

Esprit du module n°2

Le module 2 du rallye prépare à l'épreuve du rallye cycle 2 et du rallye cycle 3.

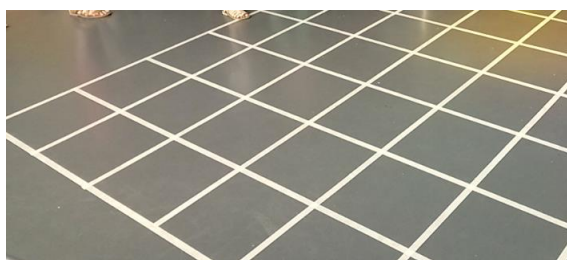
Ce module se décline sous la forme de 2 parcours :

- le parcours 1 est destiné aux élèves et/ou aux enseignants qui découvrent les activités de programmation (séances 1 à 4),
- le parcours 2 est destiné aux élèves et/ou aux enseignants qui ont déjà abordé des activités de programmation en dehors de ce rallye (séances 4 à 6).

			Parcours 1	Parcours 2
Activités débranchées	Activité 1	S'orienter et se déplacer en utilisant des repères	x	
	Activité 2	Coder et décoder pour prévoir, représenter et réaliser des déplacements dans des espaces familiers (Annexes 2 et 3)	x	
	Activité 3	Utiliser un langage relatif (Annexe 4 et/ou 4bis)	x	x
Activités branchées	Activité 4	Rédiger un programme pour permettre à un robot de se déplacer (Annexe 5 ou 5bis)	x	x
	Activité 5	Rédiger un programme pour permettre à un robot de se déplacer en minimisant le nombre de déplacements (Annexes 6 et 1bis)		x
	Activité 6	Rédiger un programme pour permettre à un robot de se déplacer en utilisant les boucles (Annexe 5 ou 5bis)		x

Les activités proposées dans chaque séance peuvent être répétées plusieurs fois.

Dans chaque parcours, vous trouverez une ou plusieurs activités dites débranchées. Ces activités ne nécessitent pas d'ordinateur ou de tablette. Il s'agit de vivre « grande nature » les déplacements d'un robot.



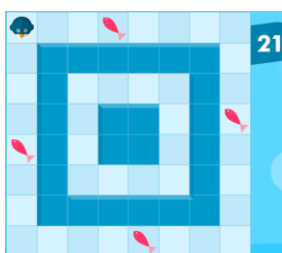
Au cours **des séances 1 et 2**, les élèves vont découvrir **comment construire un programme** et les incontournables, c'est à dire, la création d'un code commun, le choix d'un point d'arrivée et d'un point de départ. Au cours de la séance 3, **les élèves vont pouvoir utiliser le langage relatif**. Nous attirons votre attention sur les trois ordres : avancer, pivoter à droite et pivoter à gauche. Pivoter ne veut pas dire avancer !



C'est à partir de **la séance 4**, que vous trouverez des **activités « dites branchées »**. En effet, ces activités s'appuient sur l'application Tuxbot, téléchargeable gratuitement et qui ne nécessite pas de connexion internet pour son utilisation.

La séance 4 permet aux élèves de découvrir l'application. Il s'agit de coder les déplacements d'un pingouin pour lui permettre de ramasser des poissons et cela en 20 ordres.

C'est ici que s'arrête le parcours n°1.



Le parcours n°2 débute à la dernière séance dite débranchée, c'est à dire la séance 3.

Au cours de la séance 5, le programme permettant au pingouin de ramasser les poissons nécessite plus que 20 ordres. Ce n'est donc pas possible avec une bande de programmation contenant 20 ordres. **Vont devoir surgir d'autres commandes, par exemple, la boucle ou la répétition.**

La séance 6 permet aux élèves de créer des programmes en utilisant ces nouvelles fonctions.



Pour aller plus loin, une séquence utilisant un robot « bluebot » vous est proposée dans l'annexe 7.

Ce module permet une approche de la programmation. Ces compétences sont aussi développées via le projet départemental « 2+4h kids » : il permet de réaliser des activités transdisciplinaires en lien avec l'initiation à la programmation.

<http://www.ia72.ac-nantes.fr/vie-pedagogique/enseigner-avec-le-numerique/se-former-au-numerique/2-4-heures-kids-2017-2018-1053164.kjsp>

Mathématiques	Résolution de problèmes : Le déplacement des robots	Classe/niveau : C2-C3
Référentiel institutionnel :		
Compétences travaillées du socle <ol style="list-style-type: none"> 1. Chercher : Prélever et organiser les informations nécessaires à la résolution de problèmes. S'engager dans une démarche, questionner, émettre des hypothèses. Tester, essayer plusieurs pistes de résolution. (Domaines 2 et 4) 2. Modéliser : Utiliser les mathématiques pour résoudre quelques problèmes issus de situations de la vie quotidienne. (Domaines 1,2 et 4) 3. Représenter : Appréhender différents systèmes de représentation. Utiliser diverses représentations de situations spatiales. (Domaines 1 et 5) 4. Raisonner : Tenir compte d'éléments divers pour modifier son jugement. Prendre progressivement conscience de la nécessité et de l'intérêt. (Domaines 2,3 et 4) 5. Communiquer : Utiliser l'oral et l'écrit, le langage naturel puis quelques représentations et quelques symboles pour expliciter des démarches, argumenter des raisonnements. (Domaines 1 et 3) 		Compétences mathématiques et connaissances associées <p>(Se) repérer et (se) déplacer en utilisant des repères</p> <ul style="list-style-type: none"> ✓ Situer des objets ou des personnes les uns par rapport aux autres ou par rapport à d'autres repères. ✓ S'orienter et se déplacer en utilisant des repères. ✓ Coder et décoder pour prévoir, représenter et réaliser des déplacements dans des espaces familiers, sur un quadrillage, sur un écran.
Objectif pédagogique de l'activité : L'élève doit être capable d'identifier un déplacement, le coder et décoder un autre déplacement.		
Quelques éclairages didactiques et/ou pédagogiques		
Le langage absolu	Le langage relatif	Conseils pour la mise en œuvre
On parle de déplacement absolu, lorsque l'effet des instructions ne dépend pas de l'orientation initiale du « mobile » qui les reçoit. Par exemple : « tourne-toi vers le tableau », « avance de deux pas vers la fenêtre », « avance de trois pas vers l'est ».	On parle de déplacement relatif, lorsque l'effet des instructions dépend de l'orientation initiale du « mobile » qui les reçoit. Par exemple : « effectue un quart de tour vers la droite », « avance de trois pas ». Attention, dans ce langage, lorsque l'on pivote, on n'avance pas en même temps.	- En fonction des réussites et du plaisir des élèves, l'enseignant pourra proposer, tout à fait judicieusement, les activités 2 et 3 durant plusieurs séances en modifiant les variables didactiques (programme plus long, obstacle, passage obligé). - L'ordre des ordres est primordial, une référence historique est la bande programme de la machine universelle de A.Turing
Pour aller plus loin, télécharger le document d'accompagnement des programmes « l'initiation à la programmation cycle 2 et 3 » sur Eduscol.		

	Déroulement de l'activité 1 : S'orienter et se déplacer en utilisant des repères	Durée
Activité préalable	Proposer aux élèves des activités de latéralisation et de déplacement en extérieur (se déplacer tout droit, tourner à droite, pivoter à droite, avancer de trois pas à droite, tourner à gauche, pivoter à gauche, reculer de quatre pas, etc.). Le maître, puis les élèves donnent les consignes de déplacement. Ces activités dans le cadre de l'EPS sont tout à fait préconisées, en amont, pour permettre aux élèves de se repérer dans l'environnement proche.	
Matériel	Pour un groupe : un quadrillage tracé sur le sol de la cour, une feuille ou une ardoise un crayon à papier, une gomme, un crayon effaçable. Pour l'enseignant : préparer un parcours de plots par groupe (photo ci-dessous) ou organiser des ateliers.	
Explicitation	« <i>Au cours de la séance, vous allez apprendre à vous orienter et à vous déplacer en utilisant des repères.</i> »	5'
Recherche (par groupe de 4)	« <i>Voici un espace délimitant le déplacement d'un robot. Dans chaque groupe, l'un de vous est le robot. Cependant, un robot ne sait pas se déplacer seul, il doit obéir à des consignes. D'autres élèves doivent donc piloter le robot : vous allez ainsi rédiger un programme de déplacement. Un des élèves est secrétaire et note ce que le robot doit faire pour se déplacer.</i> »	20'
Analyse (observation de l'enseignant pendant la phase de recherche)	<p>Procédures possibles :</p> <ul style="list-style-type: none"> ➤ Les élèves définissent les conditions (départ et arrivée). ➤ Les élèves utilisent un type de langage uniforme. ➤ Les élèves testent spontanément leur programme et corrigent si besoin. ➤ Les élèves définissent le parcours du robot puis le code en le déplaçant. ➤ Les élèves ajoutent sur le programme une indication permettant de comprendre le sens de la lecture. <p>Exemples d'erreurs possibles :</p> <ul style="list-style-type: none"> ➤ Les élèves ne tiennent pas compte des conditions (départ et arrivée). ➤ Les élèves oublient des déplacements. ➤ Le code utilisé ne présente pas de caractéristique régulière (code différent pour annoncer un même déplacement). ➤ Le sens de lecture du programme sur la bande n'est pas indiqué (le programme peut se lire dans plusieurs sens). 	
Synthèse	<p>Mise en commun des procédures utilisées :</p> <ul style="list-style-type: none"> - Lister les erreurs, pour prendre conscience des éléments incontournables pour permettre à un robot d'effectuer un déplacement. - Formaliser la mise en place d'un langage commun qui permette d'éviter ou d'amoindrir les erreurs des élèves. 	15'
Structuration finale institutionnalisation	<p>« Qu'avez-vous appris au cours de la séance ? »</p> <p>Éléments à faire émerger avec les élèves :</p> <ul style="list-style-type: none"> - se mettre d'accord sur les conditions (départ et arrivée) - utiliser un type de langage : utilisation de mots, flèches, signes, etc - définir l'orientation du support utilisé <p>→ S'orienter et se déplacer en utilisant des repères</p>	15'



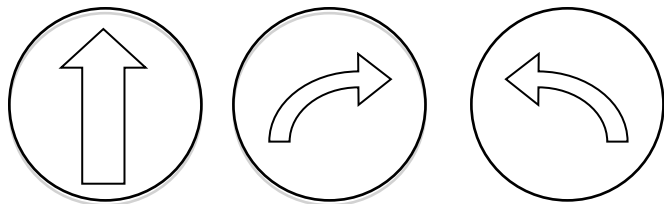
Exemple

	Déroulement de l'activité 2 : Coder et décoder pour prévoir, représenter et réaliser des déplacements dans des espaces familiers	Durée
Retour sur la séance précédente	<u>Rappel des éléments de structuration :</u> - se mettre d'accord sur les conditions (départ et arrivée) - utiliser un type de langage : utilisation de mots, flèches, signes, etc - définir l'orientation du support utilisé	5'
Remarque préalable	L'activité a lieu dehors ou dans une salle. Le quadrillage permet de symboliser le parcours (Annexe 3). Les flèches proposées ci-dessous seront utiles pour la mise en commun.	
Matériel	Pour chaque groupe d'élève : une bande de papier pour marquer les déplacements (Annexe 2), le quadrillage (Annexe 3), un crayon à papier et une gomme. Pour l'enseignant : préparer un quadrillage matérialisé dans la cour (photo ci-dessus), les flèches ci-dessous pour la mise en commun.	
Explicitation	« <i>Au cours de la séance, vous allez apprendre à coder et décoder des déplacements.</i> »	5'
Recherche 1 (par groupe de 4)	« <i>Voici un espace délimitant le déplacement d'un robot. Vous fixez ensemble le point de départ puis le point d'arrivée (à noter sur le quadrillage).</i> <i>Dans chaque groupe :</i> - un élève est le robot, - un élève pilote le robot, - un élève note sur la bande ce que le pilote indique, - un élève trace le déplacement sur le quadrillage. »	20'
Analyse 1 (observation de l'enseignant pendant la phase de recherche)	Procédures possibles : ➤ Les élèves définissent les conditions. ➤ Les élèves utilisent un type de langage univoque. ➤ L'ordre des ordres est clair. Exemples d'erreurs possibles : ➤ Le code utilisé ne présente pas de caractéristique régulière (code différent pour annoncer un même déplacement). ➤ Le programme peut se lire dans plusieurs sens.	
Remarque	L'enseignant garde les parcours tracés sur les quadrillages (pour la fin de la séance 2). Les groupes gardent leur bande. La situation 2 illustre la mise en commun (synthèse) et permet de comprendre les indications incontournables à l'élaboration d'un message.	
Recherche 2 (par groupe de 4)	« <i>Vous allez échanger vos bandes. Sur la nouvelle bande, est indiqué un programme. Vous devez exécuter ce programme avec votre robot. Dans chaque groupe :</i> - un élève est le robot, - un élève pilote le robot, - un élève trace le déplacement sur le quadrillage, - un élève vérifie. »	20'

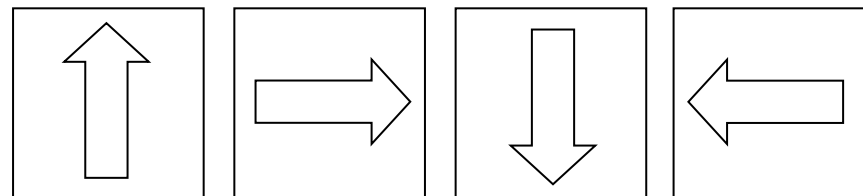
Analyse (observation de l'enseignant pendant la phase de recherche)	Procédures possibles : ➤ Les élèves réussissent à déchiffrer le programme. Exemples d'erreurs possibles : ➤ Les conditions (départ et arrivée) ne sont pas indiquées. ➤ Le code utilisé est illisible ou incompréhensible. ➤ L'ordre des déplacements n'est pas indiqué.	
Synthèse	Mise en commun des procédures utilisées : - Comparer si les productions sur papier pointé des 2 groupes (recherche 1 et recherche 2) concordent. - Lister les erreurs, pour prendre conscience des éléments incontournables pour permettre à un robot d'effectuer un déplacement. - Formaliser la mise en place d'un langage commun avec des flèches (absolu et/ou relatif).	15'
Structuration finale institutionnalisation	« Qu'avez-vous appris au cours de la séance ? » Éléments à faire émerger avec les élèves : - valider les programmes qui fonctionnent - se mettre d'accord et élaborer un code commun en s'appuyant sur les programmes qui fonctionnent (absolu et/ou relatif) - prendre conscience de l'importance de l'ordre des ordres dans la séquence de programmation (une bande à cases ou une bande numérotée) → Coder et décoder pour prévoir, représenter et réaliser des déplacements dans des espaces familiers.	15'

Exemple de flèches code (pour l'enseignant pour la mise en commun) :

Langage relatif



Langage absolu

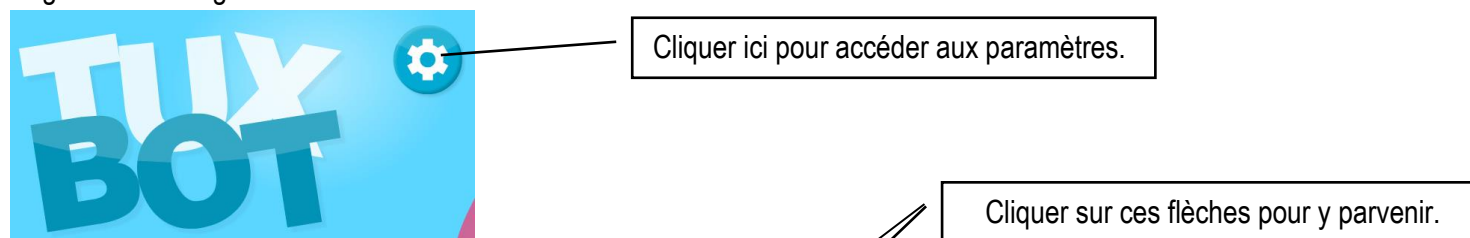


	Déroulement de l'activité 3 : Utiliser un langage relatif	Durée
Retour sur la séance précédente	<p>Rappel des éléments de structuration :</p> <ul style="list-style-type: none"> - valider les programmes qui fonctionnent - se mettre d'accord et élaborer un code commun en s'appuyant sur les programmes qui fonctionnent (absolu et/ou relatif) - prendre conscience de l'importance de l'ordre des ordres dans la séquence de programmation (une bande à cases ou une bande numérotée) <p>Pour la suite des activités, le langage relatif sera utilisé.</p> <p><u>Attention, dans ce langage, lorsque l'on pivote, on n'avance pas en même temps.</u></p>	5'
Remarque	Cette séance a lieu dans la classe. Les élèves utilisent les quadrillages ainsi que les bandes (Annexe 4 ou annexe 4bis avec la case arrivée libre). Le robot est représenté par une coccinelle pour permettre aux élèves de prendre compte de l'orientation, la coccinelle avance en avant !	
Matériel	<p>Pour chaque élève : annexe 4 avec la case cible posée (ou annexe 4bis avec la case arrivée libre).</p> <p>Pour l'enseignant : annexe 4 (ou annexe 4bis) agrandie, vidéo-projetée (ou fichier sur TBI)</p>	
Explicitation	« Vous allez apprendre à coder un déplacement en tenant compte des cases « départ » et « arrivée » en langage relatif. »	5'
Recherche (individuelle puis 2)	« Nous avons perçu l'importance d'un code commun et les avantages du langage relatif. Maintenant vous allez programmer le déplacement de cette coccinelle. Elle est orientée dans la case départ comme indiqué. Elle doit arriver dans la case arrivée, celle où il y a la fleur. Chaque élève écrit son programme et son camarade le teste. »	20'
Analyse (observation de l'enseignant pendant la phase de recherche)	<p>Procédures possibles :</p> <ul style="list-style-type: none"> ➤ Les élèves s'approprient le nouveau code. ➤ Les élèves tracent le chemin sur le quadrillage. ➤ Les élèves codent le déplacement case par case. ➤ Les élèves codent tout le déplacement puis vérifient l'ensemble. ➤ Les élèves vérifient le programme avant de le donner à leur camarade. <p>Exemples d'erreurs possibles :</p> <ul style="list-style-type: none"> ➤ Les conditions ne sont pas respectées. ➤ Le code « pivoter » n'est pas bien utilisé (rappel : pivoter ne veut pas dire avancer). ➤ Les élèves ne vérifient pas le programme avant de le donner à leur camarade. ➤ Les élèves ne tiennent pas compte du sens de lecture du programme. 	
Synthèse	<p>Mise en commun des procédures utilisées : Au tableau, ou sur le TBI, les différents programmes sont affichés.</p> <p>Les élèves viennent éprouver le programme.</p>	15'
Structuration finale institutionnalisation	<p>« Qu'avez-vous appris au cours de la séance ? »</p> <p>Éléments à faire émerger avec les élèves :</p> <ul style="list-style-type: none"> - les flèches relatives sont plus adaptées pour le langage de programmation - toute action attendue du robot doit figurer dans le programme <p>→ Utiliser le langage relatif (NB : quand le robot pivote, il n'avance pas).</p>	15'

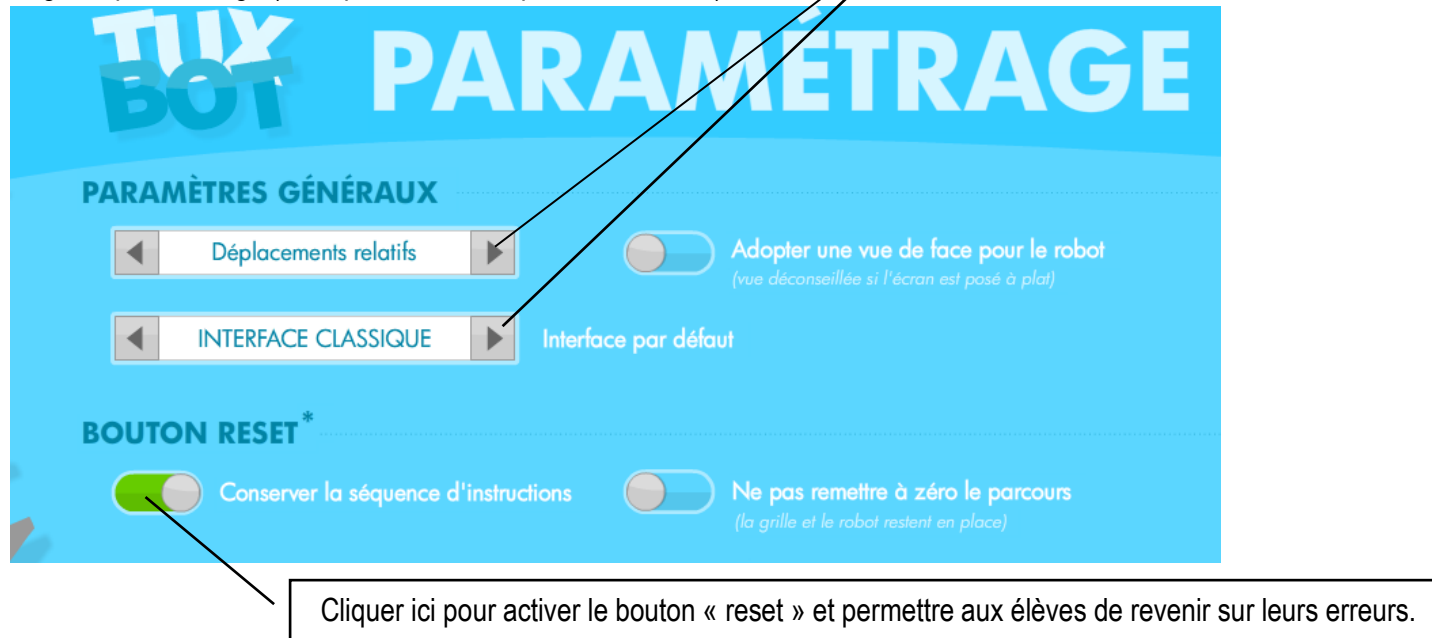
	Déroulement de l'activité 4 : Rédiger un programme pour permettre à un robot de se déplacer	Durée
Retour sur la séance précédente	<p>Rappel des éléments de structuration :</p> <ul style="list-style-type: none"> - les flèches relatives sont plus adaptées pour le langage de programmation - toute action attendue du robot doit figurer dans le programme <p>Pour la suite des activités, le langage relatif sera utilisé.</p> <p><u>Attention, dans ce langage, lorsque l'on pivote, on n'avance pas en même temps.</u></p>	5'
Remarque	<p>Tuxbot est une application permettant de programmer les déplacements d'un robot virtuel. Cette application est téléchargeable sur le site académique de Nantes.</p> <p>Cette application peut être paramétrée de différentes manières, vous trouverez ci-dessous le paramétrage adapté à la séance 4.</p>	
Matériel	<p>Pour chaque groupe d'élèves :</p> <ul style="list-style-type: none"> - un ordinateur ou tablette avec l'application (sur Windows et Android) est téléchargeable ici et ne nécessite pas de connexion internet pour la mise en œuvre (http://appli-etna.ac-nantes.fr:8080/ia53/tice/ressources/tuxbot/index.php). 20 défis de difficultés graduelles sont disponibles - le carnet de programmation (1 exemplaire pour 2 élèves) (Annexe 5 ou 5bis en version compact) <p>Pour l'enseignant :</p> <ul style="list-style-type: none"> - un ordinateur avec l'application - le carnet de programmation format A3 (Annexe 5 ou 5bis en version compact) 	
Explicitation	« Dans cette séance, vous allez apprendre à rédiger un programme pour permettre à un robot de se déplacer. »	5'
Recherche (par 2)	<p>« Vous devez programmer le parcours du pingouin de telle manière qu'il ramasse tous les poissons présents sur le plateau. Pour cela, le pingouin doit s'arrêter ou passer par la case où il y a un poisson.</p> <p>Le programme ne peut comporter qu'un maximum de 20 instructions. Le pingouin ne doit pas tomber dans l'eau ni sortir de la grille.</p> <p>A deux, vous allez réfléchir au programme, le tester en direct. S'il fonctionne alors vous devez reporter les instructions sur votre carnet de programmation et passer au défi suivant. »</p>	20'
Analyse (observation de l'enseignant pendant la phase de recherche)	<p>Procédures possibles :</p> <ul style="list-style-type: none"> ➤ Les élèves s'approprient le nouveau dispositif (l'application et/ou le carnet de programmation). ➤ Les élèves testent le programme au fur et à mesure. ➤ Les élèves codent le déplacement case par case ou les élèves codent tout le déplacement puis vérifient l'ensemble. ➤ Les élèves vérifient le programme et corrigent si besoin. ➤ Les élèves recopient sur papier ou sur le carnet de programmation le programme validé. <p>Exemples d'erreurs possibles :</p> <ul style="list-style-type: none"> ➤ Le code « pivoter » n'est pas bien utilisé (rappel : pivoter ne veut pas dire avancer). ➤ Les élèves ne lancent pas le programme. ➤ Les élèves ne tiennent pas compte des obstacles ou sortent du quadrillage. ➤ Les élèves ne ramassent pas le poisson (ou les poissons à partir du défi n°5). ➤ Les élèves ne corrigent pas leur programme lorsqu'il apparaît comme erroné. 	

Synthèse	Mise en commun des procédures utilisées : Au tableau, ou sur le TBI (ou vidéoprojecteur), les élèves viennent proposer leur séquence de programmation dans le but d'exposer la diversité des programmes possibles.	15'
Structuration finale institutionnalisation	« Qu'avez-vous appris au cours de la séance ? » Éléments à faire émerger avec les élèves : <ul style="list-style-type: none"> - les instructions en langage relatif notamment tourner à droite/gauche qui correspond à pivoter d'un quart de tour vers la droite/gauche (sans avancer) - la diversité des programmes possibles pour déplacer les robots - les avantages de l'outil numérique (essai/erreur, auto-évaluation) → Utiliser un outil numérique pour rédiger un programme qui permet à un robot de se déplacer.	15'


Page de démarrage :



Page de paramétrage (telle qu'elle doit être pour la séance 4) :



	Déroulement de l'activité 5 : Rédiger un programme pour permettre à un robot en minimisant le nombre de déplacements	Durée
Retour sur la séance précédente	<u>Rappel des éléments de structuration :</u> - les instructions en langage relatif notamment tourner à droite/gauche qui correspond à pivoter d'un quart de tour vers la droite/gauche (sans avancer) - la diversité des programmes possibles pour déplacer les robots - les avantages de l'outil numérique (essai/erreur, auto-évaluation)	5'
Remarque	La grille (« Défi expert n°2 ») proposée dans cette activité ne fait partie du programme « Tuxbot ». Vous trouverez ci-dessous la procédure pour l'insérer dans ce programme : - le paramétrage adapté de la séance 5 - l'importation de la nouvelle grille dans le programme « Tuxbot » de l'ordinateur de l'enseignant (en vidéo-projection) NB : Dans cette grille, le nombre d'instructions possibles ne permet pas de ramasser tous les poissons (nécessité de minimiser le nombre de déplacements en incluant des répétitions/boucles).	
Matériel	- Pour les élèves : - la fiche « Défi expert n°2 » (1 par élève) pour la recherche 1 (Annexe 6) - la fiche « Défi expert n°2 » (1 par 2 élèves) pour la recherche 2 (Annexe 6) - Les flèches-codes à découper (pour la recherche 2) (Annexe 1bis). - Pour l'enseignant : - l'application (sur Windows et Android) est téléchargeable ici et ne nécessite pas de connexion internet pour la mise en œuvre (http://appli-etna.ac-nantes.fr:8080/ia53/tice/ressources/tuxbot/index.php). 20 défis de difficultés graduelles sont disponibles. - l'importation du « Défi expert n°2 »	
Explicitation	« Dans cette séance, vous allez apprendre à rédiger un programme pour permettre à un robot de se déplacer en minimisant le nombre de déplacements. »	5'
Recherche 1 (individuel et collectif)	« Nous devons programmer le parcours du pingouin de telle manière qu'il ramasse tous les poissons présents sur le plateau. Le programme ne peut comporter qu'un maximum de 20 instructions. Le pingouin ne doit pas tomber dans l'eau ni sortir de la grille. Individuellement, vous allez réfléchir au programme du défi expert n°2 et écrire le programme sur la fiche « Défi expert n°2 ». »	20'
Analyse 1 (observation de l'enseignant pendant la phase de recherche)	Procédures possibles : ➤ Les élèves testent le programme au fur et à mesure. ➤ Les élèves codent le déplacement case par case. ➤ Les élèves codent tout le déplacement puis vérifient l'ensemble. ➤ Les élèves vérifient le programme et corrigent si besoin. Exemples d'erreurs possibles : ➤ Le code « pivoter » n'est pas bien utilisé (rappel : pivoter ne veut pas dire avancer). ➤ Les élèves ne tiennent pas compte des obstacles.	

	<ul style="list-style-type: none"> ➤ Les élèves sortent du quadrillage. ➤ Les élèves ne corrigent pas leur programme lorsqu'il apparaît comme erroné. ➤ Les élèves ne ramassent pas tous les poissons. ➤ Les élèves ajoutent des cases à la grille d'instructions. 	
Synthèse	<p>Mise en commun des procédures utilisées :</p> <p>Au tableau, ou sur le TBI (ou vidéoprojecteur), les élèves viennent proposer leur séquence de programmation et exposent les difficultés rencontrées.</p> <p>Les échanges portent sur la nécessité d'optimiser le nombre d'instructions dans l'optique de préparer la recherche suivante (introduction de la notion de répétitions / boucles).</p>	15'
Structuration intermédiaire	<p>« Qu'avez-vous appris au cours de la recherche ? »</p> <p>Éléments à faire émerger avec les élèves :</p> <ul style="list-style-type: none"> - optimiser les déplacements en créant des répétitions (boucles), - représenter les instructions répétitions/boucles à l'aide des symboles suivants (). <p>→ Utiliser un outil numérique pour rédiger un programme qui permet à un robot de se déplacer en utilisant des boucles.</p>	15'
Recherche 2 (par 2)	<p><i>« Vous devez programmer le parcours du pingouin de telle manière qu'il ramasse tous les poissons présents sur le plateau. Le programme ne peut comporter qu'un maximum de 20 instructions. Le pingouin ne doit pas tomber dans l'eau ni sortir de la grille. A deux, vous allez réfléchir au programme en utilisant les instructions répétitions/boucles (fiche à découper) du défi expert n°2 et écrire le programme sur la fiche « Défi expert n°2 ». »</i></p>	20'
Analyse 2 (observation de l'enseignant pendant la phase de recherche)	<p>Procédures possibles :</p> <ul style="list-style-type: none"> ➤ Les élèves s'appuient sur le programme précédent, le corrigent et le complètent. ➤ Les élèves codent le déplacement case par case. ➤ Les élèves codent tout le déplacement sans tenir compte des contraintes (nombre d'instructions maximum) puis remplacent avec des instructions répétitions/boucles. ➤ Les élèves vérifient le programme et corrigent si besoin. <p>Exemples d'erreurs possibles ou difficultés rencontrées :</p> <ul style="list-style-type: none"> ➤ Les élèves ne comprennent pas les procédures de répétitions (x1 : répéter une fois l'action qui suit, x2 : répéter 2 fois l'action qui suit, etc.). ➤ Les élèves ne tiennent pas compte des obstacles ou les élèves sortent du quadrillage. ➤ Les élèves ne corrigent pas leur programme lorsqu'il apparaît comme erroné. ➤ Les élèves ne ramassent pas tous les poissons. ➤ Les élèves ajoutent des cases à la grille d'instructions. 	
Synthèse	<p>Mise en commun des procédures utilisées :</p> <p>Au tableau, ou sur le TBI, les élèves viennent proposer leur séquence de programmation dans le but d'exposer la diversité des programmes possibles.</p> <p>Les échanges peuvent porter sur l'utilisation (l'action se place après le signe « répétition ») et l'optimisation des procédures de répétitions.</p>	15'

Structuration finale institutionnalisation	<p>« Qu'avez-vous appris au cours de la séance ? »</p> <p>Éléments à faire émerger avec les élèves :</p> <ul style="list-style-type: none"> - optimiser les déplacements en utilisation des répétitions (boucles) - utiliser les procédures de répétitions : l'action se place après le signe « répétition » <p>→ Utiliser un outil numérique pour rédiger un programme qui permet à un robot de se déplacer en utilisant des boucles.</p>	15'
--	--	-----

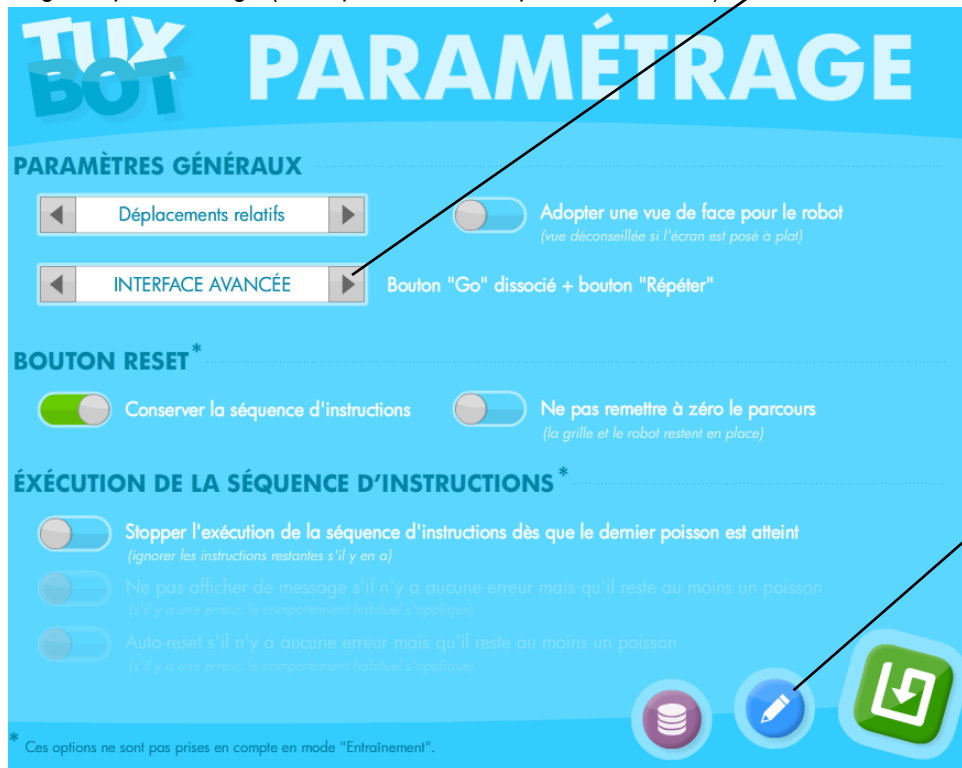
Page de démarrage :



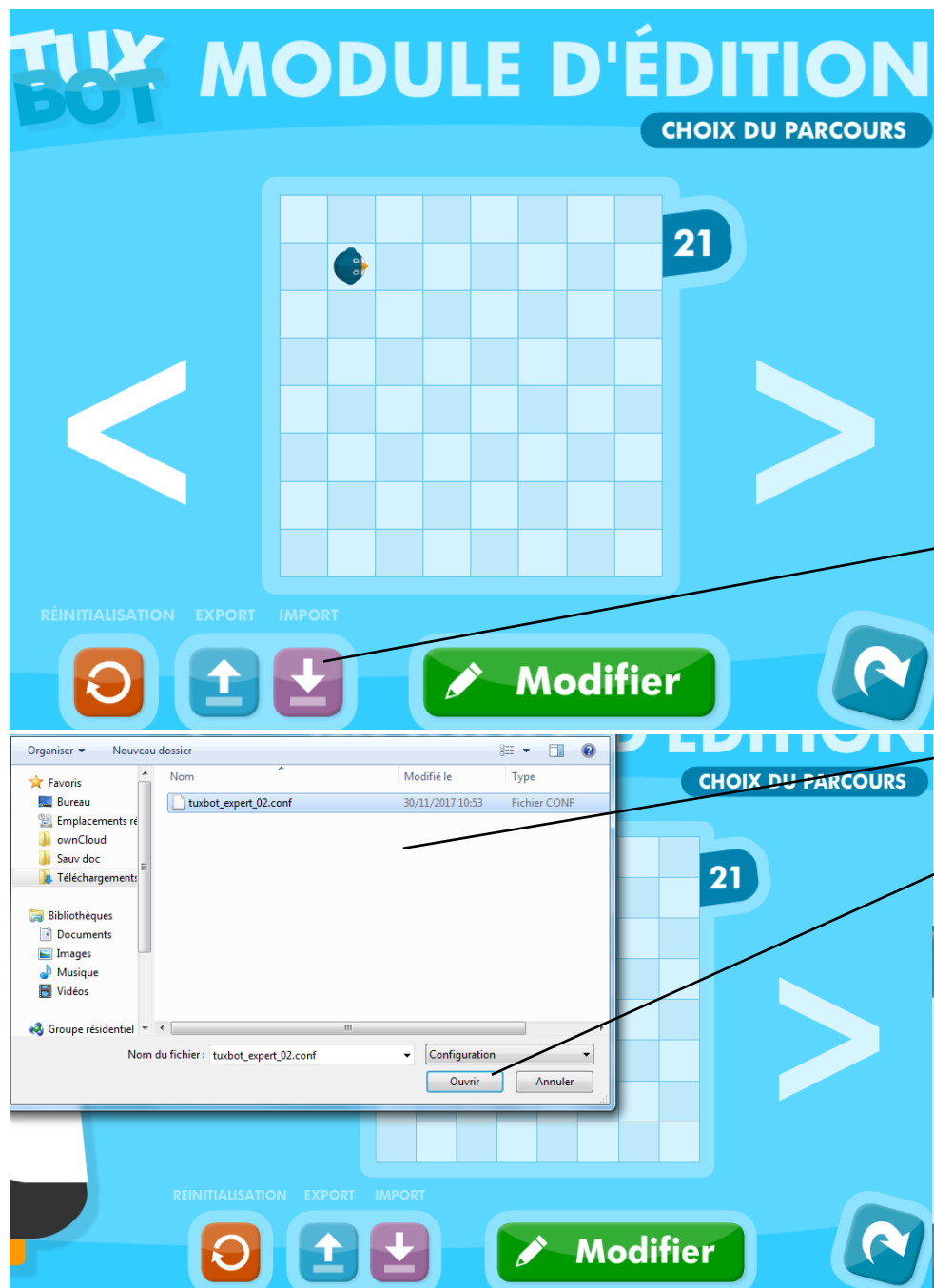
Cliquer ici pour accéder aux paramètres.

Cliquer sur cette flèche pour y parvenir

Page de paramétrage (telle qu'elle doit être pour la séance 5) :



Cliquer sur ici pour accéder au module d'édition.



Cliquer ici pour importer le défi expert n°2.

Ouvrir le dossier où est enregistré « tuxbot_expert_02.conf ».

Cliquer sur « ouvrir ».

Valider.

	Déroulement de l'activité 6 : Rédiger un programme pour permettre à un robot de se déplacer en utilisant les boucles	Durée
Retour sur la séance précédente	<u>Rappel des éléments de structuration :</u> - optimiser les déplacements en utilisant des répétitions (boucles) - utiliser les procédures de répétitions : l'action se place après le signe « répétition »	5'
Remarque	Les éléments de paramétrage correspondent à ceux de la séance 5 (sans importation).	
Matériel	- Pour les élèves : - l'application (sur Windows et Android) est téléchargeable ici et ne nécessite pas de connexion internet pour la mise en œuvre (http://appli-etna.ac-nantes.fr:8080/ia53/tice/ressources/tuxbot/index.php). 20 défis de difficultés graduelles sont disponibles. - le carnet de programmation (1 exemplaire pour 2 élèves) (Annexe 5 ou 5bis en version compact). - Pour l'enseignant : - l'application avec les paramétrages de l'activité 5 (sans importation) - le carnet de programmation format A3 (Annexe 5 ou 5bis en version compact)	
Explicitation	« Dans cette séance, vous allez apprendre à rédiger un programme pour permettre à un robot de se déplacer en utilisant les répétitions/boucles. »	5'
Recherche (par 2)	« Vous devez programmer le parcours du pingouin de telle manière qu'il ramasse tous les poissons présents sur le plateau. Le programme ne peut comporter qu'un maximum de 20 instructions. Le pingouin ne doit pas tomber dans l'eau ni sortir de la grille. A deux, vous allez réfléchir au programme en utilisant les répétitions/boucles (touche R sur le logiciel), le tester en direct. S'il fonctionne alors vous devez reporter les instructions sur votre carnet de programmation et passez au défi suivant. Attention, vous devez utiliser le moins d'instructions possibles. »	20'
Analyse (observation de l'enseignant pendant la phase de recherche)	Procédures possibles (voir procédures activités 5). Exemples d'erreurs possibles ou difficultés rencontrées : <ul style="list-style-type: none"> ➤ Les élèves ne comprennent pas les procédures de répétitions. ➤ Le code « pivoter » n'est pas bien utilisé (rappel : pivoter ne veut pas dire avancer). ➤ Les élèves ne tiennent pas compte des obstacles ou les élèves sortent du quadrillage. ➤ Les élèves ne corrigent pas leur programme lorsqu'il apparaît comme erroné. ➤ Les élèves ne ramassent pas tous les poissons. 	
Synthèse	Mise en commun des procédures utilisées : Au tableau, ou sur le TBI (ou vidéoprojecteur), les élèves viennent proposer leur séquence de programmation dans le but d'exposer la diversité des programmes possibles. Les échanges portent sur la façon de réaliser le moins de déplacements possibles.	15'
Structuration finale institutionnalisation	« Qu'avez-vous appris au cours de la recherche ? » Éléments à faire émerger avec les élèves : utiliser les répétitions (boucles), minimiser les déplacements. → Utiliser un outil numérique pour rédiger un programme qui permet à un robot de se déplacer en utilisant des boucles.	15'

Pour aller plus loin, une séquence utilisant un robot « bluebot » vous est proposée dans l'annexe 7.