



Manuel de l'Enseignant (GAR)

Table des matières

A propos	1
Pédagogie générale	1
Contexte d'utilisation	2
Les leçons	2
Les types de niveaux	2
Les fiches pédagogiques	3
Mise en place d'un atelier	4
L'évaluation automatique	4
Nous contacter	6
Annexe	7
Guide rapide	7
Fiches pédagogiques.....	8

A propos

Coding Park est une plateforme ludique qui aide les enfants à comprendre la pensée informatique par le jeu vidéo. Conçue à la fois pour les élèves et les enseignants, la plateforme permet aux élèves d'acquérir les compétences requises progressivement dans le jeu. Elle propose aux enseignants des ressources interactives prêtes à l'emploi, ainsi qu'un outil de supervision d'ateliers permettant d'évaluer la progression des élèves.

Pédagogie générale

Dans l'univers de Coding Park, Golden Quest est un parcours de chasse aux trésors dans un univers de robots et de pirates. Les enfants créent des algorithmes pour résoudre des problèmes logiques avec pour objectif de guider Cody, le petit robot pirate, jusqu'au trésor caché dans chaque exercice.

Cycle 2 : s'orienter et se déplacer en utilisant une grille (repère) ; réaliser des déplacements dans l'espace et les coder pour qu'un autre élève puisse les reproduire ; produire des représentations d'un espace restreint et s'en servir pour communiquer des positions ; programmer les déplacements d'un robot ou ceux d'un personnage sur un écran.

Cycle 3 : se repérer, décrire ou exécuter des déplacements sur une carte ; accomplir, décrire, coder des déplacements dans des espaces familiers ; programmer les déplacements d'un robot sur un écran en utilisant un logiciel de programmation.



Contexte d'utilisation

La plateforme convient pour un usage lors d'ateliers récurrents en classe et/ou à la maison en autonomie.

- Lors d'ateliers et stages réalisés au sein d'écoles, centres aérés et autres organismes de formation, l'enseignant dispose d'un outil lui permettant de proposer du contenu pédagogique et d'un moyen pour mesurer et suivre la progression des élèves.
- En autonomie, la plateforme propose une aide contextuelle dans l'univers du jeu, avec un contenu s'adapte en fonction de la progression de l'élève, ainsi qu'un lexique complet des actions qu'il est possible de faire.

Les leçons

Le parcours d'apprentissage Golden Quest est un jeu de chasse aux trésors. Il présente une progression sur 16 leçons ou chapitres, qui adressent les bases de la programmation à travers les aventures de Cody et de son équipage. Les élèves ont le choix entre Blockly, Pseudocode, ou Python.



Les types de niveaux

Le parcours présente trois types de niveaux :

- Les **niveaux verts** : niveaux dans lesquels les élèves devront coder les actions de Cody,
- Les **niveaux bleus** : niveaux inversés, le code est déjà donné à l'élève, celui-ci doit dessiner le chemin minimal,
- Les **niveaux rouges** : niveaux contre-la-montre, les élèves doivent trouver une solution avec la contrainte temps.



Les fiches pédagogiques

Pour vous accompagner lors de la création et de la mise en place de vos ateliers, nous vous avons préparé une fiche pédagogique pour chaque leçon. Sur ces fiches, vous pourrez retrouver les pré-requis, les objectifs ainsi que le plan de chaque leçon.

Le plan de votre leçon vous donnera des astuces et solutions pour accompagner vos élèves et pour venir à bout de chaque niveau. Par ailleurs, sur ces mêmes fiches, nous compilons également les erreurs les plus fréquentes que nous avons constatées lors de nos ateliers.

La liste complète des fiches pédagogiques est donnée en annexe.



Objectifs

- ✓ Prendre en main l'éditeur de code
- ✓ Se familiariser avec la syntaxe du langage
- ✓ Comprendre la logique des déplacements

Pré requis

- ✓ Au minimum un ordinateur pour deux élèves et une connexion internet
- ✓ La création en amont d'une session de codage par l'enseignant
- ✓ Tous les élèves doivent être connectés avec leur identifiant de session

Plan de la leçon

1. Click Play : Cliquer sur Play et observer ce que fait Cody
2. Golden Tooth : Ecrire `down()` à la ligne 7
3. The Jumpy Frog : Ecrire `jump(3)` à la ligne 10
4. The Bushes: Introduire la notion de bug et corriger la ligne 8
5. The Skeleton : 2 bugs à la ligne 7 et 9, mettre les bons paramètres
6. The Snake : Introduire brièvement la notion de boucle, remplacer 3 par 8
7. The Dreadful Five : Rajouter l'instruction `right()` à la ligne 5
8. The Teleporters: Corriger le bug à la ligne 4, remplacer 6 par 5
9. The Teleporter Direction : Expliquer le sens d'entrée/sortie d'un téléporteur
10. End of Part I : Compléter le code à partir de la ligne 7

Mise en place d'un atelier

Aucune configuration supplémentaire n'est nécessaire pour organiser un atelier. Une classe virtuelle vous est automatiquement attribuée et activée par défaut. Vos élèves y accèdent directement dès leur connexion à l'ENT, sans qu'aucune action supplémentaire ne soit requise de votre part. Les élèves sont progressivement associés aux groupes de manière fluide. Ils ont un accès complet à l'ensemble du catalogue de ressources, sans autre intervention de votre part.

L'Évaluation automatique

Vous l'avez remarqué, les solutions des élèves sont évaluées par rapport à une solution optimale connue a priori. Il existe plusieurs paramètres entrant en jeu dans le calcul du score, dont les plus importants sont :

- **Le nombre d'essais/erreurs** : la note est pondérée par le nombre de fois que l'élève teste son algorithme avant d'aboutir à une solution. Le premier essai n'est pas comptabilisé.
- **La longueur de la solution** : il s'agit ici d'évaluer la solution en termes de complexité algorithmique et de performances d'exécution. Par exemple, écrire `jump(3)` est plus pertinent qu'écrire 3 fois `jump(1)`.

Une note sur 10 est affectée à la solution de l'élève, qui est ensuite transformée en étoiles (5 étoiles au maximum, chaque étoile représente 2 points). Enfin, l'élève obtient toujours une médaille selon les règles suivantes :

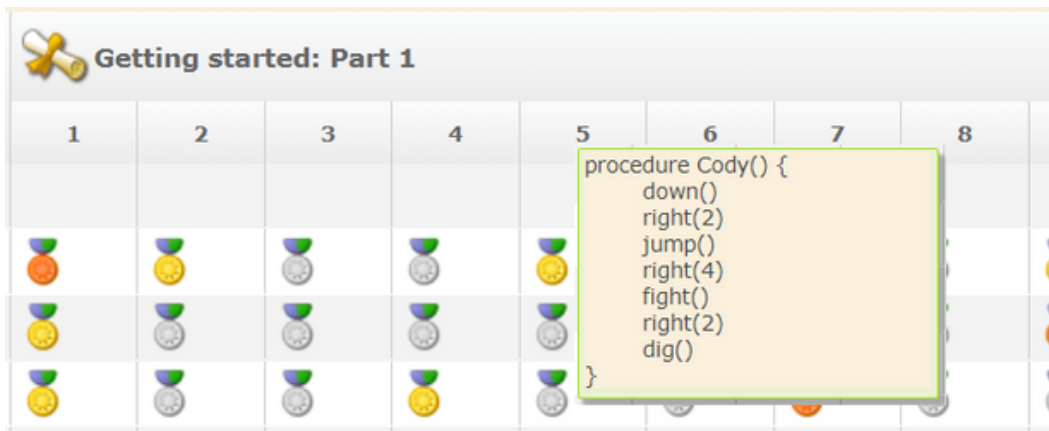
- Médaille d'or : le score est de 10 sur 10
- Médaille d'argent : le score vaut entre 7 et 9 sur 10
- Médaille de bronze : le score est inférieur à 6 sur 10



Sur le **Tableau de bord**, seules les médailles apparaissent dans la grille. En positionnant le curseur de la souris au-dessus d'une médaille, le score de l'élève (note sur 10) est affiché.



Enfin, en positionnant le curseur de la souris sur le numéro d'un exercice, la solution est affichée.




Nous contacter









Nous sommes à votre écoute, n'hésitez pas à nous envoyer un mail à l'adresse support@codingpark.io pour toutes questions relatives au fonctionnement de la plateforme. Aussi, nous sommes ouverts à toutes suggestions qui puissent améliorer l'expérience utilisateur, l'évaluation automatique, ou tout autre domaine connexe pouvant rendre la plateforme meilleure d'un point de vue pédagogique.

Annexe

Guide Rapide



Guide Rapide

<p>up() haut</p>  <p>left() gauche</p>   <p>right() droite</p>  <p>down() bas</p>	<p>jump() sauter</p>  <p>fight() se battre</p>  <p>dig() creuser</p>  <p>pushButton() appuyer sur un bouton</p> 	<p>repeat(...) répéter</p> <p>while(...) tant que</p> <p>if(...) Si</p> <p>var variable</p>
---	---	---

© Copyright - All rights reserved Coding Park, visit www.codingpark.io

Ne pas jeter sur la voie publique

Fiches pédagogiques



Getting Started - Part I

Objectifs

- ✓ Prendre en main l'éditeur de code
- ✓ Se familiariser avec la syntaxe du langage
- ✓ Comprendre la logique des déplacements

Pré requis

- ✓ Au minimum un ordinateur pour deux élèves et une connexion internet
- ✓ La création en amont d'une session de codage par l'enseignant
- ✓ Tous les élèves doivent être connectés avec leur identifiant de session

Plan de la leçon

1. Click Play : Cliquer sur Play et observer ce que fait Cody
2. Golden Tooth : Ecrire `down()` à la ligne 7
3. The Jumpy Frog : Ecrire `jump(3)` à la ligne 10
4. The Bushes: Introduire la notion de bug et corriger la ligne 8
5. The Skeleton : 2 bugs à la ligne 7 et 9, mettre les bons paramètres
6. The Snake : Introduire brièvement la notion de boucle, remplacer 3 par 8
7. The Dreadful Five : Rajouter l'instruction `right()` à la ligne 5
8. The Teleporters: Corriger le bug à la ligne 4, remplacer 6 par 5
9. The Teleporter Direction : Expliquer le sens d'entrée/sortie d'un téléporteur
10. End of Part I : Compléter le code à partir de la ligne 7

Nos conseils

- Commencer par expliquer ce qu'est la programmation en termes simples
- Présenter l'objectif du jeu : mener Cody le robot pirate vers un trésor
- Demander aux élèves de toujours tester le début de solution qui leur est donnée, d'observer ensuite ce qui ne va pas, et enfin de corriger/compléter
- Insister sur la prononciation et l'orthographe des actions de base
- Laisser chacun des élèves progresser à son propre rythme

Astuces

- Pour sauter dans un sens donné, Cody doit prendre de l'élan
- La direction de sortie d'un téléporteur est la même que la direction d'entrée

Les erreurs les plus fréquentes

- Oubli des parenthèses derrière une action ou une fonction
- Suppression involontaire d'un crochet "{ " ou "}"
- Mauvaise orthographe lors de l'écriture d'une action (ex : righth, duwn...)



Getting Started - Part 2

Objectifs

- ✓ Prendre en main l'éditeur de code
- ✓ Se familiariser avec la syntaxe du langage
- ✓ Comprendre la logique des déplacements

Pré requis

- ✓ Au minimum un ordinateur pour deux élèves et une connexion internet
- ✓ La création en amont d'une session de codage par l'enseignant
- ✓ Tous les élèves doivent être connectés avec leur identifiant de session

Plan de la leçon

1. The Hat : Combattre un pirate à la fois en commençant par celui du dessus
2. The Bandits : Esquiver le groupe de pirate en empruntant le téléporteur
3. Take a little Run : Prendre de l'élan avant le saut en allant vers la gauche
4. Jumping : Faire le tour de l'île pour prendre l'élan dans la bonne direction
5. The Button : Utiliser `pushButton()` pour activer un téléporteur
6. Several Buttons : Désactiver les téléporteurs, sauter, puis les activer à nouveau
7. Desactivate and Activate : D'abord combattre le pirate afin de créer le chemin
8. Calling Procedures : Appeler la procédure `L()` à la ligne 8
9. Calling with Parameter : Écrire `fightLine()` à ligne 7 pour battre la rangée
10. End of Part 2 : Activer tous les téléporteurs les uns à la suite des autres

Nos conseils

- Rappeler aux enfants les notions acquises à la leçon précédente
- Rappeler comment insérer une ligne dans l'éditeur avec la touche « Entrée »
- Rappeler le vocabulaire de déplacement vu la semaine dernière
- Laisser chacun des élèves progresser à son propre rythme

Astuces

- Pour sauter dans une direction, Cody doit prendre de l'élan
- Pour activer/désactiver un téléporteur, appuyer sur le bouton de la même couleur

Les erreurs les plus fréquentes

- Oublie des parenthèses derrière un mot ou une fonction
- Suppression involontaire d'un crochet "{" ou "}"
- Mauvaise orthographe d'un des mots demandés, notamment `pushButton()`



Getting Started - Part 3

Objectifs

- ✓ Apprendre de nouveaux mots de vocabulaire en Anglais
- ✓ Écrire un code de plus de 15 lignes en autonomie
- ✓ Savoir résoudre un exercice inversé, i.e. dessiner la carte à partir du code

Pré requis

- ✓ Au minimum un ordinateur pour deux élèves et une bonne connexion internet
- ✓ La création en amont d'une session par le professeur/animateur
- ✓ Tous les élèves doivent être connectés avec leur identifiant de session

Plan de la leçon

1. Double Teleportation : Prendre le téléporteur Jaune dans la bonne direction
2. Little Antartica : Prendre le téléporteur Jaune puis le reprendre dans l'autre sens
3. The Not Complicated : Faire le tour par-dessus les petits rochers de gauche
4. Happy Winter : Rejoindre le téléporteur jaune pour arriver jusqu'au trésor
5. Between The Trees : Rejoindre le téléporteur violet puis le téléporteur orange
6. The White Snail : Un téléporteur bleu est caché derrière le premier arbre
7. The Tetris Island : Prendre le bon chemin en allant directement vers le bas
8. The Good Choice : Rejoindre le téléporteur bleu en haut de l'île
9. The Four Corners : Relier les téléporteurs entre eux avec le Level Editor
10. The Special One : Activer le téléporteur bleu puis l'éteindre pour accéder au jaune

Nos conseils

- Rappeler aux enfants la possibilité de créer des lignes grâce à la touche entrée
- Donner des pistes de réflexion à chacun des niveaux
- Les niveaux sont plus longs que les précédents, rassurer les enfants sur leur rapidité
- Laisser les enfants s'aider entre eux sur cette leçon

Astuces

- Lorsque deux téléporteurs sont côte à côte, il est possible de passer de l'un à l'autre, i.e. en sortant du premier, Cody rentre directement dans le second.

Les erreurs les plus fréquentes

- Suppression des accolades présentent au début et à la fin du texte
- Mauvaise orthographe d'un des mots demandés (ex : righth, duwn...)
- Dans un exercice inversé, ex : The Four Corner, il faut dessiner le chemin minimal menant Cody au trésor. Plus le chemin est approximatif, moins la note est bonne.



Procédures

Objectifs

- ✓ Aborder les notions de base en programmation
- ✓ Comprendre ce qu'est une procédure et comment l'utiliser
- ✓ Apprendre de nouveaux mots de vocabulaire en Anglais

Pré requis

- ✓ Au minimum un ordinateur pour deux élèves et une bonne connexion internet
- ✓ La création en amont d'une session par le professeur/animateur
- ✓ Tous les élèves doivent être connectés avec leur identifiant de session

Plan de la leçon

1. The Shapes : Appeler la procédure `othershape()` sur la ligne 7
2. The Tree : Inverser les appels de procédures aux lignes 5 et 6
3. The Big Fight : Appeler la procédure `bigFight` à la ligne 6
4. Two Palmtrees : Remplacer `down()` à la ligne 11 par `up()`
5. The Button : Compléter la procédure à partir de la ligne 12
6. Broken Bridges : Echanger les appels de procédures aux lignes 6 et 7
7. Black Box Island : Appeler la procédure `blackBox(...)` à la ligne 6
8. Islands : Appeler les procédures aux lignes 7/8 et compléter la ligne 25
9. Zigzag : Dessiner le chemin minimal avec le Level editor
10. Procedures : Mettre les bons noms de procédures aux lignes 7 et 8

Nos conseils

- Prendre le temps d'expliquer la notion de procédure (ou fonction) avant de commencer la leçon
- Ne pas hésiter à guider les enfants sur les deux premiers niveaux
- Il y a une concordance entre le code et la forme géométrique, raisonner par analogie par rapport à ce qui est déjà écrit

Astuces

- Cody ne peut pas se battre et sauter simultanément, il lui faut soit combattre le pirate en amont soit trouver un autre chemin

Les erreurs les plus fréquentes

- Mauvaise orthographe du nom d'une fonction
- Il est parfois nécessaire de modifier à plusieurs endroits : les endroits où il existe un bug sont indiquées avec la notation `//bug`
- Suppression involontaire de crochets au début ou à la fin d'une procédure



Parameters

Objectifs

- ✓ Comprendre la notion de paramètre d'entrée d'une procédure
- ✓ Comprendre la notation des paramètres dans une définition de procédure
- ✓ Savoir appeler une procédure en remplaçant les paramètres par des valeurs

Pré requis

- ✓ Au minimum un ordinateur pour deux élèves et une bonne connexion internet
- ✓ La création en amont d'une session par le professeur/animateur
- ✓ Tous les élèves doivent être connectés avec leur identifiant de session
- ✓ Avoir précédemment expliqué la notion de procédures

Plan de la leçon

1. Parameter : Mettre un 4 à la place du 3 pour la fonction snake() à la ligne 6
2. Snakes : Ajouter la fonction snake() sur la ligne 5 et 6 avec les bons paramètres
3. Lines : Ajouter la fonction line() à la ligne 6 et 7 avec les bons paramètres
4. The Ls : Dessiner le chemin minimal avec le Level editor
5. The Ls 2 : Ajouter la fonction shape() à la ligne 7 avec les bons paramètres
6. The Nested Snakes : Completer les lignes 13,14 et 15 avec les bons paramètres
7. The Cs : Completer la procédure C() qui correspond aux îles de droite
8. Teleporters : Ajouter island() à la ligne 5,6,7 avec les bons paramètres
9. The Intestine : Ajouter la fonction p() à la ligne 7 et 8 avec les bons paramètres
10. The Lines : Modifier le bug de la ligne 15 en remplaçant le 2 par le paramètre x

Nos conseils

- Commencer par expliquer la notation abstraite des paramètres (x, y, z, ...), cette notion fait office de « conteneur » qui prend la valeur numérique qu'on lui donne
- Bien prendre le temps de tester le code initial, la plupart du temps il suffit juste de faire le bon appel de procédure
- Montrer la concordance entre les noms de procédures et les formes géométriques

Astuces

- Prendre en compte le sens d'entrée/sortie des téléporteurs

Les erreurs les plus fréquentes

- Suppression involontaire de crochets au début ou à la fin d'une procédure
- Bonne réponse mais placée dans la mauvaise fonction
- Oubli des parenthèses derrière un mot/fonction



Loops : Repeat

Objectifs

- ✓ Aborder les boucles de répétitions plus en profondeur
- ✓ Comprendre ce que fait un bout de code dans une boucle de répétition
- ✓ Savoir associer à une itération une forme géométrique (une partie de l'île)

Pré requis

- ✓ Au minimum un ordinateur pour deux élèves et une bonne connexion internet
- ✓ La création en amont d'une session par le professeur/animateur
- ✓ Tous les élèves doivent être connectés avec leur identifiant de session

Plan de la leçon

1. Basic Repeat : Corriger le bug de la ligne 7 en mettant la bonne valeur
2. Bushes : Ecrire le code adéquat pour parcourir la première Île
3. Repeat : Chaque boucle repeat(...) équivaut à une moitié de l'Île
4. Skeletons : Chaque boucle repeat(...) correspond à une rangée de pirates
5. Nested Repeat : Le 1er repeat(...) correspond au nombre de rangées de pirates
6. Stairways : Dessiner le chemin minimal avec le Level editor
7. Skeletons : : Le 1er repeat(...) correspond au nombre de rangées de pirates
8. The W : Il faut modifier les boucles repeat(...) aux lignes 5 et 11
9. Two Loops : Compléter la deuxième boucle de répétition à la ligne 15
10. Battle : Ecrire fight() à la ligne 7 et 13, et right() à la ligne 8 et 14

Nos conseils

- Rappeler la première leçon ou la notion de répétition avait été abordée (The Snake)
- Rappeler que la modification à faire est souvent simple et indiquée par //bug
- Insister sur l'orthographe, le temps passé à écrire est plus long pour cette leçon

Astuces

- Cody ne peut attaquer qu'un seul squelette à la fois
- Cody ne peut attaquer que les squelettes sur les cases limitrophes

Les erreurs les plus fréquentes

- Suppression involontaire de crochets au début ou à la fin d'une procédure
- Suppression involontaire de crochets ou parenthèses dans une boucle repeat(...)
- Mauvaise orthographe d'un des mots demandés (ex : repete au lieu de repeat)
- Ecrire la solution dans la mauvaise boucle repeat(...)
- Oubli des parenthèses derrière un mot ou une fonction



Variables - Part 1

Objectifs

- ✓ Découvrir les variables et leur rôle dans un programme informatique
- ✓ Comprendre comment déclarer une variable
- ✓ Comprendre comment affecter une valeur à une variable
- ✓ Comprendre comment utiliser les variables à travers des puzzles simples

Pré requis

- ✓ Au minimum un ordinateur pour deux élèves et une bonne connexion internet
- ✓ La création en amont d'une session par le professeur/animateur
- ✓ Tous les élèves doivent être connectés avec leur identifiant de session
- ✓ Avoir déjà pratiqué la leçon Parameters

Plan de la leçon

1. The Increment Island : Mettre $x+2$ entre les parenthèses à ligne 10
2. The Variable Bandits : Incrémenter x de 1 ($x=x+1$) à la ligne 15
3. The Three Arms : Dessiner le chemin minimal avec le Level editor
4. The Three Arms Repeated : Incrémenter x de 1 ($x=x+1$) à la ligne 15
5. Take the Stairs : Ajouter `step(...)` à la ligne 11 et 12 avec le bon paramètre

Nos conseils

- Rappeler la leçon Parameters, en particulier comment les paramètres sont représentés par des noms abstraits (x, y, z, \dots). La même chose s'applique à une variable qui est une sorte de « conteneur » qui prend la valeur qu'on lui affecte
- Pour le début de la leçon, observer l'aspect géométrique et symétrique des îles
- La modification à faire est souvent simple et quasi-systématiquement indiquée en vert et qu'il s'agit le plus souvent d'un simple bug à corriger
- Prendre le temps d'expliquer la notion d'incrément, ex : $x=x+1$ se lit « x prend la valeur actuelle de x augmentée de 1 »

Les erreurs les plus fréquentes

- Oubli des parenthèses derrière un mot/fonction
- Suppression involontaire de crochets ou parenthèses devant un nom de procédure ou à l'intérieur d'une boucle de répétition
- Mauvaise écriture d'un signe, ex : $<$, $>$, $=$
- Oubli de la ligne qui incrémente x (ex : $x=x+1$) à partir du deuxième exercice



Variables - Part 2

Objectifs

- ✓ Découvrir les variables et leur rôle dans un programme informatique
- ✓ Découvrir la notion de portée (scope) d'une variable
- ✓ Comprendre comment utiliser les variables à travers des puzzles simples

Pré requis

- ✓ Au minimum un ordinateur pour deux élèves et une bonne connexion internet
- ✓ La création en amont d'une session par le professeur/animateur
- ✓ Tous les élèves doivent être connectés avec leur identifiant de session
- ✓ Avoir déjà pratiqué la leçon Variables – Part 1

Plan de la leçon

1. Variable : Affecter la valeur 4 à la variable step à la ligne 14 (step = 4)
2. Addition : L'île est ici découpée en 3, rajouter down() sur la ligne 16
3. The Spiral : Entre les deux down(), le serpent diminue de 4 cases : mettre $x=x-4$
4. Global Variable : Dans ce niveau Cody exécute chaque fonction 3 fois
5. Same Name : Ce niveau est explicatif, expliquer ce qu'il se passe
6. Parameter Name : Ce niveau est explicatif, expliquer ce qu'il se passe
7. Limited Scope : Ce niveau est explicatif, aucune modification n'est nécessaire

Nos conseils

- Rappeler aux élèves la leçon « Variables -Part 1 », pratiquée précédemment, en particulier les deux notions importantes : (1) comment déclarer une variable, et (2) comment affecter une valeur à une variable
- Prendre le temps d'expliquer la notion d'incrément, ex : $x=x+1$ se lit « x prend la valeur actuelle de x augmentée de 1 »
- Dans Same Name (portée d'une variable), demander aux élèves combien de pas Cody va vers le bas avant d'appuyer sur le bouton de lecture. Essayez de changer la valeur des étapes (1) globalement à la ligne 5 et (2) localement à la ligne 10 et faites deviner aux élèves le résultat de l'exécution
- Dans Parameter Name, demander aux élèves de deviner le nombre de pas que Cody prend vers la droite. Expliquer alors que le paramètre a plus de priorité que la variable globale, et donc c'est le 5 dans l'appel de procédure de la ligne 8 qui est pris

Les erreurs les plus fréquentes

- Suppression involontaire de crochets ou parenthèses devant un nom de procédure ou à l'intérieur d'une boucle de répétition
- Mauvaise écriture d'un symbole, ex $<$, $>$, $=$, $! =$



Loops : While

Objectifs

- ✓ Comprendre le fonctionnement d'une boucle while(...)
- ✓ Comprendre la condition d'arrêt d'une boucle while(...)
- ✓ Savoir utiliser la boucle while(...) pour résoudre un problème

Pré requis

- ✓ Au minimum un ordinateur pour deux élèves et une bonne connexion internet
- ✓ La création en amont d'une session par le professeur/animateur
- ✓ Tous les élèves doivent être connectés avec leur identifiant de session
- ✓ Avoir déjà pratiqué les leçons « Loops : Repeat » et « Variables – Part 1 et 2 »

Plan de la leçon

1. The While : Remplacer la valeur 4 par 5 sur la ligne 5
2. Minus One : La variable x ne sera utilisée que pour la fonction right(...)
3. Plus One : Ecrire $x=x+1$ pour incrémenter la valeur de x par 1
4. The Boa Island : Le serpent diminue de 2 cases, décrémenter x de 2 : $x=x-2$
5. Teleporters : La variable x ne sera utilisée que pour la fonction jump(...)
6. Skeletons : La variable x ne sera utilisée que pour la fonction right(...)
7. The Race : La variable x sera utilisée pour jump(...), incrémenter x de 1 : $x=x+1$
8. Multiplication (inversé) : Dessiner le chemin minimal avec le Level Editor
9. Spaceships : La variable x n'est utilisée que pour les fonctions right(...) et left(...)
10. The last While (inversé) : Dessiner le chemin minimal avec le Level editor

Nos conseils

- Expliquer la différence entre repeat(...) et while(...) : dans la première, le nombre de répétitions est connu à l'avance (sortie de la boucle au bout de n itérations), alors que dans la deuxième, une condition de sortie (expression booléenne) est évaluée à chaque itération pour déterminer si le programme sort ou non de la boucle
- La modification à faire est souvent simple et quasi-systématiquement indiquée en vert ; il s'agit le plus souvent d'un simple bug à corriger

Astuces

Pour activer/désactiver un téléporteur il faut appuyer sur le bouton de la couleur adéquate avec la fonction `pushButton()`

Les erreurs les plus fréquentes

- Suppression involontaire de crochets ou parenthèses devant un nom de procédure ou à l'intérieur d'une boucle de répétition
- Oubli d'incrémenter une variable à l'intérieur d'une boucle (ex : $x=x+1$)



Conditionals

Objectifs

- ✓ Découvrir les expressions conditionnelles en programmation
- ✓ Comprendre la structure logique d'un bloc : `if (condition) { ... } else { ... }`
- ✓ Savoir formuler des expressions conditionnelles pour résoudre un problème

Pré requis

- ✓ Au minimum un ordinateur pour deux élèves et une bonne connexion internet
- ✓ La création en amont d'une session par le professeur/animateur
- ✓ Tous les élèves doivent être connectés avec leur identifiant de session
- ✓ Avoir déjà pratiqué les leçons « Variables – Part 1 » et « Variables – Part 2 »

Plan de la leçon

1. Conditional : Mettre 4 dans la condition d'if, pour se battre au bon moment
2. The Button : Mettre la bonne valeur à la ligne du bug pour appuyer sur le bouton
3. Skeletons : Mettre la bonne valeur à la ligne du bug pour sauter au bon moment
4. Not Equal : Il n'est pas nécessaire d'augmenter de 1 à la ligne 9
5. Greater or Equal : Mettre la valeur 3 dans le bloc if pour combattre en premier
6. Else if : Remplacer la valeur ayant provoqué le bug
7. Or : Les boutons sont placés sur la 3ème et la 5ème colonne
8. Teleporters : Dans ce niveau on utilise la fonction `jump(...)` sauf pour la valeur 3
9. And : Il faudra se battre entre les valeurs 2 et 6
10. End of Conditionals : Rajouter une condition pour appuyer sur le bouton

Nos conseils

- Expliquer la structure d'un bloc `if (condition) { ... } else { ... }`
- Pour le début de la leçon, mettre en avant les similarités avec les leçons précédentes
- Prendre le temps d'expliquer la formulation d'une expression conditionnelle, notamment l'utilisation des opérateurs `>`, `<`, `==`, `!=`

Astuces

- Pour sauter dans une direction donnée, Cody doit prendre de l'élan

Les erreurs les plus fréquentes

- Oubli des parenthèses derrière un mot/fonction
- Suppression involontaire de parenthèses dans la condition du bloc if
- Suppression d'un crochet `{` ou `}` à la suite de la condition ou dans le bloc else
- Mauvaise écriture d'un symbole, ex : `<`, `>`, `==`, à l'intérieur de la condition du bloc if



Puzzles

Objectifs

- ✓ Maîtriser les notions acquises jusqu'à présent (procédures, repeat, while, if)
- ✓ Exploiter ces notions pour résoudre des niveaux avec une algorithmique avancée
- ✓ Savoir écrire des bouts de code plus ou moins complexes en autonomie

Pré requis

- ✓ Au minimum un ordinateur pour deux élèves et une bonne connexion internet
- ✓ La création en amont d'une session par le professeur/animateur
- ✓ Tous les élèves doivent être connectés avec leur identifiant de session
- ✓ Avoir déjà pratiqué la leçon « Loops : While » et « Loops : Repeat »

Plan de la leçon

1. Chess : Une fois le bon chemin trouvé, il suffit de sauter jusqu'au trésor
2. The Pyramid : Il faut se battre contre les rangées tout en avançant en diagonale
3. The Bridges : Appeler la fonction Go aux lignes 5, 6 et 7 avec les bons paramètres
4. Vertical and C's : Les noms des fonctions correspondent aux chemins pris par Cody
5. The Squares : Atteindre d'abord les boutons puis prendre les bons chemins
6. The Horde : Battre les pirates par colonnes en utilisant la fonction repeat(...)
7. Snake Twins : Créer un bloc while(...), les deux îles sont construites en miroir

Nos conseils

- Expliquer qu'il s'agit, dans la grande majorité, de problèmes de logique
- Être patient(e), ces niveaux demandent une part de réflexion importante
- Dans les exercices plus complexes (niveau 2 et 3), la solution demandée est souvent simple à écrire/corriger

Astuces

- Pour sauter dans une direction, Cody doit prendre de l'élan
- Pour activer/désactiver un téléporteur, appuyer sur le bouton de la même couleur
- Le sens de sortie d'un téléporteur est important : la direction prise en rentrant dans un téléporteur est la même que celle prise à la sortie du téléporteur

Les erreurs les plus fréquentes

- Suppression involontaire de blocs de code, notamment les déclarations de variables
- Suppression involontaire de crochets ou de parenthèses dans un appel de procédure, ou à l'intérieur d'un bloc de code (while, repeat, if, ...)
- Les formes géométriques étant simples, les enfants ont tendance à contourner le problème en effaçant le code qui leur est donné en énoncé



Loops : Nested

Objectifs

- ✓ Maîtriser notions acquises sur les boucles de répétition `repeat(...)` et `while(...)`
- ✓ Savoir imbriquer une boucle de répétition dans une autre
- ✓ Savoir déclarer et incrémenter des variables à l'intérieur des boucles

Pré requis

- ✓ Au minimum un ordinateur pour deux élèves et une bonne connexion internet
- ✓ La création en amont d'une session par le professeur/animateur
- ✓ Tous les élèves doivent être connectés avec leur identifiant de session
- ✓ Avoir déjà pratiqué les leçons Loops : Repeat et Loops : While
- ✓ Avoir déjà pratiqué les leçons Variables – Part 1 & 2 et Procedures

Plan de la leçon

1. Nested Whiles : Se déplacer de x cases à droite et de y cases en bas
2. Nested Whiles 2 : Même principe, mais les déplacements sont différents
3. Nested Whiles 3 : Ici, les îles diminuent de taille au lieu d'augmenter
4. Nested Whiles 4 : C'est un mix des trois exercices vus précédemment
5. Skeleton : Deux boucles `repeat(...)`, une pour la verticale, une pour l'horizontale
6. The Arrow : Deux boucles `repeat(...)`, une pour la montée, une pour la descente

Nos conseils

- Montrer aux enfants la légère évolution des îles à chacun des niveaux
- Rappeler la notion de variables, et expliquer que x et y représentent souvent les déplacements suivant l'axe horizontal et vertical de chacune des îles
- Même si la solution attendue peut-être un peu plus longue que d'habitude, la ligne de code où écrire est toujours indiquée

Astuces

- Pour activer ou désactiver un téléporteur il faut appuyer sur le bouton de la couleur adéquate et respecter le sens de direction des téléporteurs

Les erreurs les plus fréquentes

- Suppression involontaire de crochets ou de parenthèses lors d'un appel de procédure, ou à l'intérieur d'un bloc de code (`while`, `repeat`, `if`, ...)
- Confusion entre l'axe horizontal et vertical de l'île
- Les formes géométriques étant simples, les enfants ont tendance à contourner le problème en effaçant le code qui leur est donné en énoncé



Recursion

Objectifs

- ✓ Réviser toutes les notions acquises jusque-là dans le parcours
- ✓ Comprendre le concept de récursivité dans un programme
- ✓ Avoir une intuition visuelle de la récursivité
- ✓ Illustrer la récursivité à travers une suite de Fibonacci

Pré requis

- ✓ Au minimum un ordinateur pour deux élèves et une bonne connexion internet
- ✓ La création en amont d'une session par le professeur/animateur
- ✓ Tous les élèves doivent être connectés avec leur identifiant de session
- ✓ Avoir déjà pratiqué toutes les leçons du catalogue avant celle-ci

Plan de la leçon

1. Recursion : Il suffit d'appeler la fonction `recursion(...)` avec le paramètre $x+1$
2. Fibonacci : Ce niveau est explicatif, aucune modification n'est nécessaire, il sert d'introduction à la suite de Fibonacci. Ici on s'arrête au 7ème élément de la suite

Nos conseils

- Prendre le temps d'introduire la notion de récursivité avec des exemples simples
- Expliquer que la récursivité peut être vue comme une sorte de « mise en abîme »
- Pour illustrer la récursivité, s'inspirer du logo de la « Vache qui rit » dont les boucles d'oreilles contiennent l'image de la « Vache qui rit », dont les boucles d'oreilles contiennent l'image de la « Vache qui rit », etc.
- Un programme informatique ne peut pas supporter une infinité d'appels récursifs. Il se bloque (bug) car il ne peut pas chainer des fonctions indéfiniment. Pour cette raison, nous devons écrire une condition de sortie qui doit être vérifiée systématiquement par la fonction pour décider de continuer ou non
- Les suites de Fibonacci sont un bon exemple de récursivité, s'inspirer d'exemples dans la nature, montrer des photos de plantes, de coquillages, etc.
- Dans cette suite de Fibonacci, un élément est égal à la somme des deux éléments précédents. Dérouler sur un tableau ce qu'il se passe en partant de deux éléments $\text{fibonacci}(0) = 1$ et $\text{fibonacci}(1) = 1$. L'élément suivant, i.e. $\text{fibonacci}(2) = 1 + 1 = 2$. L'élément suivant, i.e. $\text{fibonacci}(3) = 2 + 1 = 3$, etc.

Les erreurs les plus fréquentes

- Suppression involontaire de crochets ou de parenthèses devant un nom de procédure, ou à l'intérieur d'un bloc de code (`while`, `repeat`, `if`, ...)
- Les formes géométriques sont simples, les enfants ont tendance à contourner le problème en effaçant le code qui leur est donné en énoncé



Objectifs

- ✓ Faire une synthèse de toutes les bases acquises jusqu'à présent
- ✓ Utiliser des opérateurs mathématiques dans un programme
- ✓ Savoir identifier des patterns visuels et leur associer des fonctions

Pré requis

- ✓ Au minimum un ordinateur pour deux élèves et une bonne connexion internet
- ✓ La création en amont d'une session par le professeur/animateur
- ✓ Tous les élèves doivent être connectés avec leur identifiant de session
- ✓ Avoir déjà pratiqué les leçons « Procédures » et « Variables – Part 1 & 2 »

Plan de la leçon

1. Multiplication : Pour compléter la fonction $\text{zeta}(\dots)$, observer le ratio « nombre de pas vers la droite » / « nombre de pas vers le bas », en déduire que le nombre de pas vers le bas vaut $2 \cdot x$
2. Division : Dessiner le chemin minimal à l'aide du Level Editor
3. The Third : Dessiner le chemin minimal à l'aide du Level Editor

Nos conseils

- Rappeler aux enfants la leçon « Variables -Part 1 », pratiquée précédemment, en particulier les deux notions importantes : (1) comment déclarer une variable, et (2) comment affecter une valeur à une variable
- Pour le début de la leçon, observer l'aspect géométrique qui se répète (pattern) et expliquer que chaque pattern est parcouru par la fonction $\text{zeta}(\dots)$
- Lors d'un appel de fonction $\text{zeta}(2 \cdot x)$, la multiplication $2 \cdot x$ est évaluée avant l'appel effectif de la fonction. En règle générale, la précedence des opérateurs/fonctions va toujours de la droite vers la gauche

Astuces

- L'instruction `jump()` doit nécessairement figurer dans la fonction qui s'applique aux patterns visuels (par exemple la fonction $\text{zeta}(\dots)$ du premier niveau) car cette instruction permet de passer au pattern suivant

Les erreurs les plus fréquentes

- Suppression involontaire de crochets ou de parenthèses devant un nom de procédure, ou à l'intérieur d'un bloc de code (`while`, `repeat`, `if`, ...)
- Les formes géométriques étant simples, les enfants ont tendance à contourner le problème en effaçant le code qui leur est donné en énoncé