



Problématique n°1 :

Comment faire acquérir les connaissances des programmes des LGT et LP dans des situations favorables à l'utilisation de l'algorithmique et la programmation tout en réinvestissant les compétences du cycle 4 ?

Objet d'étude :

Etude des équations avec l'algorithmique et la programmation.

Sommaire

I.	Présentation synthétique du projet	3
II.	Analyse des différents travaux menés au collège	4
1.	Collège : TP informatique 1 « Programmer pour trouver des solutions » (Voir Annexe 1).....	4
2.	Collège : TP informatique 2 « Découverte des listes » (Voir Annexe 2)	8
3.	Collège : TP informatique 3 « Utilisation des scripts précédents » (Voir Annexe 3)	8
4.	Collège : Activité « L’aquarium de Bubulle et Didier » (Voir Annexe 4)	10
III.	Analyse des différents travaux menés en LGT.....	14
1.	LGT : Activité « L’aquarium » (Voir Annexe 5)	14
2.	LGT : TP informatique 1 « Résoudre une équation avec Python » (Voir Annexe 6).....	14
3.	LGT : Jeux informatiques « Un monstre et un chat » et « Devine un nombre » (Voir Annexe 7).....	15
4.	LGT : TP informatique 2 : « Un premier algorithme de dichotomie » (Voir Annexe 8)	17
IV.	Analyse des différents travaux menés en LP	18
5.	LP : Activité « L’emballage » (Voir Annexe 9)	18
V.	Bilan : passage du logiciel Scratch au logiciel Python	19
VI.	Annexes : énoncés des différents travaux.....	20
1.	Collège : TP informatique 1 « Programmer pour trouver des solutions ».....	20
2.	Collège : TP informatique 2 « Découverte des listes »	21
3.	Collège : TP informatique 3 « Utilisation des scripts précédents »	22
4.	Collège : Activité « L’aquarium de Bubulle et Didier »	22
5.	LGT : Activité « L’aquarium »	24
6.	LGT : TP informatique 1 « Résoudre une équation avec Python »	25
7.	LGT : Jeux informatiques « Un monstre et un chat » et « Devine un nombre ».....	26
8.	LGT : TP informatique 2 « Un premier algorithme de dichotomie »	27
9.	LP : Activité « L’emballage »	29

I. Présentation synthétique du projet

<i>Description rapide du projet</i>	<ul style="list-style-type: none">- Au collège et 3^{ème} prépa pro : Découverte de la notion d'équation avec la programmation par Scratch de tests au hasard, de tests par balayage puis de tests par balayage ciblé par l'utilisateur.- En LGT : Reprise de la programmation par Python pour rechercher des solutions à des équations dont la méthode de résolution n'est pas connue en seconde. Approche de la méthode de dichotomie comme amélioration de la méthode par balayage.- En LP : Utilisation d'un nouvel outil TiC (Scratch puis Python) pour formaliser une démarche de résolution du premier degré en seconde ; puis, en première, pour la résolution d'une équation du second degré. Utilisation d'une méthode par balayage (via scratch) pour faire une étude rapide lors de la résolution d'une problématique (recherche d'un extremum par exemple).
<i>Niveaux concernés</i>	<ul style="list-style-type: none">- Au collège : quatrième et troisième- En LGT : seconde- En LP : 3^{ème} prépa pro, seconde, première et Terminale
<i>Logiciels</i>	Scratch et Python
<i>Objectif général</i>	Aborder les équations avec l'algorithmique et la programmation en réinvestissant les compétences du cycle 4 en LGT et LP et en assurant une transition du logiciel Scratch vers Python

II. Analyse des différents travaux menés au collège

1. Collège : TP informatique 1 « Programmer pour trouver des solutions » (Voir Annexe 1)

Ce premier TP informatique propose aux élèves de créer des programmes sur Scratch qui permettent de trouver des solutions à différentes situations en procédant à différents tests. Le but est de montrer aux élèves, à la fin, que des programmes de simple test au hasard ne suffisent pas lorsque les recherches deviennent plus sophistiquées.

Analyse a priori

Objectif :

Le but de cette activité est de laisser les élèves programmer afin de trouver des solutions à des équations (sans évoquer d'ailleurs le terme d'équations). En créant par eux-mêmes des programmes, l'objectif attendu est qu'ils découvrent les limites de leurs programmes qui peuvent être trop simples dans certaines situations et qu'ils se posent la question d'une amélioration possible à leur programme qui pourrait alors être utilisable pour n'importe quelle solution.

Situations 1 et 2 :

Ils peuvent créer des programmes très simples qui donnent le résultat de la suite d'opérations. Le programme réalise alors ce que pourrait faire une calculatrice, avec l'avantage qu'il n'y a pas besoin de retaper à chaque fois la suite de calculs. Donner uniquement le nombre choisi au départ permet d'obtenir le résultat final. Les nombres attendus n'étant pas trop compliqués, ce genre de programme suffit et permet de trouver la solution en testant plusieurs nombres. Certains élèves peuvent utiliser le côté plus ludique de Scratch en utilisant le « si sinon ». Le programme peut alors faire le calcul attendu et tester si le résultat obtenu est le résultat attendu. Si les élèves ne font pas apparaître le résultat du calcul, ils devraient se rendre compte qu'il est alors difficile de trouver une solution car ce programme ne donnerait aucune piste, aucune direction de recherche. Certains élèves, à cette étape, n'auront peut-être pas besoin de programmer pour trouver les solutions. Ce besoin se fera alors ressentir pour eux dans les situations suivantes.

Certains élèves vont peut-être utiliser la boucle « répéter jusqu'à » pour pouvoir tester de façon plus aisée plusieurs nombres.

Situation 3 :

Il faut maintenant comparer les résultats de deux suites d'opérations. Demander au lutin d'afficher les résultats des deux suites d'opérations semble nécessaire. Par la suite, les élèves devraient voir la difficulté de chercher à tâtons une solution. Certains vont peut-être écrire sur une feuille de papier les résultats obtenus et chercher ainsi un nombre qui fait en sorte que les résultats soient de plus en plus proches. L'idée est alors de leur demander de faire en sorte que le programme donne tous les résultats, pour ne plus avoir à écrire sur une feuille à côté. Le programme doit se suffire à lui-même. Le but est alors que les élèves comprennent l'intérêt pour ce genre de recherches de faire apparaître les résultats des deux suites d'opérations pour tous les nombres que l'on propose, à l'image d'un tableur.

Améliorations possibles :

Un regard critique sur leurs propres programmes est demandé aux élèves afin qu'ils réfléchissent mieux à l'utilisation de la programmation. Plusieurs programmes sont possibles, les programmes simples trouvent rapidement leurs limites et les programmes plus compliqués demandent des manipulations que les élèves ne connaissent peut-être pas. L'intérêt est de bien souligner le fait que la programmation est ici au service d'une recherche mathématique. Plus cette recherche est sophistiquée, plus le programme doit se donner les moyens de faire la bonne recherche. Une amélioration de la programmation, nécessaire à la recherche de solutions de plus en plus difficiles à trouver, est une étape à comprendre pour bien faire le lien entre le collège et le lycée.

Situation 4 :

Cette situation, réservée aux plus rapides, est proche de la situation 3. Toutefois, la forme change, et le nombre à trouver est plus grand.

Analyse a posteriori

Difficulté :

Une difficulté a été rencontrée par plusieurs groupes. Ces derniers emboîtaient mal les blocs opératoires et ne programmaient donc pas le bon calcul. Un travail de programmation des calculs est donc intéressant à faire avec les élèves au préalable, notamment en début de cycle 4 quand les priorités dans les calculs sont abordées. Cette difficulté a été exprimée par plusieurs élèves :

Améliorations possibles :

Indiquer les avantages et les inconvénients de chacun des programmes réalisés et indiquer les améliorations possibles qui permettraient de créer des programmes plus performants pour répondre à ce genre de question :

m.1.° : c'pau... m'avons pas réussi à emboîter les bons blocs
m.2.° : c'pau → avons souvent des problèmes d'écart au niveau du résultat

Scripts réalisés par élèves :

Deux groupes d'élèves n'ont pas réalisé de scripts « d'essais-erreurs » pour les deux premières situations, mais ont programmé une remontée des calculs pour trouver la solution. Cette méthode s'est révélée être efficace car elle donnait toute de suite le bon résultat. Mais ces groupes ont compris qu'ils ne pouvaient plus faire la même chose pour la situation 3. Comme les élèves étaient par binômes, ceux-ci ne seront pas restés bloqués et ont réussi à penser autrement pour programmer un script à la situation 3.



Améliorations possibles :

Indiquer les avantages et les inconvénients de chacun des programmes réalisés et indiquer les améliorations possibles qui permettraient de créer des programmes plus performants pour répondre à ce genre de question :

L'avantage des deux premiers programmes est qu'on avait déjà le résultat donc c'est plus rapide et le 3eme programme prend plus de temps car il faut chercher un nombre précis qui n'est pas connu dans le cahier.

Les élèves ont réalisé des scripts simples demandant à l'utilisateur de choisir un nombre, puis le lutin donnait le résultat de la suite de calculs. Les situations étant de plus en plus compliquées, les élèves ont vite compris que cette méthode pouvait prendre du temps et l'ont formulé de différentes façons :



Améliorations possibles :

Indiquer les avantages et les inconvénients de chacun des programmes réalisés et indiquer les améliorations possibles qui permettraient de créer des programmes plus performants pour répondre à ce genre de question :

Difficulté : on ne sait pas où chercher si il est positif ou négatif
Autres difficulté on a des nombres totalement opposé, Comment trouver un nombre commun à ses nombres opposé

Améliorations possibles :

Indiquer les avantages et les inconvénients de chacun des programmes réalisés et indiquer les améliorations possibles qui permettraient de créer des programmes plus performants pour répondre à ce genre de question :

Un encadrement de nombre
pas besoin de calculer dans sa tête
et faudrait qu'ils nous marquent plusieurs pour
exemple entre a et b et que on est pas besoin
d'écrire juste de cliquer

Améliorations possibles :

Indiquer les avantages et les inconvénients de chacun des programmes réalisés et indiquer les améliorations possibles qui permettraient de créer des programmes plus performants pour répondre à ce genre de question :

les inconvénients sont : le temps, comme on essaie
des nombres au hasard, ça peu prendre du temps
pour trouver le bon nombre

L'avantage c'est que les calculs se font automatiquement

Il faudrait une option qui nous donne les calculs
possibles pour trouver un nombre donné.

2. Collège : TP informatique 2 « Découverte des listes » (Voir Annexe 2)

Suite au premier TP informatique, il a été convenu que garder des scripts trop simples ne permet pas toujours de trouver facilement la solution recherchée. Ce TP informatique a pour but de faire découvrir l'utilisation des listes sur Scratch aux élèves. Ceux-ci pourront alors utiliser ces dernières pour trouver les solutions aux situations pour lesquelles ils ne trouvaient pas de solution auparavant. Par la suite, les élèves découvrent la recherche de solution par balayage. Celui-ci devient de plus en plus précis au cours du TP informatique. Tous les élèves ne peuvent pas atteindre la fin du TP. Ce TP doit être abordé par les élèves en fonction de leur compétence en programmation. L'objectif qui doit être atteint par tout le monde est l'acquisition de l'utilisation des listes. Les élèves ayant développé de plus grandes capacités en programmation peuvent aborder plus facilement la partie « balayage ». Ce TP informatique a pour but d'apporter aux élèves des connaissances supplémentaires pour répondre aux questions de recherche de solutions aux diverses situations et fournira aux élèves qui le souhaitent des outils déjà programmés et réutilisables pour le troisième TP informatique.

3. Collège : TP informatique 3 « Utilisation des scripts précédents » (Voir Annexe 3)

Ce troisième TP informatique propose aux élèves de trouver la solution à une situation similaire aux situations du premier TP informatique, mais la solution recherchée est plus difficile à trouver. Les élèves sont alors contraints de programmer des scripts plus sophistiqués s'ils veulent trouver la solution.

Analyse a priori

La situation 1 est formulée de la même façon que dans le premier TP informatique afin que les élèves reconnaissent rapidement ce qu'ils doivent faire. Cette situation leur demande du temps car la solution recherchée est plus compliquée à trouver.

La situation 2, qui ne sera abordée que par les élèves les plus avancés, est présentée sous forme d'expression littérale afin d'habituer les élèves à ce registre sémiotique pour petit à petit les mener vers les équations.

Analyse a posteriori

Les élèves reconnaissent rapidement le type de situation et voient rapidement le type de script qu'il faut programmer. Sur 20 groupes, 7 groupes ont programmé des scripts simples comme ci-dessous. 2 groupes ont réussi malgré tout à trouver la solution, mais n'ont pas eu le temps de passer à la situation 2.



10 groupes ont utilisé les listes et ont programmé des scripts comme ci-dessous. 7 groupes ont trouvé la solution.



3 groupes ont essayé de faire des scripts plus précis. 2 groupes ont alors trouvé la solution. Ils modifiaient le pas directement dans la zone de programmation :



4. Collège : Activité « L'aquarium de Bubulle et Didier » (Voir Annexe 4)

Analyse a priori

Contexte :

Cette activité est donnée après que les élèves ont vu en classe comment mettre un problème en équation et comment rechercher une solution à cette dernière à l'aide d'une calculatrice, du tableur ou du logiciel Scratch. Cette activité constitue donc un travail de fin d'apprentissage de la notion d'équation et de sa modélisation.

Choix didactiques :

L'objectif de cette activité n'étant pas l'étude du volume du pavé droit, le choix a été fait de donner les mesures en cm et les volumes en cm^3 . De même, un formulaire indiquant les volumes de différents solides a été donné aux élèves pendant l'activité, afin que le calcul de volume ne constitue pas à un obstacle. Les élèves ont la possibilité d'utiliser toutes les ressources dont ils disposent. Ils peuvent par conséquent utiliser le cahier d'exercices dans lequel des exercices avec Scratch ont été travaillés. Il leur suffit alors de recopier des modèles de scripts. De plus, ils peuvent réutiliser tous les travaux sur Scratch qui ont été réalisés avec eux au préalable.

Déroulement de l'activité :

Les élèves se mettent en groupes de 3- 4 élèves. L'activité est distribuée et lue en classe entière afin de bien expliquer tout ce qui est attendu. Les élèves doivent dans un premier temps rendre un document écrit dans lequel ils doivent expliquer leurs différentes idées, écrire l'équation correspondante et donner la réponse qu'ils ont trouvée grâce à leur script. Ce passage par l'écrit a pour but de bien faire réfléchir les élèves afin qu'ils n'utilisent l'ordinateur qu'une fois que le problème a bien été compris. Ils n'utilisent l'ordinateur que lorsqu'ils ont bien déterminé en quoi ce dernier pouvait leur être utile.

Aide pour les groupes ayant fini trop tôt :

Si certains groupes trouvent rapidement une réponse erronée, nous pouvons alors leur demander de calculer le volume du nouvel aquarium avec leur valeur pour voir si cette dernière est cohérente. Au contraire, si un groupe trouve rapidement une bonne réponse, nous pouvons alors demander de créer un script plus sophistiqué.

Attendus pour le script :

Cette activité étant réalisée après tout le travail réalisé sur les tests pour trouver des nombres, un script demandant un nombre à l'utilisateur puis réalisant un calcul qu'il affiche pour s'approcher du nombre recherché est au minimum attendu. Les élèves ayant accès aux différents scripts qu'ils avaient réalisés dans les travaux précédents, une amélioration de ce script, en utilisant des listes par exemple, peut être également attendue.

Evaluation par compétences :

Cette activité peut permettre d'évaluer des compétences du cycle 4. Un tableau d'indicateurs permet d'évaluer de façon plus précise :

N°	Domaine	Compétence	Insuffisant	Fragile	Satisfaisant	Très bien
1	D1.3	Mettre un problème simple en équation (Modéliser)				
2	D1.3	Écrire un algorithme ou un programme qui permet une interaction avec l'utilisateur ou entre les objets qu'il utilise, en réponse à un problème donné (Raisonnement)				
3	D2	Définir et respecter une organisation et un partage des tâches dans le cadre d'un travail de groupe (Chercher)				

N°	Indicateurs	Insuffisant	Fragile	Satisfaisant	Très bien
1	<input type="checkbox"/> Mise en place d'une inconnue <input type="checkbox"/> Utilisation de la relation entre la longueur et la hauteur <input type="checkbox"/> Multiplication de la longueur par la largeur par la hauteur <input type="checkbox"/> Ecriture d'une égalité	1 indicateur	2 indicateurs	3 indicateurs	4 indicateurs
2	<input type="checkbox"/> Le script a un lien avec la situation <input type="checkbox"/> Le script permet de rechercher le nombre cherché <input type="checkbox"/> Le script possède des listes ou des éléments permettant de faciliter la recherche du nombre cherché	0 indicateur	1 indicateurs	2 indicateurs	3 indicateurs
3	Les élèves trouvent leur place au sein du groupe et utilisent leurs compétences en commun pour aboutir à une production correcte	Aucune production correcte	Un travail d'équipe est mis en place mais n'aboutit pas	Un travail d'équipe est mis en place et une production est correcte	Le document écrit permet de comprendre le raisonnement <u>et</u> le script a du sens

Analyse a posteriori

Cette activité a été réalisée avec 6 groupes d'élèves. Le 1^{er} groupe n'a pas réussi à mettre le problème en équation et n'a pas réussi non plus à créer un script cohérent. Le 2nd groupe a mis en place des techniques pour créer l'équation mais ces dernières n'ont pas abouti et ce groupe n'a pas créé de script cohérent. Le 3^{ème} groupe n'a pas trouvé l'équation correspondante mais a réussi à créer un script cohérent :



Les trois autres groupes ont bien réussi à mettre le problème en équation et ont tous rédigé d'une façon analogue :

Eléments de recherche :

largeur : 20 cm hauteur : x longueur : $x + 30$ cm.

..... x désigne la hauteur du pavé

Mise en équation :

$20 \times x \times (x + 30) = 25\ 000 \text{ cm}^3$

Réponse à la question :

$x = 23,40573$

donc la hauteur de l'aquarium est de 23,40573
donc la longueur de l'aquarium est de 53,40573

Pour trouver la solution, le 4^{ème} groupe a créé un script avec une liste et a trouvé une solution proche de la valeur exacte :

The image shows a Scratch script on the left and a list of results on the right. The script starts with a 'when space is clicked' event, followed by 'ask for value of x and wait', 'delete all elements from the list hauteur', 'set x to response', 'ask for step and wait', and 'set pas to response'. It then enters a loop where it repeatedly adds the value of $x * x + 30 * 20$ to the 'hauteur' list, increments x by 'pas', and increments a counter from 1 to 11. The list on the right shows the values of 'hauteur' for each iteration, starting at 24837.8 and ending at 25686.45. A '+' sign at the bottom of the list indicates it can be expanded to show 'longueur: 11'.

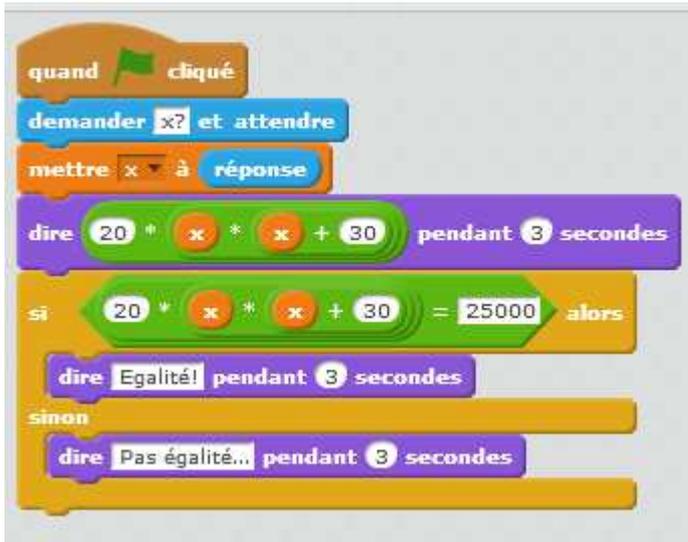
Les 5^{ème} et 6^{ème} groupes ont créé le même type de script et ont pu approcher une solution. Ce dernier est très intéressant et a permis à un groupe de se poser des questions. Ce script fait dire au lutin la réponse au calcul :

A Scratch 'say' block with the text '20 * x * x + 30' and a duration of 3 seconds.

Puis il fait vérifier si le nombre choisi est bien solution de l'équation :

A Scratch 'if' block with the condition '20 * x * x + 30 = 25000' and the label 'alors'.

Or, les élèves ont pu observer une limite du logiciel. En effet, le lutin indique une valeur arrondie quand il « dit » le résultat du calcul. Mais quand il fait le calcul, il doit le faire avec la valeur exacte, donc le lutin indiquait que le nombre choisi (23,40573 n'était pas solution alors qu'il affirmait finalement le contraire juste avant) :



III. Analyse des différents travaux menés en LGT

1. LGT : Activité « L'aquarium » (Voir Annexe 5)

Analyse a priori

Objectif :

Le but est d'amener les élèves à mettre un problème en équation et de les confronter à une équation qu'ils ne savent pas résoudre.

L'équation est du second degré, il n'y a pas de factorisation simple ni de racine entière.

$$20x(x + 30) = 25\ 000 \Leftrightarrow 20x^2 + 600x = 25\ 000$$

Equation du second degré insoluble en seconde, et avec une solution négative.

$$x_0 \simeq 23,41\ cm$$

Le travail de familiarisation avec la calculatrice graphique a été mené en début d'année et après quelques tentatives infructueuses de résolution, certains élèves pourraient prendre l'initiative « d'étudier à la calculatrice » la fonction $x \mapsto 20x^2 + 600x$, soit par la représentation graphique, soit par les tables de valeurs.

Analyse a posteriori

Les élèves ont travaillé par groupe de 4. Dans presque tous les groupes, les élèves ont cherché à mettre en équation d'eux-mêmes et ont abouti assez rapidement à l'équation voulue avec plus ou moins d'aide.

Ils ont cherché à résoudre, certains ont même cru trouver une solution, mais ils se sont vite sentis démunis. Deux ou trois élèves ont réalisé d'eux-mêmes des tableaux de valeurs de $20x + 600x^2$ et ont obtenu une valeur approchée au dixième.

2. LGT : TP informatique 1 « Résoudre une équation avec Python » (Voir Annexe 6)

Analyse a priori

Objectif :

Le but est maintenant d'obtenir des valeurs approchées d'une solution x_0 de l'équation :

$$20x^2 + 600x = 25\ 000$$

en utilisant le langage Python.

Les boucles bornées et la notion de fonction ont déjà été abordées. Ils vont découvrir la boucle conditionnelle à cette occasion.

Les élèves sont d'abord incités à une recherche naïve de solution, puis à utiliser une méthode de balayage et enfin à cibler le résultat avec la boucle conditionnelle. Cette dernière méthode offre une réelle amélioration par rapport aux tables de valeurs de la calculatrice.

Afin de palier les difficultés de syntaxe (puissance, implémentations pour les fonctions et les boucles), les élèves partent à chaque étape d'un exemple analogue.

La question 4, plus abstraite et réservée aux élèves les plus rapides, leur demande de créer une fonction qui renvoie un résultat et les libère de l'instruction « print »

Analyse a posteriori

Le professeur a été évidemment très sollicité pour régler les problèmes de syntaxe.

Dans la partie 2, quelques élèves ont fait remarquer que le tableau de valeurs obtenu n'offrait pas d'avantage par rapport à la calculatrice. C'est donc la boucle conditionnelle puis la question 4 qui leur montrera l'intérêt de la programmation.

3. LGT : Jeux informatiques « Un monstre et un chat » et « Devine un nombre » (Voir Annexe 7)

Analyse a priori

Objectif :

Le but est de préparer l'élève à la méthode de dichotomie à partir de deux exercices ludiques.

- Le premier est un exercice en ligne tiré d'une épreuve du concours Castor Informatique. Pour coincer un monstre avec un minimum de barrages, l'élève doit **couper en deux** la zone dans laquelle se trouve le monstre.

Attraper le monstre

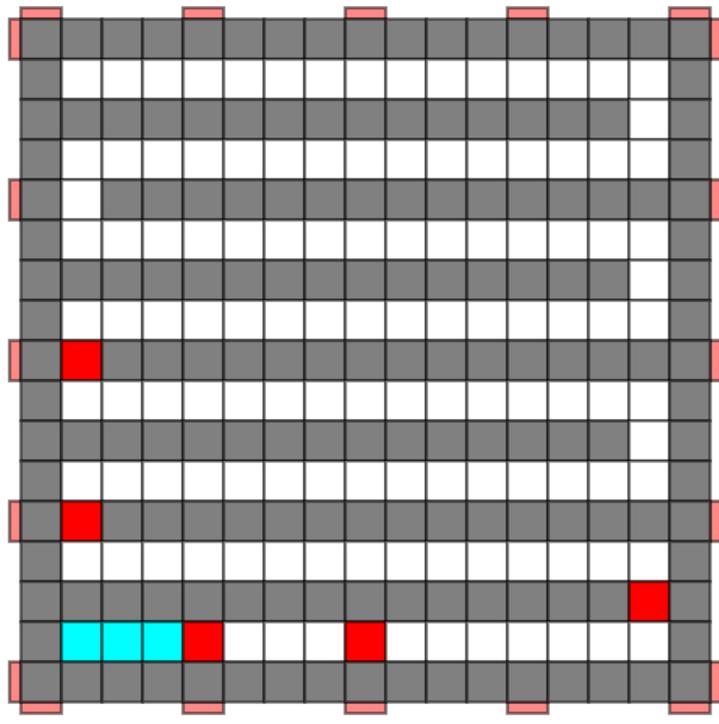
Un monstre vient de s'introduire dans les douves du donjon de Castor !



Votre objectif est de coincer le monstre pour qu'il ne puisse plus bouger, en utilisant un **minimum** de barrages.

Les cases bleues montrent à chaque fois la zone où se cache le monstre.

Nombre de barrages utilisés : 5.



- Le deuxième exercice demande de trouver un entier entre 1 et 100 (puis entre 1 et 1000) avec le moins d'essais possibles.

Scratch génère un entier aléatoire et le lutin répond « plus grand », « plus petit » et finalement « gagné » avec le nombre d'essais effectués. L'élève ne programme pas le script, il se contente de jouer.

r 50
score 0



Plus petit.
Nouvelle tentative

```

quand est cliqué
mettre n à nombre aléatoire entre 1 et 100
demander Devine le nombre entier que j'ai choisi (entre 1 et 100) et attendre
mettre r à réponse
mettre score à 0
répéter jusqu'à r = n
si réponse < n alors
demander Plus grand. Nouvelle tentative et attendre
mettre r à réponse
sinon
demander Plus petit. Nouvelle tentative et attendre
mettre r à réponse
mettre score à score + 1
dire Gagné ! En pendant 1 secondes
dire score pendant 1 secondes
dire tentatives pendant 2 secondes
  
```

Il est demandé de noter la liste des intervalles dans lesquels se situe le nombre cherché et de faciliter ainsi le passage du « on coupe l'intervalle en deux » au calcul de $(a+b)/2$.

Si la méthode utilisée est optimale (dichotomie), le nombre est trouvé en 10 tentatives maximum puisque $2^{10} = 1024$.

Exemple de résolution optimale avec 736 :

Nombre testé	plus grand (+) ou plus petit (-) ?	intervalle de recherche
500	+	[500 ; 1000]
750	-	[500 ; 750]
625	+	[625 ; 750]
688	+	[688 ; 750]
719	+	[719 ; 750]
735	+	[735 ; 750]
743	-	[735 ; 743]
739	-	[735 ; 739]
737	-	[735 ; 737]
736	gagné	

Analyse a posteriori

Le premier exercice a été assez rapidement réussi par la grande majorité des élèves.

Le passage au domaine numérique dans le deuxième exercice s'avère plus laborieux. Un petit nombre d'élèves finit par calculer spontanément les centres des intervalles de recherche mais beaucoup coupent les intervalles en deux au jugé. Le calcul de la moyenne $(a+b)/2$ a été amené suite à des interrogations orales.

4. LGT : TP informatique 2 : « Un premier algorithme de dichotomie » (Voir Annexe 8)

Analyse a priori

Objectif :

L'idée est d'amener l'élève à faire lui-même l'expérience de la méthode afin de l'assimiler. Le choix a été fait de coller à l'équation $V(x) = 25\,000$ et donc de comparer $V(x)$ à 25 000 plutôt que d'étudier le signe de $V(x) - 25\,000$, a priori moins parlant.

Cette activité a été faite dans les jours qui suivaient « Un monstre et un chat » pour faciliter le calcul de m .

Analyse a posteriori

Le remplissage du tableau n'a pas posé de problème majeur à l'exception de la dernière colonne, dont la compréhension est essentielle avant de programmer.

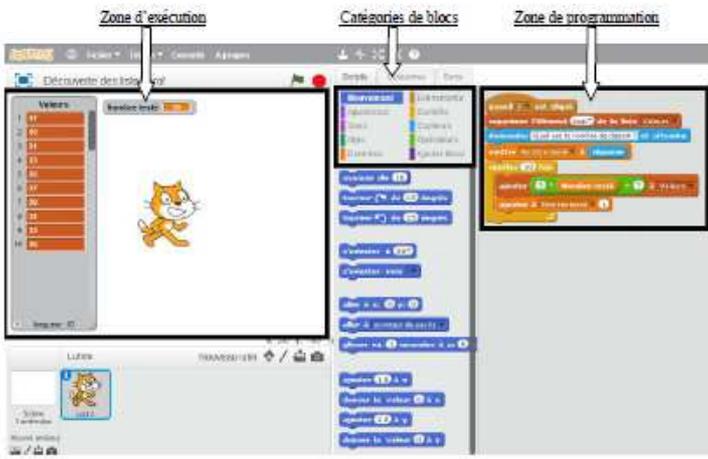
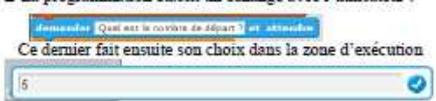
Bien que l'essentiel de la fonction dichotomie ait été rédigé, la condition « Tant que $b - a < p$ » a donné du fil à retordre aux élèves qui ont eu du mal à comprendre le rôle de la variable p .

IV. Analyse des différents travaux menés en LP

5. LP : Activité « L’emballage » (Voir Annexe 9)

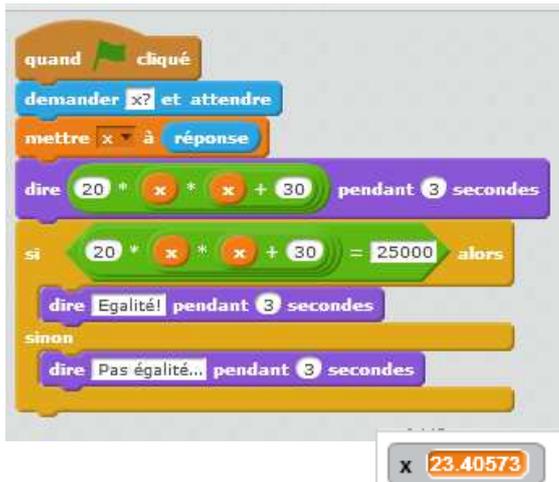
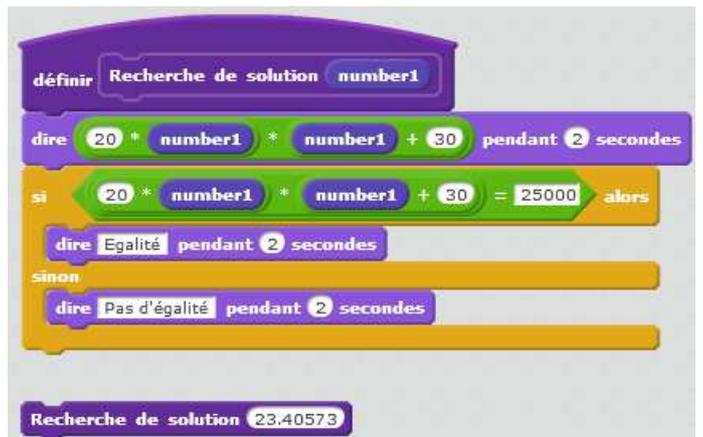
V. Bilan : passage du logiciel Scratch au logiciel Python

Voici les différences que nous avons pu établir entre les deux logiciels :

<u>Du langage Scratch</u>	<u>au langage Python</u>
	
<ul style="list-style-type: none"> La zone d'exécution se trouve à gauche de l'écran et peut avoir un aspect ludique La programmation se fait à l'aide d'étiquettes :  La programmation établit un échange avec l'utilisateur :  	<ul style="list-style-type: none"> La zone d'exécution se trouve à droite de l'écran et signale les erreurs d'écriture de programmation La programmation se fait à l'aide de phrases : <code>print(2*1+7)</code> La programmation établit un échange avec la machine par la définition d'une variable informatique : <code>def Test(deb):</code> La variable est ensuite choisie dans la zone d'exécution : <code>> Test(6)</code>

Passage de Scratch à Python au collège :

Pour habituer les élèves avec Scratch à ne plus s'adresser à un utilisateur mais à l'ordinateur, comme ils devront le faire avec Python, il est possible d'inciter ces derniers à ne plus utiliser l'étiquette « demander » et à préférer la création d'un bloc avec une entrée. Ils s'habituent ainsi à la création d'une fonction avec une variable.

	
--	--

Passage de Scratch à Python au lycée :

Une difficulté que rencontrent les élèves avec le logiciel Scratch est la programmation d'une succession de calculs. Multiplier l'emboîtement d'étiquettes les unes dans les autres peut rapidement devenir difficile. Ce problème n'est pas rencontré avec Python. Ce constat peut servir de point d'appui pour justifier l'utilisation d'un nouveau logiciel de programmation en classe de seconde.

VI. Annexes : énoncés des différents travaux

1. Collège : TP informatique 1 « Programmer pour trouver des solutions »

Programmer pour trouver des solutions

Créer un programme pour chaque situation donnée qui permette de trouver la ou les solutions attendues.

Situation 1 :

Antoine choisit un nombre. Il le multiplie par 3,5 puis il ajoute 7 au produit obtenu. Au final, il obtient -35 .
Quel nombre a-t-il choisi au départ ?

Solution :

.....

Situation 2 :

Pauline choisit un nombre. Elle le multiplie par $-6,8$ puis elle soustrait 15 au produit obtenu.
Au final, elle obtient $-33,36$. Quel nombre a-t-elle choisi au départ ?

Solution :

.....

Situation 3 :

Pauline et Antoine choisissent tous les deux le même nombre au départ.

Pauline multiplie ce nombre par 4 et ajoute 3 au résultat obtenu.

Antoine multiplie le nombre par -3 et ajoute 53,4 au résultat obtenu.

Quand ils ont terminé, ils s'aperçoivent qu'ils obtiennent exactement le même résultat à la fin.

Quel nombre ont-ils choisi au départ ?

Solution :

.....

Améliorations possibles :

Indiquer les avantages et les inconvénients de chacun des programmes réalisés et indiquer les améliorations possibles qui permettraient de créer des programmes plus performants pour répondre à ce genre de question :

.....

.....

.....

.....

.....

Situation 4 :

Programme de calculs A

■ Choisir un nombre

■ Multiplier par 7

■ Ajouter 15

Programme de calculs B

■ Choisir un nombre

■ Multiplier par -9

■ Ajouter 1 635,8

Pauline a choisi le même nombre au départ pour les deux programmes et a trouvé à la fin un même nombre.

Quel nombre a-t-elle choisi au départ ?

Solution :

.....

Découverte des listes

I Tester une expression pour les nombres entiers de 1 à 10

Créer un programme exécuté « quand a est cliqué ».

Le lutin donne la valeur de l'expression $2x + 7$ quand $x = 1 ; x = 2 ; x = 3 ; x = 4 ; x = 5 ; x = 6 ; x = 7 ; x = 8 ; x = 9$ et $x = 10$ et stocke ces valeurs dans une liste.

II Tester une expression pour dix nombres entiers consécutifs au choix

Créer un programme exécuté « quand b est cliqué ». Le lutin donne la valeur de l'expression $2x + 7$ pour dix nombres entiers consécutifs choisis par l'utilisateur, et stocke ces valeurs dans une liste.

Aide : Pour comprendre le principe de la question réponse voir le tutoriel sur le lien suivant :

<http://mcetechno.ovh/epi.html>



Aide : Pour comprendre le principe du stockage d'une donnée dans une variable voir le tutoriel sur le lien suivant :

<http://mcetechno.ovh/epi.html>



III Tester une expression pour dix nombres au choix

Créer un programme exécuté « quand c est cliqué ». Le lutin donne la valeur de l'expression $2x + 7$ pour dix nombres choisis par l'utilisateur. L'utilisateur doit indiquer le premier nombre qu'il veut tester et le pas* régulier entre les dix valeurs qu'il désire. Les dix valeurs doivent être stockées dans une liste.

Exemples de pas* : 1 ; 3 ; 5 ; 7 ; 9 ; 11 ; 13 le pas entre les valeurs est ici de 2.

9 ; 9,5 ; 10 ; 10,5 ; 11 ; 11,5 ; 12 le pas entre les valeurs est ici de 0,5.

IV Tester une expression pour plusieurs nombres au choix

Créer un programme exécuté « quand d est cliqué ». Le lutin donne la valeur de l'expression $2x + 7$ pour plusieurs nombres choisis par l'utilisateur. L'utilisateur doit indiquer le premier nombre qu'il veut tester et le pas* régulier entre les valeurs qu'il désire et indique le nombre de valeurs qu'il veut tester. Toutes les valeurs doivent être stockées dans une liste.

3. Collège : TP informatique 3 « Utilisation des scripts précédents »

Utilisation des scripts précédents

Créer un programme pour chaque situation donnée qui permette de trouver la ou les solutions attendues.

Situation 1 :

Programme de calculs A

- Choisir un nombre
- Multiplier par 7
- Ajouter 15

Programme de calculs B

- Choisir un nombre
- Multiplier par - 9
- Ajouter 2 138,536

Pauline a choisi le même nombre au départ pour les deux programmes et a trouvé à la fin un même nombre. Quel nombre a-t-elle choisi au départ ?

Solution :

.....

Situation 2 :

Considérons l'expression littérale $A = x^2 + x - 2$.

Pour quelle(s) valeur(s) de x , cette expression est-elle égale à 4 ? :

Solution :

.....

4. Collège : Activité « L'aquarium de Bubulle et Didier »

Activité : l'aquarium de Bubulle et Didier

Bubulle et Didier ont grandi. Ils se sentent maintenant à l'étroit dans leur aquarium en pavé droit de 15 L. Aïcha l'a compris et souhaite leur en construire un nouveau d'une capacité de 25 L (soit $25\,000\text{ cm}^3$). Comme ses étagères sont étroites, la base de l'aquarium doit être un rectangle de 20 cm de large. Aïcha voudrait aussi qu'il fasse 30 cm de plus en longueur qu'en hauteur.

Quelle doit être la hauteur de l'aquarium ?



- 1) Mettre en équation cet énoncé.
- 2) Modéliser ce problème à l'aide du logiciel Scratch pour trouver la réponse.

NOMS Prénoms :

Feuille réponse

N°	Domaine	Compétence	Insuffisant	Fragile	Satisfaisant	Très bien
1	D1.3	Mettre un problème simple en équation (Raisonner)				
2	D1.3	Écrire un algorithme ou un programme qui permet une interaction avec l'utilisateur ou entre les objets qu'il utilise, en réponse à un problème donné (Modéliser)				
3	D2	Définir et respecter une organisation et un partage des tâches dans le cadre d'un travail de groupe (Chercher)				

Eléments de recherche :

.....
.....
.....
.....
.....
.....

Mise en équation :

.....
.....
.....
.....
.....
.....

Nom de la personne qui a ouvert sa session :

.....

Réponse à la question :

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

5. LGT : Activité « L'aquarium »

L'aquarium

Bubulle et Didier ont grandi. Ils se sentent maintenant à l'étroit dans leur aquarium de 15 L. Aïcha l'a compris et souhaite en construire un nouveau d'une capacité de 25 L (soit $25\,000\text{ cm}^3$).

Comme ses étagères sont étroites, la base de l'aquarium doit être un rectangle de 20 cm de large. Aïcha voudrait aussi qu'il fasse 30 cm de plus en longueur qu'en hauteur. Quelle doit être la hauteur de l'aquarium ?



6. LGT : TP informatique 1 « Résoudre une équation avec Python »

Résoudre une équation avec Python

Objectif : Trouver **une solution** x_0 de l'équation $20x^2 + 600x = 25\,000$ pour résoudre le problème de l'aquarium.

```
x=3
```

```
c=x**2
```

```
print(c)
```

1. a) Dans l'éditeur, reproduire le programme ci-contre.

L'exécuter plusieurs fois en changeant la valeur de départ de x et dire ce qu'il fait.

b) Avec une fonction.

Créer une fonction nommée c : - dont l'argument est un nombre x.
- qui renvoie le carré de x.

Tester cette fonction dans la console.

Créer une fonction nommée V : - dont l'argument est un nombre x.

- qui renvoie le volume de l'aquarium en fonction de x.

En faisant plusieurs tests, donner **deux nombres entiers** encadrant la solution x_0 cherchée.

c) Soyons plus précis, donner une valeur approchée de x_0 à 10^{-1} près.

2. Balayages.

a) A la suite de ce qui a été fait dans la partie 1, reproduire le programme ci-contre.
L'exécuter et dire ce qu'il fait.

```
x=3
```

```
for i in range(5):
```

```
    print(x,c(x))
```

```
    x=x+2
```

On parle de « balayage » : pour la variable x, la valeur de début est et le pas est

b) En utilisant un autre balayage avec la fonction V, vérifier les résultats obtenus en 1.c).

valeur de début : pas :

c) Soyons plus précis, donner une valeur approchée de x_0 à 10^{-2} près.

..... valeur de début : pas :

3. Des balayages qui s'arrêtent.

a) A la suite de ce qui a été fait dans la partie 1, reproduire le programme ci-contre.
L'exécuter et dire ce qu'il fait.

```
x=0
```

```
car=c(x)
```

```
while car<150:
```

```
    x=x+1
```

```
    car=c(x)
```

```
    print(x,car)
```

b) En s'inspirant de la question a), donner une valeur approchée de x_0 à 10^{-3} près.

4. Définir une fonction avec pour arguments :

- la valeur de départ de x : d.
- le pas : p

... et qui renvoie :

- une valeur approchée de x_0
- le volume correspondant à cette valeur approchée.

8. LGT : TP informatique 2 « Un premier algorithme de dichotomie »

Résolution approchée de $V(x) = 25\,000$ (à 10^{-4} près)

On veut déterminer une valeur approchée de la solution cherchée x_0 par un procédé plus ingénieux que le balayage :

$V(0) = 0$ et comme $V(50) = 20 \times 50 \times 80 = 80\,000$,

On peut en déduire que x_0 est compris entre 0 et 50.

Information peu précise ! x_0 appartient à $[0 ; 50]$, intervalle **d'amplitude 50**.

Etape 1. Alors on coupe $[0 ; 50]$ en deux intervalles de même amplitude : $[0 ; 25]$ et $[25 ; 50]$.

Puis on détermine de quel « côté » se trouve x_0 en calculant :

$$V(25) = \dots\dots\dots$$

Comme V est croissante sur $[0 ; 50]$, on peut comparer x_0 à 25 : $x_0 \dots\dots 25$

Information plus précise : x_0 appartient à, intervalle de **d'amplitude**

Etape 2. Alors on coupe en deux intervalles de même amplitude : et

Et ainsi de suite.

On peut ainsi compléter le tableau ci-dessous :

	Intervalle $[a ; b]$	Amplitude	Centre m de $[a ; b]$	$V(m) \simeq \dots$	Donc m prend le rôle de...
Etape 1	$[0 ; 50]$	50	25		
Etape 2					
Etape 3					
Etape 4					
Etape 5					

Fastidieux ! A Python d'exécuter cet algorithme pour nous, en créant une fonction nommée « **dicho** » :

- l'argument **p** de **dicho** correspondra à la **précision souhaitée** (c'est-à-dire que l'amplitude du dernier intervalle $[a ; b]$ doit être plus petite que **p**)
- **dicho** doit renvoyer la dernière valeur de m .

<p>Fonction dicho(p) :</p> <p>$a \leftarrow 0$ $b \leftarrow 50$</p> <p>Tant que $b - a \dots\dots\dots$</p> <p style="padding-left: 2em;">$m \leftarrow \dots\dots\dots$</p> <p style="padding-left: 2em;">Si $V(m) \dots\dots\dots$ alors $a \leftarrow m$</p> <p style="padding-left: 2em;">Sinon $b \leftarrow m$</p> <p style="padding-left: 2em;">FinSi</p> <p>Renvoyer m</p> <p>Fin Fonction</p>	<pre>def dicho(p): a=0 b=50</pre>
--	---

Corrigé

	Intervalle $[a ; b]$	Amplitude	Centre m de $[a ; b]$	$f(m) \approx \dots$	Donc m prend le rôle de...
Etape 1	$[0 ; 50]$	50	25	27 500	b
Etape 2	$[0 ; 25]$	25	12,5	10 625	a
Etape 3	$[12,5 ; 25]$	12,5	18,75	18 281	a
Etape 4	$[18,75 ; 25]$	0,625	21,875	22 695	a
Etape 5	$[21,875 ; 25]$	0,3125	23,4375	25 049	b

Fonction dichotomie :

```

 $a \leftarrow 0$ 
 $b \leftarrow 50$ 
Tant que  $b - a > p$ 
     $m \leftarrow (a + b)/2$ 
    Si  $V(m) < 25\,000$ 
        alors  $a \leftarrow m$ 
    Sinon  $b \leftarrow m$ 
    FinSi
Renvoyer  $m$ 
Fin Fonction
    
```

def dichotomie:

```

a=0
b=50
while b-a>p:
    m=(a+b)/2
    if V(m)<25 000:
        a=m
    else:
        b=m
return m
    
```

9. LP : Activité « L’emballage »

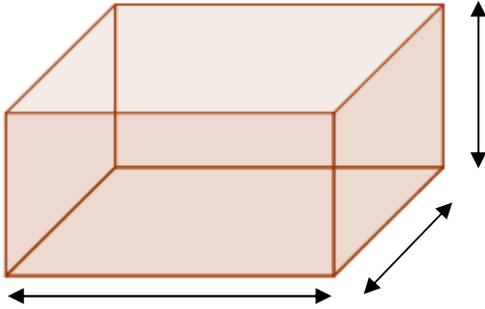
L’emballage

Une usine d’emballage reçoit la commande d’une entreprise pour la conception d’un contenant de 25 L, soit $25\,000\text{ cm}^3$, avec les contraintes suivantes :

Une profondeur de 20 cm, une hauteur non définie mais une largeur de 30 cm de plus que la hauteur.

Problématique : Quelle doit-être la hauteur de l’emballage afin d’obtenir un volume de $25\,000\text{ cm}^3$?

1. À l’aide d’une inconnue, inscrire les côtes sur le schéma suivant. (CHERCHER)



1. Vérifier que le volume de la boîte, pour une hauteur de 10 cm, est de $8\,000\text{ cm}^3$. (CHERCHER)

$$V = \text{largeur} \times \text{profondeur} \times \text{hauteur}$$

2. Exprimer le volume de la boîte en fonction de x . (MODÉLISER)

$$V(x) = \dots\dots\dots$$

3. Proposer une méthode, permettant de déterminer la hauteur nécessaire. (RAISONNER/COMMUNIQUER)

.....

.....

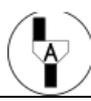
.....

.....

.....

.....

.....

 **Appel professeur** : proposer votre protocole afin de le réaliser.

Travail réalisé par :

ACHAINTRE Olivier (Professeur au collège)

BLANZAT Walter (Professeur au lycée général et technologique)

DESCOURS Christophe (Professeur au lycée professionnel)