

# SAFARI DES OBJETS CONNECTÉS

## CONCEPTION D'OBJETS INTELLIGENTS

thématique : mobilité durable, transport et régulation



## Introduction

Ce protocole vise à **co-concevoir des objets intelligents**, qui sont des **objets intelligents** dans le contexte des villes intelligentes, capables de **percevoir l'environnement extérieur via des capteurs, de traiter des données observées et d'effectuer des actions via des actionneurs**.

Les objets intelligents peuvent également **interagir entre eux ou avec les humains**. Lors de la conception, il est important d'inciter les participants à réfléchir à la **structure de l'objet intelligent et à son interaction avec l'environnement extérieur**.

Les ressources fournies dans ce document comprennent des lignes directrices pour la réalisation du protocole avec des élèves de différents âges et niveaux, détaillant les activités proposées avec des phases, une durée et des exemples pratiques pour permettre aux modérateurs ou aux éducateurs intéressés de reproduire les ateliers proposés.

Ce document comprend également des **outils d'évaluation permettant de vérifier le niveau d'implication et d'apprentissage** obtenu à la suite des activités menées et l'impact sur la sensibilisation, sur la nature des objets intelligents et leur rôle dans les villes intelligentes.

La structure proposée a été validée par des chercheurs de plusieurs universités italiennes avec le soutien de l'association à but non lucratif Perlatecnica ETS et avec une contribution particulière du Département d'Informatique de l'Université de Salerne et de l'Université Libre de Bolzano, obtenant de bons résultats en termes d'implication et d'apprentissage, résultats qui seront détaillés à la fin du manuel.

### Disciplines



**technologie et ingénierie**  
**physique et chimie**  
**art et design**

### Objectifs de développement durable





# L'activité en bref

## Structure du protocole

Le protocole Safari des objets connectés est divisé en deux phases complémentaires qui s'appuient l'une sur l'autre pour créer une compréhension globale de la conception collaborative d'objets intelligents. Chaque phase intègre plusieurs sous-objectifs, présentés ci-dessous :

### Étape 1 - Familiarisation et idéation

1

#### Familiarisation



Comprendre les termes, les composants élémentaires (capteurs, actionneurs) et les mécanismes d'interaction



2

#### Idéation



Proposer un objet intelligent, en tenant compte de la structure et des mécanismes d'interaction



### Étape 2 - Conception et prototypage

1

#### Conception



Mise en œuvre de la logique : acquisition des données par le biais de capteurs, traitement et sortie par le biais d'actionneurs



2

#### Prototypage



Création d'un prototype physique ou d'une simulation de l'objet intelligent



**Étape 1 - Familiarisation et idéation.** Au cours de cette étape, les participants sont invités à se familiariser avec les concepts fondamentaux de ce projet et, par la suite, à proposer une idée de l'état embryonnaire de l'objet qu'ils souhaitent concevoir. Les sous-objectifs sont les suivants :

1. La **familiarisation** avec les termes, les composants élémentaires d'un objet intelligent, tels que les capteurs et les actionneurs, et les mécanismes d'interaction de ces éléments analysés isolément
2. **L'idéation** dans laquelle les participants sont invités à proposer un objet intelligent.



Selon l'âge des participants, il peut être nécessaire de définir un périmètre et/ou une mission qui contextualise les objets à concevoir. Par exemple, les élèves peuvent être invités à formuler des hypothèses sur la manière de rendre une maison ou un parc intelligents, ou encore une école ou un gymnase. Ils peuvent être invités à concevoir des objets qui assurent la sécurité, surveillent la santé de l'environnement ou rendent un espace partagé interactif et ludique. Alors que les premières sont des propositions de domaines d'application, les secondes sont des propositions de missions. Il est important de réfléchir à la structure de l'objet et au mécanisme d'interaction dès la phase de conception.

**Étape 2 - Conception et prototypage.** La deuxième étape est entièrement consacrée à la mise en œuvre, via des ateliers pratiques en groupe, combinant de courtes présentations des concepts par une approche frontale, permettant aux élèves de mettre immédiatement en pratique le concept introduit. Ils peuvent construire le projet progressivement, au fur et à mesure de la présentation des concepts, selon une approche progressive. Lorsqu'ils

estiment qu'il n'est pas nécessaire d'introduire le concept introduit dans leur projet, ils sont invités à proposer un exemple simple pour démontrer sa clarté et justifier l'inutilité de son exploitation dans l'objet conçu. La complexité des exemples proposés lors de la partie introductive doit être incrémentale, de la version la plus simple (par exemple, sans boucles) aux concepts les plus complexes. En cas de manque de temps, les concepts doivent être introduits en privilégiant ceux qui permettent l'implémentation d'objets intelligents.

Les sous-objectifs sont les suivants :

- La **conception** de la logique objet en mettant en œuvre l'acquisition des « entrées » via des capteurs, leur traitement et la détermination de la réaction en termes de sortie externalisée via des actionneurs,
- Le **prototypage** de l'objet, au moyen de prototype physique ou de simulation.

---

## Pour bien démarrer

**Durée :** La durée totale des activités est d'au moins 2 heures par jour pendant 3 jours, comme vérifié lors d'activités modérées par des chercheurs et des éducateurs de l'Association Perlatecnica et des Universités de Salerne et de l'Université libre de Bolzano.

**Niveau de difficulté :**



**Matériel nécessaire :** Les outils utilisés peuvent être hybrides (numériques + physiques), privilégiant la composante physique, notamment lors des phases de familiarisation et de prototypage.

**Quelques conseils d'organisation :**

- La phase de **familiarisation** peut être mise en œuvre de manière individuelle ou collaborative, en adoptant une approche magistrale traditionnelle ou en privilégiant l'interaction. Par exemple, une approche ludique ou une exploration linguistique axée sur l'étymologie des mots pour maîtriser le vocabulaire est recommandée. Une phase physique, où la structure des différents composants est présentée, doit être incluse.
- La phase de **conception** peut être entièrement physique, menée individuellement ou en collaboration. Elle privilégie la représentation de l'objet en le rendant intelligent, en activant des capteurs et des actionneurs, en simulant le comportement par des scénarios SI... ALORS... et en contextualisant l'interaction grâce à des techniques narratives. Elle peut également être purement numérique, en adoptant des langages de programmation textuels ou par blocs, selon l'âge et les compétences des élèves concernés. Si les élèves du secondaire peuvent aborder le C ou le C++ comme langages de programmation, les élèves du primaire ou les débutants devraient privilégier des langages visuels comme Make Code, qui minimisent la complexité syntaxique et permettent aux participants de se concentrer sur l'implémentation logique de l'objet intelligent.
- La phase de **prototypage** doit être aussi physique que possible afin de permettre aux participants de mettre la main à la pâte, d'acquérir une expérience concrète de la construction de circuits (simples) et de vérifier dans quelle mesure l'objet réalisé correspond à l'hypothèse. Une approche collaborative est suggérée pour favoriser la réflexion et le brainstorming.

# Glossaire

Mots-clés/Concepts	Définitions
<b>Actionneur</b>	Composant d'un objet intelligent qui exécute une action physique en réponse à des données ou à des signaux (par exemple, un moteur, une lumière, un haut-parleur).
<b>Atelier pratiques</b>	Une méthode pédagogique qui combine des présentations théoriques avec une mise en pratique immédiate, favorisant l'apprentissage par la réalisation de projets concrets.
<b>Capteur</b>	Un dispositif intégré dans un objet intelligent, capable de mesurer des données environnementales (par exemple, la température, la lumière, la pression) pour informer le comportement de l'objet.
<b>Conditions</b>	Un ensemble de conditions combinées à l'aide d'opérateurs logiques (ET, OU) pour définir des comportements plus avancés en fonction de plusieurs critères.
<b>Cycles / Boucles</b>	Un concept de programmation utilisé pour répéter une action ou un comportement plusieurs fois (par exemple, onstart, forever ou repeat dans MakeCode).
<b>MakeCode</b>	Une plateforme de programmation visuelle utilisée pour développer les comportements d'objets intelligents, principalement via des blocs qui représentent des fonctions de codage.
<b>Objets intelligents</b>	Objets physiques équipés de capteurs et d'actionneurs leur permettant de collecter, d'interpréter et de répondre aux données environnementales.
<b>Raconter des histoires</b>	Une technique utilisée pour décrire et contextualiser les idées d'objets intelligents à travers de courts récits, facilitant ainsi la compréhension de leurs interactions avec l'environnement.
<b>Véhicules autonomes</b>	Véhicules capables de se déplacer sans intervention humaine, grâce à des capteurs, des algorithmes et des systèmes intelligents intégrés.



# Protocole

## Étape 1 - Familiarisation et idéation



**Contexte et description du problème à résoudre dans cette étape :** Au cours de cette première étape, les élèves se familiariseront avec les concepts de base qui permettent de découvrir le protocole Safari des objets connectés et, par la suite, proposeront une première idée pour l'objet qu'ils souhaitent concevoir.

**Objectifs d'apprentissage :** Acquérir la terminologie nécessaire pour passer à la phase d'idéation. Termes à maîtriser : objet intelligent, capteur, actionneur, exemples de capteurs, exemples d'actionneurs. Stimuler l'imagination en inventant de nouveaux objets intelligents basés sur les capteurs et actionneurs disponibles.

## Conceptualisation

Avant de commencer les activités, il est nécessaire de recueillir des informations **sur la perception et le niveau de connaissances des élèves concernant les villes et les objets intelligents, ainsi que sur leurs attentes**. Cette évaluation préalable permettra d'estimer ultérieurement l'impact des activités proposées et d'évaluer le protocole.

Utilisez des activités stimulantes pour transformer la pré-évaluation traditionnelle en expériences participatives. Par exemple :

- Envisagez de mettre en place une **évaluation par émojis**, où les élèves sélectionneront des expressions exprimant leur niveau de confort avec les sujets spécifiques que vous aborderez. Par exemple, demandez-leur d'afficher des émojis exprimant leurs sentiments à l'égard de la programmation par blocs, des concepts de ville intelligente ou de leur familiarité avec les objets connectés. Cette représentation visuelle révèle rapidement les attitudes des élèves envers chaque matière.
- Vous pouvez également essayer un **exercice de positionnement des connaissances** en désignant différentes zones de votre classe comme zones de connaissances, du niveau débutant à expert. Le déplacement physique des élèves vers ces zones permet d'obtenir une représentation visuelle immédiate de la compréhension collective de chaque sujet abordé.
- Un **tableau interactif des attentes** peut également être utilisé : les élèves peuvent y coller des post-its indiquant leurs compétences actuelles et leurs objectifs à la fin de l'unité. Ce tableau devient un document évolutif consultable tout au long de votre enseignement pour suivre les progrès et célébrer les progrès.

Ces stratégies d'évaluation interactives fournissent des informations précieuses de manière attrayante, vous aidant à identifier les lacunes dans les connaissances tout en établissant un environnement d'apprentissage positif dès le départ.

## Investigation par les élèves

### **Familiarisation - Création de connaissances fondamentales**

Durant cette première phase, les élèves doivent acquérir la terminologie nécessaire pour poursuivre l'idéation. Les enseignants peuvent présenter la structure des activités proposées (durée, modalités, outils utilisés, attentes), et introduire les objets connectés en répondant à des questions telles que :

### 1. Questions d'introduction



**Vous êtes-vous déjà demandé ce qu'est un objet intelligent ? Qu'est-ce qui le rend intelligent ?** Son intelligence réside dans sa capacité à réagir de manière autonome à ce qu'il perçoit autour de lui : son environnement, ses interactions avec les personnes et les autres objets.

**Tous les objets peuvent-ils devenir intelligents ?** Potentiellement oui, s'ils sont dotés d'un mécanisme leur permettant de réagir de manière autonome aux stimuli externes.

### 2. Exemples d'objets intelligents



**Une machine à laver est-elle intelligente ?** Une machine à laver peut être intelligente si elle peut adapter la durée du cycle de lavage à la charge ou au degré de saleté des vêtements.

**Les stores peuvent-ils devenir intelligents ?** Oui, s'ils ajustent leur comportement en fonction de l'intensité lumineuse.

### 3. Quels composants constituent un objet intelligent ?



**L'objet que nous souhaitons rendre intelligent + un ou plusieurs capteurs pour percevoir les stimuli externes + Un ou plusieurs actionneurs pour réagir.** Les capteurs peuvent être comparés aux sens humains, tandis que les actionneurs peuvent être comparés aux membres ou à la voix que les humains utilisent pour effectuer des actions.

### 4. Capteurs que nous pouvons utiliser



Thermomètre pour mesurer la température

Accéléromètre pour détecter et mesurer l'accélération d'un objet en mouvement

Capteur d'humidité

Boussole pour gérer l'orientation

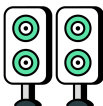
Capteurs infrarouges pour déterminer la distance par rapport à un agent externe

Capteurs de contact pour l'interaction avec un agent externe

Capteurs de lumière

**Et bien d'autres...**

### 5. Actionneurs que nous pouvons utiliser



Des haut-parleurs pour transmettre un message

Des écrans pour afficher un message

Des moteurs

**Et bien d'autres...**

Il est conseillé de combiner les définitions avec des exemples concrets d'objets intelligents pour rendre la phase de familiarisation plus concrète (et digeste).

### Processus d'idéation

Cette phase vise à stimuler l'imagination en inventant de nouveaux objets intelligents basés sur les capteurs et actionneurs disponibles.

À ce stade, il est crucial de faire un atelier pratique, permettant aux élèves de travailler individuellement ou en groupe en répondant à la mission donnée lors de l'étape de conceptualisation : « **Concevoir des objets intelligents qui aident ou entravent les véhicules autonomes** ».

Grâce à l'activité de familiarisation réalisée lors de la phase de conceptualisation, les élèves maîtrisent la terminologie et les éléments de base. Les enseignants peuvent ensuite demander aux élèves de proposer des objets **intelligents en termes de représentation, en détaillant leur structure et leurs composants**, en décrivant leur comportement en termes de conditions « QUAND/SI... ALORS... » et en détaillant l'interaction avec d'autres objets, véhicules et humains au **moyen de courts récits**.

Les élèves doivent disposer du matériel ou des instructions nécessaires à la réalisation de la tâche. Cette phase peut être réalisée comme une **activité débranchée, en racontant la structure et l'interaction avec des outils traditionnels, comme le papier et le crayon, éventuellement accompagnés de fiches**. La narration peut également être réalisée par une **approche numérique utilisant les outils de narration et d'archivage disponibles**, comme le projet Storylet (<http://www.isislab.it:19984/il-progetto-storylet>).

Afin de faciliter la tâche des élèves, notamment des plus jeunes et des moins expérimentés, il est important de déterminer le contexte et/ou la mission à laquelle les objets proposés doivent répondre. Dans le cadre des initiatives du projet SteamCity, la mission proposée est de « **Concevoir des objets intelligents qui facilitent ou entravent la conduite des véhicules autonomes** ».

### Exemple d'une approche Unplugged de l'activité

Dans cette activité, les élèves concevront des objets intelligents pour les environnements urbains, en se concentrant plus particulièrement sur les objets pouvant assister ou entraver les véhicules autonomes. L'activité utilise une approche autonome avec des tableaux d'idéation et des jeux de cartes, permettant aux élèves d'aborder des concepts technologiques sans ordinateur.

Avant le cours, préparez un **tableau d'idéation** pour chaque élève et trois jeux de cartes : **des objets pouvant être rendus « intelligents », divers capteurs et différents actionneurs**. Vous pouvez inclure des feux de circulation, des bancs, des poubelles et des panneaux de signalisation ; des capteurs de proximité, de température et de luminosité dans le jeu de capteurs ; et des LED, des haut-parleurs et des moteurs dans le jeu d'actionneurs.



Commencez par vérifier que les élèves ont bien compris les concepts de base des objets intelligents, des capteurs et des actionneurs, tels qu'ils ont été abordés dans les leçons précédentes. Présentez ensuite l'activité avec une histoire contextuelle :




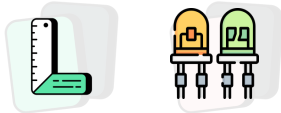
Hier, j'ai reçu un appel du maire de notre ville qui nous demandait si nous pouvions contribuer à la conception d'objets intelligents pour faciliter ou entraver la circulation des véhicules autonomes. Le maire nous a accordé une liberté totale pour créer des objets allant du divertissement à la gestion du trafic, en passant par les bornes d'information et les systèmes d'intervention d'urgence.



#### Présentation du tableau d'itération

<b>OBJET</b> 	Description du comportement d'un objet sous forme d'histoire décrivant qui/quoi déclenche le comportement et la réaction de l'objet
<b>CAPTEUR</b> <b>ACTIONNEUR</b> 	Description du comportement avec les termes <b>QUAND &lt;CONDITION&gt; ALORS &lt;REACTION&gt;</b>

#### Exemple de tableau d'itération

<b>FEU DE CIRCULATION</b> 	Le feu de circulation interagit avec les véhicules autonomes. À l'approche d'un véhicule, il cède le passage aux autres véhicules, arrêtant ainsi les piétons.
<b>DISTANCE</b> <b>LED</b> 	QUAND un véhicule s'approche ALORS afficher le feu jaune pour les piétons pendant 10 secondes, afficher le feu vert pour le véhicule pendant 60 secondes afficher le feu rouge pour le véhicule afficher le feu vert pour les piétons

Pendant que les élèves travaillent sur leurs propres créations, circulez dans la salle pour les guider. Encouragez-les à réfléchir aux deux aspects de la description du comportement. Le format structuré QUAND-ALORS garantit la **cohérence logique** de leur création, tandis que la description narrative les aide à prendre en **compte les interactions réelles**.


Certains élèves pourraient avoir du mal à visualiser l'interaction de leur objet avec l'environnement. Posez-leur des questions incitatives telles que : « **Qui ou quoi utiliserait cet objet ?** » ou « **Quel problème cela résout-il pour les personnes ou les véhicules ?** » D'autres pourraient avoir du mal à traduire le récit en règles structurées. Aidez-les en

leur demandant : « **Quelle condition précise déclenche ce comportement ?** » et « **Que se passe-t-il étape par étape ?** ».

Pour les élèves qui terminent rapidement, proposez de créer des objets plus complexes qui répondent à de multiples conditions. Par exemple, un lampadaire intelligent qui s'atténue lorsqu'il n'y a personne à proximité pour économiser de l'énergie, mais s'allume lorsqu'il détecte un mouvement et clignote en cas d'urgence. Si le tableau d'idéation semble trop contraignant pour certains élèves, proposez-leur **une feuille blanche**. L'important est qu'ils prennent en compte à la fois les composants (objet, capteurs, actionneurs) et le comportement (narratif et structuré).

Bien que cet exemple se concentre sur les objets intelligents pour véhicules autonomes, la même approche s'applique à d'autres contextes. Vous pouvez mettre les élèves au défi de concevoir des objets pour un parc intelligent, une école intelligente ou une maison intelligente. Il suffit d'adapter les cartes d'objets disponibles au nouveau contexte. L'approche de la double description reste pertinente dans tous les contextes. **La description narrative aide les élèves à relier leur conception aux besoins et expériences humains, tandis que la description structurée les prépare à la phase de programmation qui suivra.**

**Conseils pour les enseignants :** Lorsque vous guidez les élèves dans cette activité, gardez à l'esprit plusieurs considérations importantes :

- 
- **Privilégier les objets autonomes.** Les objets intelligents doivent être des systèmes automatiques qui perçoivent les conditions extérieures et réagissent en conséquence. De nombreux élèves, notamment ceux ayant une expérience des jeux vidéo, pourraient être tentés de concevoir des objets qui répondent principalement à l'interaction humaine directe plutôt qu'aux conditions environnementales. Orientez-les progressivement vers des systèmes véritablement autonomes.
  - **Aider les élèves à équilibrer la complexité de leurs conceptions.** Évitez les objets trop simples qui n'utilisent qu'un seul capteur et actionneur avec une réponse triviale. Face à ces conceptions élémentaires, encouragez les élèves à complexifier leurs conceptions en envisageant des conditions supplémentaires ou des réponses plus sophistiquées. À l'inverse, certains élèves peuvent concevoir des systèmes extrêmement complexes, comportant de nombreux composants et des comportements complexes. Dans ce cas, aidez-les à prioriser les comportements les plus importants.
  - **Assurer la cohérence entre les composants et le comportement.** L'un de vos rôles clés est de vérifier que les représentations d'objets des élèves (capteurs et actionneurs) correspondent logiquement à leurs comportements décrits. Si un comportement nécessite la détection de distance mais qu'aucun capteur de proximité n'est inclus, signalez cette incohérence.
  - **Envisager d'intégrer l'évaluation par les pairs en attribuant des idées aléatoires aux élèves et en demandant à chacun de présenter la conception d'un autre.** Cette approche encourage une écoute attentive lors des présentations et aide les élèves à appréhender leurs propres idées à travers le point de vue des autres.

## Restitution et réflexion

Prévoyez environ **15 minutes à la fin de la phase d'idéation pour la réflexion et le partage**. Recueillez les idées proposées et encouragez les élèves à présenter brièvement leurs projets à la classe. Cette séance de partage a plusieurs objectifs. Elle favorise la **réflexion grâce à un brainstorming collaboratif**, où les élèves peuvent observer comment d'autres ont abordé le même défi. La discussion mène naturellement à l'évaluation des améliorations ou modifications possibles des projets. Les élèves et l'enseignant peuvent ensemble évaluer la faisabilité de la mise en œuvre de chaque projet lors des phases ultérieures.

Cette phase de partage vous permet également **d'harmoniser la complexité des projets en fonction des compétences de vos élèves, des outils dont vous disposez et des contraintes pratiques telles que le temps**

**imparti ou la disponibilité des capteurs.** Si certains projets vous semblent trop ambitieux compte tenu du contexte de votre classe, c'est le moment de les peaufiner.

Cette activité relie la compréhension conceptuelle à la mise en pratique. En réfléchissant à la fois en termes d'histoires (comment leur objet s'intègre dans le monde) et de règles structurées (comment il sera programmé), les élèves développent des compétences de pensée créative et computationnelle.

## Étape 2 - Conception et prototypage



**Contexte et description du problème à résoudre dans cette étape :** La deuxième étape est entièrement consacrée à l'implémentation, combinant de courtes présentations de concepts de programmation avec la mise en pratique de ces concepts. Les élèves peuvent construire le projet au fur et à mesure de la présentation, en utilisant une approche incrémentale. S'ils estiment qu'il n'est pas nécessaire d'intégrer le concept présenté dans leur projet, ils sont invités à proposer un exemple simple pour démontrer sa clarté et justifier pourquoi il n'est pas nécessaire de l'exploiter. La complexité des exemples proposés lors de la partie introductive doit être incrémentale, de la version la plus simple (par exemple, sans boucles) aux concepts les plus complexes. En cas de manque de temps, les concepts doivent être introduits en privilégiant ceux qui permettent l'implémentation d'objets intelligents.

**Objectifs d'apprentissage :** Acquérir ou consolider/affiner ses connaissances sur les concepts de programmation tels que les boucles, les conditions et les comparaisons. Appliquer les concepts présentés à l'implémentation de l'objet proposé.

### Conceptualisation

Cette phase de mise en œuvre fait suite à la séance d'idéation au cours de laquelle les élèves ont conçu des objets intelligents pour faciliter ou entraver le développement des véhicules autonomes. Lors de ces séances, les élèves donneront vie à leurs conceptions grâce à la programmation par blocs. L'approche pédagogique combine de brèves introductions aux concepts de programmation et des applications pratiques immédiates, permettant aux élèves de construire leurs projets progressivement au fur et à mesure de leur apprentissage de chaque nouveau concept.

Les activités sont structurées pour s'adapter aux différents niveaux d'expérience en programmation des élèves. Les plus expérimentés peuvent étendre leurs implémentations avec des comportements plus complexes, tandis que les débutants peuvent se concentrer sur les fonctionnalités de base. Tout au long du processus, les élèves sont encouragés à affiner leurs idées originales en fonction de ce qu'ils ont appris sur les possibilités d'implémentation.

**Table ronde d'ouverture.** Avant de découvrir l'activité, organisez une brève table ronde sur la programmation par blocs. Cela vous permettra d'évaluer les connaissances et les attitudes des élèves tout en créant une atmosphère collaborative. Posez des questions telles que : « **Avez-vous déjà utilisé la programmation par blocs ? Quelles plateformes avez-vous essayées ?** », « **Qu'est-ce qui vous semble facile ou difficile dans la programmation ?** » ou « **Quels types de projets aimeriez-vous réaliser avec la programmation ?** ». Cette discussion a plusieurs objectifs : elle permet de mettre en lumière la diversité des niveaux d'expérience de votre classe et d'aider les élèves à nouer des liens avec des pairs partageant les mêmes intérêts ou des compétences complémentaires. Elle permet également aux élèves plus expérimentés de partager leur enthousiasme avec les débutants. De plus, elle vous permet d'identifier les concepts nécessitant une attention particulière, tout en faisant de la programmation une activité collaborative plutôt que solitaire. Prenez note des réponses des élèves pour adapter votre accompagnement tout au long des séances de mise en œuvre.

**Définir le contexte.** Commencez par relier le travail de mise en œuvre à la session d'idéation précédente :



Nous allons maintenant passer à la phase de programmation pour implémenter vos idées d'objets intelligents. Lors de nos sessions d'implémentation, nous présenterons les concepts de programmation qui vous permettront d'implémenter le comportement de votre objet intelligent. N'oubliez pas que nous implémentons la logique qui détermine la réaction de votre objet aux stimuli externes, en tenant compte de sa structure en termes de capteurs et d'actionneurs. Nous examinerons d'abord des exemples pratiques ensemble, puis vous appliquerez ce que vous avez appris pour implémenter votre propre objet.



Rappelez aux élèves que la phase d'implémentation leur permet de programmer le comportement et la logique de leur objet, en supposant la présence de capteurs et d'actionneurs qui seraient configurés lors d'une phase de prototypage. Les élèves plus jeunes ou moins expérimentés risquent d'oublier la vision globale pendant la programmation ; il est donc essentiel de les rappeler régulièrement à leur idée initiale pour vérifier la cohérence et évaluer les modifications possibles de l'objet proposé.

## Investigation par les élèves

**1. Introduction aux boucles.** Pendant les 10 premières minutes, l'enseignant peut **présenter le concept de boucle à l'aide de diapositives et d'exemples dans MakeCode**, en expliquant ce qu'est une boucle et comment l'utiliser. En utilisant une approche de complexité incrémentale, commencez par **l'absence de boucle (modélisée dans MakeCode par le bloc « au démarrage »)** et poursuivez avec la boucle la plus simple, la boucle « toujours » selon la terminologie MakeCode, qui est un élément essentiel à la conception d'un objet intelligent.

Un exemple est disponible au lien suivant : <https://makecode.microbit.org/C2yeP6E4xJrR>. L'enseignant peut partir d'un code simple (comme l'animation d'un cœur battant) du bloc onstart, montrant qu'il n'est exécuté qu'une seule fois. Il peut ensuite déplacer le code du bloc « au démarrage » vers la boucle « toujours » pour encourager les élèves à remarquer les différences. Dans le spectre allant de « au démarrage » à « toujours », l'enseignant peut ensuite introduire des exemples intermédiaires, comme la boucle « répéter » en montrant que la boucle s'exécute pendant un nombre spécifique d'itérations. L'enseignant peut demander aux élèves de deviner comment répéter la boucle 5 fois, puis 10 fois.

### Programme des élèves



Une fois que l'enseignant s'assure que les élèves ont compris le mécanisme des boucles et comment les implémenter dans MakeCode, il peut arrêter le partage d'écran et leur demander **d'implémenter le comportement initial de leur objet ainsi qu'un comportement de base à répéter cycliquement, selon l'idée développée lors de la séance précédente**. Encouragez-les à distinguer les comportements qui ne doivent se produire qu'au démarrage (initialisation) de ceux qui doivent se produire cycliquement.



Il convient de préciser que les élèves **peuvent modifier leur idée durant le cours**, mais doivent explicitement **noter leurs réflexions sur la modification et garder une trace de la raison de la modification** (dysfonctionnement reconnu, découverte de nouvelles fonctionnalités implémentables, etc.).

**Créez un fichier ou un point de collecte pour les projets.** Demandez à tous les élèves d'y partager leurs programmes (par exemple, un fichier partagé Google où ils colleront le lien de partage vers leur éditeur MakeCode). À chaque étape, vous pourrez utiliser cet enregistrement pour **évaluer les programmes fournis par les élèves** afin de vérifier que la solution proposée contient les blocs requis et demander à chaque élève de présenter brièvement son projet à son état actuel. **Encouragez les élèves à identifier les erreurs logiques et syntaxiques et guidez-les dans leurs corrections.**

**2. Introduction de conditions simples.** Dans la lignée de l'introduction des boucles, l'enseignant peut introduire des conditions et des comparaisons simples à travers des exemples de complexité croissante en partageant l'écran et en présentant des diapositives et des exemples dans MakeCode. À partir de l'exemple disponible sur <https://makecode.microbit.org/Led0ghTFY8xy>, l'enseignant peut :

- Montrer l'exemple d'interaction avec le bouton A
- Modifier l'exemple pour qu'il présente le même comportement si le bouton B est enfoncé
- Afficher un comportement différent si la même condition se produit, par exemple jouer un son au lieu d'imprimer une chaîne sur l'écran

- Ajouter « else » pour obtenir un comportement dépendant de l'entrée. Par exemple, si le bouton A est enfoncé, vous dites « BONJOUR » ; sinon, vous dites « ZZZ ».

À partir de l'exemple disponible sur [https://makecode.microbit.org/\\_ewq3W0EwmHjJ](https://makecode.microbit.org/_ewq3W0EwmHjJ), l'enseignant peut :

- Montrez l'exemple d'un cœur lorsque la température est inférieure à 20
- Modifiez les conditions montrant le comportement différent selon la comparaison en utilisant > ou = au lieu de <
- Modifiez l'entrée, par exemple en remplaçant le capteur de température par le capteur de lumière
- Modifiez le comportement en remplaçant la sortie, par exemple en remplaçant l'écran par un texte ou en changeant le cœur en un soleil
- Ajoutez le reste, par exemple en montrant un soleil si la température est supérieure à 20, sinon un flocon de neige

### Programme des élèves



Une fois la présentation des exemples terminée et les doutes des élèves clarifiés, l'enseignant arrête de partager l'écran et met les élèves au défi d'affiner le comportement de leur objet en **ajoutant au moins une condition et une comparaison.**



Créez une nouvelle section dans le fichier pour collecter les projets. Demandez à tous les élèves de **partager leur programme en enregistrant une nouvelle version de leur projet.** Ouvrez-les pour vérifier que la solution proposée contient les **conditions** et demandez à chaque élève de présenter **brèvement son projet dans son état actuel.** Encouragez les élèves à identifier les erreurs logiques et syntaxiques et guidez-les pour les corriger ou suggérer des extensions du comportement.

**3. Introduction aux conditions complexes.** Comme pour l'introduction des autres constructions, l'enseignant partage l'écran et commence à présenter les conditions complexes à l'aide de diapositives et d'exemples dans MakeCode. À partir de l'exemple disponible sur [https://makecode.microbit.org/\\_LK0UUU2pUXoo](https://makecode.microbit.org/_LK0UUU2pUXoo), l'enseignant :

- Montre l'exemple où un cœur est affiché s'il fait froid (la température est inférieure à 20) ou si le bouton A est enfoncé
- Modifie l'exemple en montrant les différences en remplaçant **OU** par **ET**
- Modifie l'exemple en fonction de différentes entrées, par exemple en remplaçant la lumière par le bouton, par exemple s'il fait chaud et qu'il fait jour, il affiche le soleil
- Ajoute l'autre, par exemple comme alternative au soleil montrant la lune

### Programme des élèves



Une fois la présentation guidée des exemples terminée et les doutes des élèves clarifiés, l'enseignant arrête de partager l'écran et met les élèves au défi **d'affiner le comportement de leur objet en ajoutant au moins une condition complexe.**



Comme pour l'étape précédente, demandez à tous les élèves de **partager leurs programmes.** Ouvrez-les pour vérifier que la solution proposée contient la condition complexe. Comme il s'agit de la dernière étape, demandez aux élèves, un par un, de partager et de présenter collectivement leurs projets, en soulignant s'ils entravent ou facilitent les véhicules autonomes. Chacun est invité à proposer des améliorations ou à identifier des erreurs. Demandez aux élèves d'affiner et de finaliser la proposition d'idées en fonction des conseils reçus. Cette phase se termine par la collecte finale des contributions.

## Restitution et réflexion

Cette phase de mise en œuvre a permis de guider les élèves dans **l'application des concepts de programmation pour donner vie à leurs objets intelligents**. En introduisant et en appliquant progressivement des **boucles, des conditions simples et une logique complexe**, les élèves ont créé des programmes **fonctionnels simulant le comportement de leurs objets**.

Tout au long de ce processus, les élèves ont été encouragés à réfléchir aux **aspects techniques** de la programmation et au **comportement réel de leurs objets intelligents**. Les cycles répétés d'implémentation, suivis de réflexion et d'affinement, ont permis aux élèves de développer non seulement leurs compétences en codage, mais aussi leur esprit critique quant au fonctionnement pratique de leurs conceptions.

La dernière séance de partage a permis aux élèves de découvrir **diverses approches face à des défis similaires et d'apprendre du travail de leurs pairs**. Cet environnement d'apprentissage collaboratif renforce l'idée qu'il existe plusieurs solutions valables à un même problème, tout en soulignant l'importance de la cohérence logique et de la fonctionnalité.

Pour de nombreux élèves, il s'agissait peut-être de leur première expérience de mise en œuvre d'une conception originale en code. L'alliance entre la créativité de la phase d'idéation et la réflexion logique nécessaire à la mise en œuvre permet aux élèves de comprendre comment la programmation peut être utilisée pour **résoudre des problèmes concrets** et améliorer notre environnement grâce aux technologies intelligentes.

Au fur et à mesure qu'ils poursuivent leur parcours d'apprentissage, les élèves peuvent s'appuyer sur ces concepts de programmation fondamentaux pour **créer des conceptions et des implémentations d'objets intelligents de plus en plus sophistiqués**.



## Bibliographie

- [1] Eftychia Roumelioti, Maria Angela Pellegrino, Rosella Gennari, and Mauro D'Angelo. 2022. What Children Learn in Smart-Thing Design at a Distance: An Exploratory Investigation. In Methodologies and Intelligent Systems for Technology Enhanced Learning, 11th International Conference. Springer International Publishing, Cham, 22-31. [https://doi.org/10.1007/978-3-030-86618-1\\_3](https://doi.org/10.1007/978-3-030-86618-1_3).
- [2] Eftychia Roumelioti, Maria Angela Pellegrino, Mehdi Rizvi, Mauro D'Angelo, and Rosella Gennari. 2022. Smart-thing design by children at a distance: How to engage them and make them learn. International Journal of Child-Computer Interaction 33 (2022), 100482.
- [3] Maria Angela Pellegrino, Eftychia Roumelioti, Mauro D'Angelo, and Rosella Gennari. 2021. Engaging Children in Remotely Ideating and Programming Smart Things. In CHIItaly 2021: 14th Biannual Conference of the Italian SIGCHI Chapter (CHIItaly '21). Association for Computing Machinery, New York, NY, USA, Article 20, 1-5. <https://doi.org/10.1145/3464385.3464728>.
- [4] Maria Angela Pellegrino and Mauro D'Angelo. 2021. Engaging Children in Smart Thing Ideation via Storytelling. I-CITIES. <https://icities2021.unisa.it>