uniquement en présence d'une voiture, sans heurter le véhicule lors de la descente, etc. Toutes ces options complexifient le fonctionnement et, par la même occasion, permettent d'apprendre à coder avec une progressivité naturelle. On peut également connecter cet objet à un smartphone, si on ajoute un module Bluetooth. Les élèves les plus motivés pourront alors trouver leur compte en perfectionnant le programme du smartphone pendant que les moins aisés s'arrêteront à une application basique.

Les compétences des nouveaux programmes attendus en fin de séquence par tous les élèves sont :

2. Concevoir, créer, réaliser

CT 2.2. Identifier le(s) matériau(x), les flux d'énergie et d'information dans le cadre d'une production technique sur un objet et décrire les transformations qui s'opèrent.

4. Pratiquer des langages

CT 4.1. Décrire, en utilisant les outils et langages de descriptions adaptés : croquis, schémas, graphes, diagrammes, tableaux (représentations non normées).

5. Mobiliser des outils numériques

CT 5.1. Simuler numériquement la structure et/ou le comportement d'un objet.

Compétence spécifique au programme de technologie
CS 1.6. Analyser le fonctionnement de la structure d'un objet, identifier les entrées et sorties.

Séance 1 : découverte du fonctionnement

Pour atteindre tous ces objectifs, les élèves auront deux séances.

La maquette

Très peu de pièces à fabriquer et à acheter sont nécessaires pour construire la maquette.

Pièces fabriquées :

- lisse usinée sur Charly Robot dans du PVC expansé de 6 mm ;
- socle maquette découpé dans du PVC expansé 3 mm et pliage à chaud ;
- bloc motorisation imprimé en PLA sur Witbox ;
- plaque connectique découpée dans une plaque d'essais à bandes.

Pièces achetées :

- un capteur infrarouge d'évitement d'obstacle ;
- un servomoteur standard ;
- trois connecteurs *grove* droits ;
- une LED rouge Ø10 mm et une résistance ;
- une plaque d'essais à bandes.

Grâce au petit nombre de composants, le temps d'assemblage est très court. Le coût par maquette est inférieur à 15 €. De plus, la maquette à un fonctionnement très simple pour que l'aspect technique soit vite compris par les élèves. Tous les câblages sont en connectique *grove* afin d'éviter les problèmes de montage électrique lors des manipulations en classe. La carte Arduino et son *shield* sont indépendants de la maquette afin d'être réutilisés pour d'autres projets. Seuls les éléments à l'intérieur de la barrière ne sont pas réutilisables.

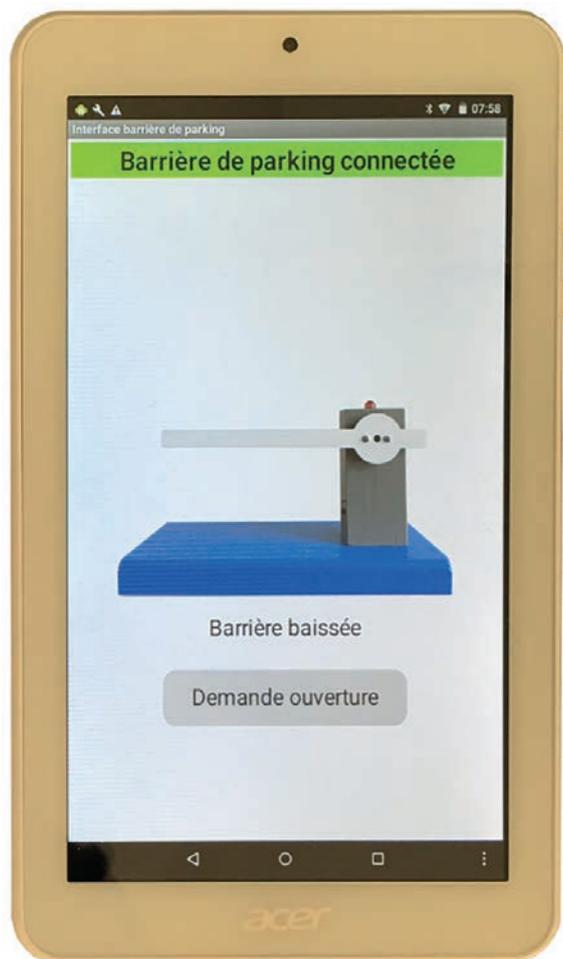
Tout l'intérêt de cette maquette est son évolutivité. Pour la prendre en main, les élèves ont pour objectif de lever la barrière pendant 10 secondes avant de la faire redescendre. Grâce à la présence d'un servomoteur, pas besoin de capteur « fin de course », seule la variation de l'angle suffit. Les élèves doivent néanmoins mettre un bouton-poussoir pour faire déclencher le programme. Simple et facile, ce scénario met tous les élèves en confiance avant de complexifier le système.



1 La maquette



2 Le programme



3 Application simple

Avant de se lancer tête baissée dans la programmation, les élèves vont devoir s'approprier le fonctionnement de la maquette, ainsi que ce qui la constitue. Le professeur explique la situation suivante : « Une voiture arrive devant la barrière, le conducteur appuie sur le bouton, la barrière se lève, reste levée 10 secondes, puis se rebaïsse. » Le professeur distribue une fiche activité avec une vue en coupe de la maquette afin que les élèves (qui ne peuvent pas ouvrir la maquette) puissent identifier ses composants. Les élèves suivent les consignes de câblage et peuvent faire fonctionner la maquette pour trouver le nom des composants et leur mode de fonctionnement.

Pour se préparer au brevet, les élèves disposent du programme Ardublock **2** et doivent reconstituer l'organigramme qui lui est associé. Ils peuvent là encore s'aider de la maquette en fonctionnement. Au départ, les élèves doivent partir d'une feuille blanche et tout recréer eux-mêmes. Pour les élèves bloqués, le professeur peut distribuer des « coups de pouce » en donnant la constitution de l'organigramme sans le contenu.

Le professeur fait remarquer par la suite que la lisse de la barrière n'est pas exactement horizontale en position fermée et pas exactement verticale en position ouverte. Les élèves ont pour mission de chercher dans les deux sous-programmes « ouverture » et « fermeture » les bonnes valeurs des servomoteurs afin d'améliorer le fonctionnement de la barrière. Ils en profitent pour décrire le fonctionnement d'un servomoteur et surtout pour expliquer quelle grandeur pilote le servomoteur. En bilan, les élèves expliqueront la différence entre un moteur et un servomoteur.

Maintenant, les élèves vont devoir simuler le passage d'une voiture qui calerait sous la barrière. Les élèves déduisent qu'il y a un problème de programmation, puisque la lisse s'abaisse, quoiqu'il arrive, sur le véhicule. Par équipe, ils doivent trouver une solution pour éviter ce problème. En s'aidant du descriptif des composants de la maquette, ils tomberont sur la nécessité de rajouter au programme une détection de présence. À eux de trouver comment intégrer le clignotement avec la non-présence de véhicule. Cette étape est l'une des plus compliquées, car les élèves doivent trouver la fonction « tant que » et bien l'utiliser avec le clignotement. Comme pour la fois précédente, un coup de pouce peut être donné par le professeur afin d'aider les moins aguerris avec la structure du programme complet.

En fin de séance, les élèves sont amenés à se poser des questions sur l'ouverture de la barrière par bouton-poussoir. Dans un parking, seules certaines personnes sont autorisées à entrer ; le bouton-poussoir n'est donc pas la solution idéale. Les élèves doivent proposer trois solutions techniques différentes pour filtrer le passage des automobilistes : elles seront testées en seconde séance.

Séance 2 : filtrage des personnes entrantes

En début de séance, le professeur fait un rappel des solutions proposées par les élèves au bilan. Si l'application sur smartphone n'est pas proposée, alors il la soumet. Il peut argumenter que seules les personnes qui ont l'application pourront entrer et donc que c'est un moyen de les filtrer. Les élèves proposent alors une problématique du type : « Comment créer

EN LIGNE

Retrouver toutes les activités, avec les tutoriels, sur le site de Christophe Vié Davidas
<http://www.cf-techno.com/sequence-13-activite-2>

Tous les liens sur <http://eduscol.education.fr/sti/revue-technologie>

une application mobile pour piloter la barrière de parking? »

Avant de programmer, les élèves doivent donner leur avis personnel sur la façon de résoudre la problématique. Ils doivent noter leurs questions, ainsi que la méthode pour y arriver. Puis, en classe entière, on note les solutions techniques proposées, la procédure de création de l'application et on liste toutes les questions. En bilan, le professeur généralise la mise en commun sur le problème de la maquette comme ceci :

« Pour créer une application mobile afin de piloter la barrière, il faut :

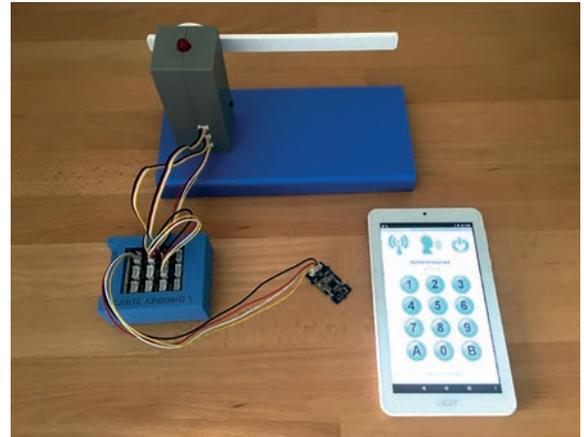
- identifier la solution technique employée pour communiquer entre le mobile et la barrière;
- définir le fonctionnement attendu de l'application;
- utiliser un logiciel de programmation d'application MIT APP Inventor 2;
- modifier le programme de la barrière pour prendre en compte cette évolution. »

Les élèves « choisissent » d'établir une liaison Bluetooth entre la barrière et l'application mobile. Ils listent les trois points à modifier de l'ancien programme de la barrière afin de remplacer le bouton-poussoir par l'application mobile :

- connexion du module de réception Bluetooth sur la carte Arduino;
- modification du programme pour remplacer le bouton-poussoir par la liaison Bluetooth;
- création de l'application mobile.

Les élèves doivent ensuite définir le fonctionnement de la barrière en décrivant la communication entre l'application et la barrière. Pour bien comprendre, ils doivent aussi faire un croquis de l'application avec ses boutons et expliquer ce qui se passe lorsque l'on appuie sur un bouton. Cette étape est très importante, car en dessinant l'écran du portable ils vont commencer sans s'en rendre compte à définir les actions de leur application. Plus ils mettront de bouton ou autres, plus le programme sera complexe. Le professeur sera là pour aider la logique des élèves en expliquant comment changer une image avec la fonction « caché » ou « visible ».

Il est impossible pour les élèves d'inventer le langage de programmation d'une connexion Bluetooth; d'ailleurs, nous, professeurs, en serions aussi incapables. Un tutoriel pas à pas est donné aux élèves afin de faire leur première application mobile (voir En ligne). Pour les élèves les moins à l'aise avec la programmation, il leur sera demandé de seulement suivre le tutoriel et de réaliser une application à deux boutons **3**. Pour les meilleurs, ils devront ajouter lors de la levée de la barrière un signal sonore avec une voix disant « Ouverture de la barrière » et faire la même chose lors de la descente. Enfin, pour les plus à l'aise et surtout les plus motivés, ils pourront



4 Application à code

ajouter un écran de déverrouillage de l'application avec un code à quatre chiffres à faire depuis un clavier numérique **4**.

Conclusion

En deux séances, les élèves auront su modifier un programme piloté localement par un programme à distance. Les plus aisés avec la programmation auront pu tester leur potentiel sur des applications de plus en plus complexes à programmer. ■