



Ressources pour le cycle terminal général et technologique

Informatique et Sciences du Numérique

Premiers pas en Python avec Rurple

Ces documents peuvent être utilisés et modifiés librement dans le cadre des activités d'enseignement scolaire, hors exploitation commerciale.

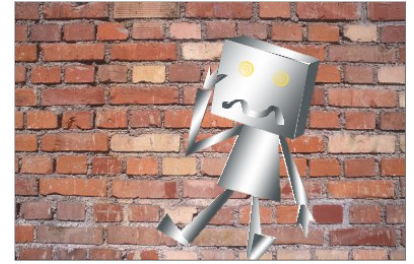
Toute reproduction totale ou partielle à d'autres fins est soumise à une autorisation préalable du Directeur général de l'enseignement scolaire.

La violation de ces dispositions est passible des sanctions édictées à l'article L.335-2 du Code la propriété intellectuelle.

Juin 2012

Présentation

Premiers pas en Python avec Rurple



1 / Thème abordé

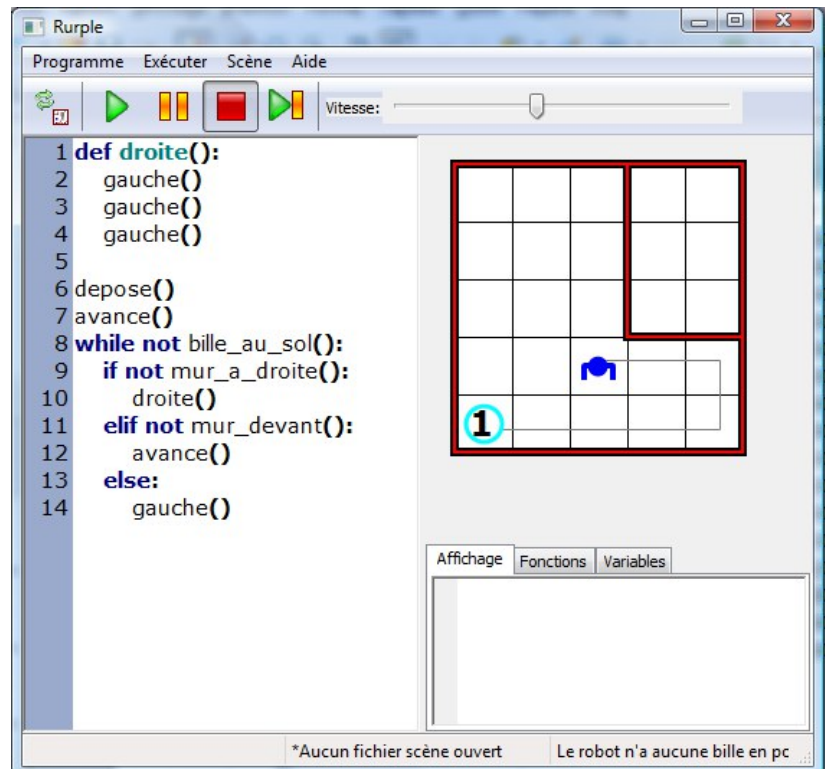
1.1 Problématique, situation d'accroche

Tout au long de l'année les élèves vont étudier et s'approprier les concepts de l'informatique, très souvent en les manipulant à travers un langage de programmation. Apprendre à s'exprimer aisément dans ce langage est par conséquent une tâche primordiale pour les élèves. Il semble judicieux de centrer les premières activités autour de cet objectif. Par la suite les activités se concentreront sur les autres concepts de l'informatique, mais par effets de bord, permettront également de consolider et d'enrichir les « compétences linguistiques » des élèves.

Le thème proposé dans cette ressource est visuel et dynamique : il s'agit de déplacer un robot (virtuel) dans un espace clos. Les notions de programmation en jeu sont élémentaires. L'état du système étant « transparent » (position du robot, présence de billes sur le sol ou dans la poche du robot, etc), on peut concevoir des programmes assez élaborés tout en se passant de l'usage de variables.

1.2 Frontières de l'étude et prolongements possibles

Les programmes que les élèves écrivent avec Rurple sont de vrais programmes Python mais avec des fonctions supplémentaires qui permettent de « piloter » le robot. Ce même environnement permet donc d'écrire des programmes Python quelconques qui laissent de côté le robot, en bénéficiant de ses outils de mise au point (mode d'exécution pas à pas, observation du contenu des variables) tout en conservant une grande simplicité.



Quant au pilotage de « vrais » robots, on en est encore loin car le positionnement dans l'espace d'un robot mobile n'est pas si simple, les « vrais » mouvements n'étant jamais identiques aux mouvements prescrits par la commande envoyée. Néanmoins, l'usage de Rurple permet un premier contact avec les problématiques du déplacement relatif dans l'espace ...

2 / Objectifs pédagogiques

2.1 Prérequis

Les élèves ne sont pas totalement débutants puisqu'ils ont été régulièrement entraînés depuis leur entrée au lycée à écrire de courts programmes (basés sur les structures d'alternative et d'itération) dans le cadre du cours de mathématiques. Les pratiques et les langages utilisés ont cependant pu être assez disparates, même au sein d'un même établissement, c'est pourquoi la première des activités que nous proposons ici ne suppose aucun prérequis. L'aisance et la rapidité avec lesquelles ses élèves la traiteront sera l'occasion pour l'enseignant de juger de leurs aptitudes dans ce domaine.

2.2 Éléments du programme

Contenus

- Élaboration progressive d'un algorithme (méthode opérationnelle visant à résoudre en un nombre fini d'étapes toutes les instances d'un problème donné).

- Éléments de base d'un langage de programmation (séquence, test, boucle). Insertion de commentaires dans un programme. Double usage du langage, lisible par un humain et interprétable par une machine. Choix de jeux de données pour tester un programme.

Compétences et capacités

Concevoir et réaliser une solution informatique en réponse à un problème :

- Concevoir et programmer un algorithme.
- Modifier un algorithme existant pour obtenir un résultat différent.
- Mettre un programme au point en le testant, en l'instrumentant.
- Développer, concevoir l'entête d'une fonction puis la fonction elle-même.
- Utiliser un outil de mise au point (mode pas à pas).

Communiquer à l'écrit et à l'oral :

- Commenter, documenter un programme.

3 / Modalités de mise en œuvre

3.1 Durée prévue pour la partie se déroulant en classe

La fiche propose deux activités :

- Activité n°1 – *Premiers pas avec Rurple*
Prévoir 2 heures. Il faut tenir compte de la prise en main de l'environnement logiciel par les élèves. Il peut être utile de prévoir des petits exercices additionnels pour les élèves les plus rapides.
- Activité n°2 – *Explorations*

Ces deux activités sont loin d'épuiser le potentiel pédagogique du logiciel Rurple. Le travail peut être poursuivi dans diverses autres directions. Les enseignants sont invités à imaginer et concevoir leurs propres activités ... et à les partager.

3.2 Type de l'animation

Les activités sont prévues pour que les élèves puissent travailler seuls et en autonomie sur un ordinateur. Les fiches élève fournissent des explications détaillées ce qui permet d'envisager qu'une partie du travail puisse être mené par les élèves chez eux. Selon le contexte et selon leurs propres habitudes, les enseignants pourront préférer fournir aux élèves des fiches moins détaillées, voire pas de fiches du tout.

3.3 Production des élèves

Dans la deuxième activité (*Explorations*), les élèves élaborent progressivement leur programme pour aboutir à un texte bien structuré et correctement documenté, aux fonctionnalités clairement spécifiées. Cette tâche de rédaction demande du temps et de la réflexion et il convient mieux de la dédier à un travail à faire à la maison.

3.4 Évaluation

Il peut être instructif pour l'enseignant d'étudier les productions de ses élèves pour mieux identifier les faiblesses ou les contre-sens de leurs textes et donc cerner où porter les remédiations. À ce stade initial de l'apprentissage il n'est pas souhaitable que ces productions soient notées.

La « performance » des élèves par rapport aux compétences listées en 2.2 peut servir de point de repère pour une évaluation ultérieure de ces compétences et donc des progrès des élèves.

4 / Outils

Les activités proposées utilisent l'environnement **Rurple NG-fr**, disponible à l'adresse:

<http://dichotomies.fr/2011/infomath/guides/python/installation-rurple>

5 / Auteur

Denis Pinsard, professeur de Mathématiques, académie de Rennes

Premiers pas en Python avec Rurple

1 / L'installation

Les activités proposées sont construites à partir de l'environnement logiciel **Rurple**. L'auteur de la version originale, André Roberge, explique sur le web¹ le but de Rurple² :

« *RUR-PLE est un environnement conçu pour vous aider à apprendre la programmation informatique avec le langage Python. À partir d'un monde imaginaire où un robot peut être programmé pour accomplir différentes tâches, vous allez apprendre ce qu'on veut dire par composer un programme informatique, en utilisant la syntaxe du langage Python. Par la suite, vous pourrez utiliser vos nouvelles connaissances dans l'environnement plus traditionnel de l'interprète inclus.* »

Paul Crowley, enthousiasmé par **RUR-PLE**, en a fait une nouvelle version qu'il a baptisée **Rurple NG**³. Il a simplifié le logiciel sur certains aspects et apporté quelques petites améliorations. La licence libre du logiciel a permis d'en proposer facilement une version francisée avec quelques modifications mineures. Cette version, baptisée **Rurple NG-fr**, est celle que nous utilisons ici. Elle est disponible en téléchargement à l'adresse:

<http://dichotomies.fr/2011/infomath/guides/python/installation-rurple>



2 / Le premier contact

Voir la fiche-professeur fournie en annexe et également disponible à l'adresse :

<http://dichotomies.fr/2011/infomath/activites/python/premiers-pas-avec-rurple/fiche-prof/>

Le robot obéit à quelques ordres élémentaires (avance, tourne à gauche, dépose une bille, ...). Après avoir utilisé les touches du clavier pour donner ses ordres, l'élève les couche par écrit, c'est-à-dire rédige son premier programme. L'immense potentialité de cette nouvelle forme de dialogue n'apparaîtra évidemment pas d'emblée à l'élève.



Le programme : c'est un texte qui suit des règles de grammaire bien précises autrement dit qui est écrit dans un langage. Ce texte est destiné à être lu et interprété par une machine pour provoquer des effets recherchés par le rédacteur.

Le robot peut être perçu par l'élève comme un « avatar » de l'interpréteur. Cette image peut aider à penser le processus d'interprétation. L'élève comprend aussi que ce n'est qu'une image et cela amène la question de « qu'est-ce qui interprète » et du « comment cela est-il interprété ». C'est un des objectifs de l'année de lever un peu le voile sur ce mystère. Face à la banalité quotidienne des ordinateurs la première étape consiste sans doute à prendre conscience qu'il y a bien un mystère.

L'élève va cependant faire l'expérience d'un robot qui ne se comporte pas du tout comme prévu ou, plus déroutant encore, de l'apparition de boîtes de dialogue ou de messages en anglais signalant visiblement une anomalie.

Le bug : ce n'est pas un accident mais plutôt un état normal d'un système informatique⁴. La rigidité de l'interpréteur, sa nature « mécanique » s'oppose au mode de pensée et au langage humain beaucoup plus souple, approximatif ou elliptique. La mise au point d'un programme est ainsi une phase incontournable, à moins de recourir à un « style » de programmation empêchant une partie des bugs.

1 <http://rur-ple.sourceforge.net/fr/rur.htm>

2 *Rurple* est un jeu de mots, entre purple (la couleur pourpre) et RUR (*Rossum's Universal Robots*), titre du roman (devenu pièce de théâtre) du Tchèque Karel Čapek qui faisait intervenir des créatures artificielles nommées robots (*robota*, en tchèque, signifie travailleur). L'image ci-dessus reproduit l'affiche d'époque (source : Wikimedia).

3 <http://www.lshift.net/blog/2009/08/13/teaching-programming-rur-ple-and-rurple-ng>

4 Voir la conférence de Gérard Berry : <http://www.universcience.fr/fr/conferences-du-college/seance/c/1248116241161/-/p/1239022827697/> et l'émission suivante sur France Culture :

<http://www.franceculture.fr/emission-place-de-la-toile-autopsie-du-bug-2011-02-20.html>

Confronté à un bug, l'élève sollicite naturellement l'aide du professeur afin qu'il résolve le problème technique qui a surgit sur sa machine. La tâche de l'enseignant est alors d'expliquer qu'il n'y a pas d'anomalie et que l'origine du comportement est à rechercher dans le texte du programme. Il s'agit ensuite d'enseigner quelques méthodes d'analyse et de mise au point afin de rendre l'élève de plus en plus autonome face au bug.

L'élève rencontre ensuite la notion de **procédure** et s'en sert afin d'écrire un texte plus concis.

L'enseignant interroge les élèves sur les avantages que peut procurer cette syntaxe offerte par le langage : mise au point plus facile, texte mieux structuré et donc plus lisible, etc.

L'élève rencontre pour finir deux notions fondamentales du langage, qui vont lui faire entrevoir les potentialités de cette forme de communication avec la machine.

- Le langage ne permet pas uniquement de dicter des instructions, il est également possible de récupérer de l'information sur l'environnement du robot. Plus tard il s'agira de lire la valeur de variables.
- Le langage offre aussi une syntaxe qui permet de répéter indéfiniment une même séquence d'instructions.

La conjonction de ces deux notions permet d'écrire un programme beaucoup plus souple, capable de s'adapter à diverses situations et différentes formes de pièces.

Toutefois, le programme s'allongeant devient plus complexe et davantage susceptible de produire des dysfonctionnements peu évidents. La chasse aux bugs devient alors cruciale...

L'instrumentation : une méthode simple et relativement efficace pour analyser les dysfonctionnements d'un programme consiste à l'instrumenter c'est-à-dire à parsemer le programme d'instructions d'affichage du contenu de certaines variables « sensibles ».

Il est commode de déclarer un variable « globale » nommée debug et de faire précéder les affichages par une instruction conditionnelle (ici en langage Python) :

```
if debug:
    print "Valeur de xrobot:",xrobot,"\n"
```

Lorsque le programme est au point, il suffit d'affecter à debug la valeur 0 et les affichages sur la console ne se font plus.

Une fois que son programme est au point, l'élève ajoute un en-tête descriptif à son texte.

Les commentaires et la documentation : le texte « source » d'un programme est aussi destiné à être lu par des humains et sa lisibilité (immédiate ou décalée dans le temps) est une de ses qualités essentielles; cela se fait au moyen de commentaires placés à des endroits stratégiques, de variables ayant des noms descriptifs, et d'une notice explicative détaillée.

Documenter le premier programme permet de marquer les esprits et d'installer l'idée que la compréhension du texte par un humain est aussi importante que son interprétation correcte par la machine.

3 / Des explorations

Voir la fiche-professeur fournie en annexe et également disponible à l'adresse:

<http://dichotomies.fr/2011/infomath/activites/python/explorations/fiche-prof>

Après avoir appris les premiers rudiments du langage lors de l'activité *Premiers pas avec Rurple*, l'élève en sait déjà suffisamment pour porter son esprit sur la résolution d'un problème algorithmique. Il s'agit de mettre au point un programme qui fait faire au robot le tour de la pièce, quelque soit la forme de celle-ci. L'idée proposée est de faire déposer au robot une bille sur la case de départ, puis de lui faire longer le mur jusqu'à ce qu'il retrouve sa bille. L'élaboration du programme est progressive. On lui propose un premier programme de six lignes, qui sur le papier peut paraître fonctionner si on n'y regarde pas de suffisamment près. L'élève est donc invité à examiner le programme puis à le tester. Il doit alors diagnostiquer l'origine du problème, trouver une solution et modifier son programme en conséquence, jusqu'à obtenir le comportement désiré. L'histoire n'est pas terminée, elle ne fait que commencer. Le programme fonctionne dans cette pièce carrée, mais que se passe-t-il si sa forme n'est plus rectangulaire ? L'élève entre alors dans un nouveau cycle d'analyse, de modifications, de mise au point et de validation de son programme, suivi de sa confrontation à une situation non encore envisagée qui nécessite un nouveau cycle.

Au fur et à mesure que l'activité se déroule, l'élève s'habitue à la démarche et l'enseignant le pousse non seulement à gagner en autonomie dans la mise au point mais aussi à prendre de lui-même des initiatives pour tester son programme dans des situations nouvelles et rendre son texte plus général et plus robuste.

À un certain moment l'élève va rencontrer un problème difficile qui va même remettre en cause le principe de l'algorithme : le retour sur la case de départ ne signifie pas nécessairement que le tour de la pièce soit achevé. Il s'agit donc de repenser l'algorithme. Les élèves doivent imaginer une solution.

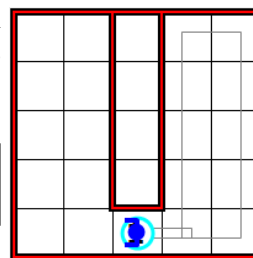
L'enseignant dirige le débat d'idées dans la classe jusqu'à voir émerger une solution consensuelle.

Il faut aussi s'interroger sur la façon de traduire cette idée dans le programme lui-même : le robot doit conserver le souvenir de certaines de ses actions. Cela conduit naturellement à la notion de variable qui apparaît pour ce qu'elle est : une case mémoire.

Le dernier temps de l'activité va être consacré à « toiletter » le programme, c'est-à-dire à produire un texte qui soit le plus clair possible pour les lecteurs humains :

- organiser le texte en paragraphes qui ont une unité logique,
- regrouper éventuellement une séquence sous la forme d'une procédure,
- modifier les noms des variables ou des procédures si ceux-ci ne sont pas suffisamment clairs,
- ajouter des en-têtes en début de programme et aux débuts des procédures pour dire ce qu'elles font,
- ajouter des commentaires pour expliquer des passages qui ne sont pas clairs par eux-mêmes.

L'enseignant montre l'importance qu'il accorde à ce travail en argumentant ses raisons et en veillant à ce qu'il ne soit pas trop rapidement expédié. Il cherche à susciter chez ses élèves le désir de produire un beau texte ...



4 / Références

4.1 À propos de Rurple

- RUR-PLE : la version originale de Rurple, téléchargeable ici : <http://sourceforge.net/projects/rur-ple/files/rur-ple/>
- Roberge André, « Apprendre le langage Python avec RUR-PLE : un jeu d'enfants ! » [en ligne] <http://rur-ple.sourceforge.net>
- Crowley Paul, « Teaching programming, RUR-PLE, and Rurple NG » [en ligne] <http://www.lshift.net/blog/2009/08/13/teaching-programming-rur-ple-and-rurple-ng>
- Pinsard Denis, « Rurple NG-fr » <http://dichotomies.fr/2011/infomath/guides/python/presentation-rurple>
- Rurple-NG-fr : la version francisée de Rurple-NG, téléchargeable ici : <http://dichotomies.fr/2011/infomath/guides/python/installation-rurple>
- Des vidéos de prise en main ont été réalisées par les IREM, on les trouve ici : <http://www.univ-irem.fr/videos/PackageTutoRurple/>

4.2 À propos de Python

- La page officielle du langage Python permet en particulier de télécharger le logiciel : « Python Programming Language – Official Website » <http://www.python.org>
- On trouve une installation de Python spécialement conçue pour l'usage en lycée, sous le nom « Amiens-Python », disponible sur SIALLE : <http://www.cndp.fr/sialle/fiche-detaillee-amiens-python-399.php>
- De nombreuses ressources et tutoriels sont disponibles sur l'internet pour apprendre le Python par soi-même. La page d'Olivier Berger est un point d'entrée parmi d'autres : <http://www.olivierberger.org/python>
- On peut aussi se servir de l'ouvrage de Gérard Swinnen : <http://inforef.be/swi/python.htm>