

# Informatique et Sciences du Numérique (ISN), ou comment se lancer dans une aventure nouvelle

Frédéric Laroche(\*)

Ayant fait beaucoup d'informatique dans les années 1980-2000, particulièrement avec divers programmes réalisés (et même commercialisés pour certains) en Turbo-Pascal, ayant également pris en charge pendant dix ans les TD d'informatique en prépa, je me suis lancé dans l'aventure ISN à deux ans de la retraite... Je vais présenter succinctement mes débuts dans cet enseignement, sachant que rien n'est stabilisé et que des évolutions vont probablement se produire rapidement.

## 1. Ma situation d'enseignant de spécialité ISN.

J'ai deux groupes : 16 et 25 élèves, tous d'origine S/SVT, sur deux établissements comportant respectivement 120 et 200 élèves de TS ; le groupe de 25 est un peu chargé et me pose quelques problèmes inattendus... Pour les salles je dispose d'une salle avec 20 postes (groupe de 16) et d'une salle avec 35 postes (groupe de 25). Les élèves peuvent venir avec leurs portables, ils disposent du wifi dans les deux salles. Le recrutement s'est fait en mai 2012 : je suis passé dans toutes les Première S pour expliquer la teneur de cette spécialité ; il ne semble pas y avoir de « geek<sup>(1)</sup> » dans les groupes mais 4 élèves au total ont déjà programmé en Java ou en C.

La communication se fait par l'intermédiaire de deux comptes gmail ouverts au nom de chaque groupe, ce qui fait que tout le monde a accès à toutes les informations : c'est pratique mais pas très sécurisé.

## 2. Le programme

« L'objectif de l'enseignement de spécialité ISN en classe terminale de la série S n'est pas de former des experts en informatique, mais plutôt de fournir aux élèves quelques notions fondamentales et de les sensibiliser aux questions de société induites. Il s'agit d'un enseignement d'ouverture et de découverte des problématiques actuelles, adapté à la société d'aujourd'hui, qui valorise la créativité et contribue à l'orientation. »

Le programme tel qu'il a été écrit essentiellement à partir des idées des informaticiens de l'INRIA (G. Dowek, T. Vieville, J.-P. Archambault, G. Berry, S. Abiteboul, C. de la Higuera, etc.) se décompose en quatre grandes parties :

---

(\*) frederic.laroche0@gmail.com

<http://laroche.lycee.free.fr>, <http://promenadesmaths.free.fr/>

(1) Geek : quelqu'un qui passe ses journées devant son ordinateur à « bidouiller », généralement très habile à trouver puis à utiliser tout outil et toute astuce sur le réseau.

Représentation de l'information	Représentation binaire, Opérations booléennes, Numérisation, Formats, ...
Algorithmique	Algorithmes simples : rechercher un élément dans un tableau trié par une méthode dichotomique ; trier un tableau par sélection ; ajouter deux entiers exprimés en binaire. Algorithmes plus avancés : tri par fusion ; recherche d'un chemin dans un graphe par un parcours en profondeur (DFS) ; recherche d'un plus court chemin par un parcours en largeur (BFS).
Langages et programmation	Types de données, Fonctions, Correction d'un programme, Langages de description (HTML)
Architectures matérielles	Éléments d'architecture, Jeu d'instructions. Réseaux : Transmission point à point, Adressage, Routage...
On peut rajouter à titre optionnel une initiation à la robotique.	

Un deuxième objectif consiste à faire prendre conscience des enjeux de la numérisation accélérée de nombreux pans de la société : sur la base d'exposés d'élèves un dialogue doit s'installer permettant de cibler quelques problématiques actuelles. La propriété intellectuelle, le « cloud computing<sup>(2)</sup> », le « fichage » des populations, les Anonymous<sup>(3)</sup>, les bugs et les tests, etc. sont des sujets actuels qui permettent des discussions animées ainsi que des prises de conscience des enjeux de l'informatique.

Par ailleurs, au cœur du dispositif se trouve la mise en activité de l'élève : « *Afin de refléter le caractère scientifique et technique propre à la discipline et de développer l'appétence des élèves en faveur de cet enseignement nouveau pour eux, il convient de les mettre en situation d'activité aussi souvent que possible. Une pédagogie de projet est à privilégier pour favoriser l'émergence d'une dynamique de groupe. Dans ce cadre, le professeur joue un rôle central : il impulse et coordonne les projets, anime les débats et met en place l'évaluation et ses modalités.* »

Quelques détails : « *Les projets réalisés par les élèves, sous la conduite du professeur, constituent un apprentissage fondamental tant pour la compréhension de l'ISN que pour l'acquisition de compétences variées. Ils peuvent porter sur des problématiques issues d'autres disciplines et ont essentiellement pour but d'imaginer des solutions répondant à l'expression d'un besoin.* »

Bien, bien, bien... on a environ 50 à 60 heures dans l'année, la grande majorité des élèves n'a pas touché un ordinateur autrement qu'à travers un jeu, un traitement de

(2) Cloud computing : consiste à utiliser des logiciels et des bases de données installées, non sur votre ordinateur ou votre réseau, mais « quelque part » sur le web. Mettre, par exemple, vos photos personnelles sur un site ouvert à vos amis, ce qui vous épargne de les leur envoyer, c'est déjà faire du cloud computing.

(3) Nom générique de collectifs informels d'internautes généralement rebelles aux tentatives de contrôle du réseau et aux pouvoirs en place. Leur nom correspond à leur type d'action, toujours anonyme, ce qui peut malheureusement permettre diverses dérives et manipulations.

texte, un tableur ou un logiciel de messagerie et souvent ils ont choisi ISN car ils n'avaient pas trop envie de faire des maths... Le prof assis devant sa vingtaine d'élèves se demande bien comment gérer des profils aussi disparates et les amener au niveau souhaité.

La lecture attentive du programme laisse rêveur et les informaticiens que j'ai consultés ont tous remarqué qu'il s'agissait grosso-modo du programme de licence qu'ils avaient absorbé pendant leurs études... (consulter par exemple [http://www.univ-paris8.fr/IMG/pdf\\_Org.\\_ens.\\_licence\\_Informatique.pdf](http://www.univ-paris8.fr/IMG/pdf_Org._ens._licence_Informatique.pdf)).

Un certain nombre de collègues intéressés par l'enseignement d'ISN ont également pris peur à la lecture de ce programme et n'ont pas donné suite, particulièrement en SI. Il est clair que l'assimilation des connaissances listées dans le programme est incompatible avec la mise en activité permanente de l'élève : il faudra faire des choix, mais lesquels ?

### 3. L'évaluation

Une des grandes nouveautés d'ISN est l'évaluation finale en interne : la note d'ISN au baccalauréat (coefficient 2) sera donnée par une commission constituée de l'enseignant responsable de l'élève et d'un autre collègue si possible. Une harmonisation des notes sera réalisée au niveau académique comme pour les TPE.

L'évaluation d'une durée de 20 minutes se fera en deux temps :

*Première partie* - Évaluation d'un projet et soutenance orale, durée : 8 minutes maximum, notée sur 8 points

Le candidat effectue une présentation orale de son projet, pendant laquelle il n'est pas interrompu. Il s'appuie pour cela sur un dossier-projet de 5 à 10 pages, hors annexes, élaboré à l'aide de l'outil informatique.

Ce projet est structuré de façon à mettre en évidence plusieurs points : Le but visé et les moyens choisis pour atteindre ce but. La démarche de projet qui a conduit au résultat tel que présenté. La dimension collaborative du projet liée au travail en équipe (2 à 3 élèves).

[...]

*Deuxième partie* - Dialogue argumenté avec la commission d'évaluation, durée : 12 minutes minimum, 12 points

La commission d'évaluation interroge le candidat sur différents aspects de son projet et sur son lien avec les compétences fixées par le programme, puis élargit ce questionnement aux autres compétences spécifiées dans le programme. Cette interrogation a notamment pour but de vérifier que le candidat s'est approprié les notions fondamentales, sait les utiliser dans un contexte particulier ou les relier aux autres enseignements scientifiques spécifiques de la série.

La grille de compétences d'où est extrait cet encadré est très chargée. Elle comprend 11 (+2) rubriques distinctes donnant lieu à évaluation sur 1, 2 ou 3 points pour chaque partie de l'évaluation (au total 16 items à évaluer). Le choix des éléments

d'appréciation restant à la charge de la commission d'examen.

Sans vouloir faire de mauvais esprit, il risque d'être un peu délicat de gérer dans le même temps l'écoute attentive de l'élève, préparer les questions à lui poser, remplir la grille de compétences, voire commenter le travail fourni pour justifier de la note mise ... surtout si on est tout seul... On peut envisager (craindre ?) que dans la deuxième partie un petit exercice de programmation ou d'algorithmique soit posé à l'élève afin de vérifier *a minima* son assimilation des éléments du programme.

Ceci renvoie d'ailleurs à une interrogation importante pour les élèves : jusqu'à nouvel ordre les bulletins trimestriels, les dossiers pour Admission PostBac (APB) et les livrets scolaires comprennent les rubriques Notes et Appréciations ; quelles notes mettre en cours d'année et quelle note mettre pour APB au mois d'avril, avant l'épreuve terminale ? Quels critères d'évaluation doit-on prendre ? N'y a-t-il pas une injustice profonde à ne pas mettre les élèves sur un pied d'égalité en face de l'évaluation ? Autant l'évaluation finale se fera probablement sans trop de heurts malgré les difficultés soulevées, autant l'évaluation en cours d'année va forcément pâtir des attentes diverses sur les capacités exigibles des élèves : entre des enseignants issus de disciplines différentes les exigibles seront forcément très divers ; par ailleurs fera-t-on un distinguo entre des élèves grands débutants et des experts du C, devra-t-on pénaliser un élève incapable de travailler en équipe, ... ; la procédure APB est déjà génératrice d'inégalités fortes, on ne voit pas bien l'intérêt d'en rajouter. La question de disposer d'un référentiel d'évaluation pour APB a été posée à L. Chéno mais aucune réponse acceptable n'a été donnée pour l'instant.

Le type d'évaluation par compétences ici proposée peut sembler pertinent dans le cadre du baccalauréat, quoique toutes les spécialités ne soient pas logées à la même enseigne. Mais il va forcément être biaisé dans la mesure où l'évaluation est effectuée par l'enseignant de l'élève : il va être impossible de faire abstraction des informations déjà disponibles sur le travail, l'investissement, la qualité des projets, etc. et l'évaluation risque de davantage ressembler à un contrôle en cours de formation (CCF) qu'à une évaluation finale. Ce n'est pas que ce soit vraiment gênant mais ça pose la question de l'égalité des élèves devant l'épreuve : les spécialistes des autres disciplines, mathématiciens ou autres, risquent d'être un peu contrariés par ce traitement inégalitaire. Par ailleurs on risque de voir venir vers ISN des élèves peu motivés, prenant alors la place des élèves intéressés par la discipline : comment faire le tri en fin de Première ? On voit que de nombreuses questions restent posées et nécessiteront probablement une remise à plat des modalités d'évaluation assez rapidement.

#### 4. Et si on programmait ?

Reprenons maintenant sur le fond : des contenus très vastes, des attendus en grande partie inatteignables, une évaluation à moitié déficiente, ... on va faire quoi ? Eh bien on va programmer : l'informatique c'est des programmes qui fonctionnent sur des ordinateurs, des tablettes, des téléphones, dans des robots, des processeurs embarqués. ISN n'est pas une spécialité d'électronique, donc sauf s'il s'agit de s'intéresser à l'architecture matérielle des machines et à leur mode de

fonctionnement, l'essentiel des projets va passer par la mise au point d'applications.

Mais pour réaliser des applications il faut un compilateur<sup>(4)</sup> ou un interpréteur, supports d'un langage spécifique ; se posent donc deux questions : quel système d'exploitation choisir (OS) ? quel langage choisir (et accessoirement quel logiciel support ou éditeur) ?

Pour répondre à la question de l'OS le contexte local doit aider : la plupart des réseaux des lycées étant sous Windows il suffira de demander à l'administrateur d'installer les divers logiciels nécessaires en début d'année. Si on ne veut pas dépendre de l'administrateur on pourra utiliser une distribution Linux disponible sur clé USB et comprenant un certain nombre d'applications préinstallées<sup>(5)</sup> : cette clé devrait être offerte par les rectorats...

Deuxième étape : le langage... Comme il en existe environ 30 000 on peut se dire qu'on a le choix... Dans la pratique il devra répondre à quelques critères :

- Être rapidement et facilement utilisable par les élèves, aussi bien en classe que chez eux.
- Disposer d'éditeurs gratuits et efficaces.
- Être suffisamment généraliste pour permettre de passer à un autre langage facilement.
- Pour une portabilité maximale être orienté objet (ce qui n'est pas le cas de langages comme PERL ou les langages propriétaires type Maple).
- Disposer de ressources nombreuses et faciles d'accès...
- ...

Un petit tour sur des sites comme le Site du Zéro (<http://www.siteduzero.com>) ou Developpez.com (<http://www.developpez.com>) montre les quelques solutions : C# (un peu délicat pour démarrer), Java, Python (ou Ruby). Pour les projets web on choisira entre PHP si on travaille directement sur le serveur web<sup>(6)</sup> ou JavaScript si on travaille dans le navigateur.

L'éditeur devra fournir les outils suivants :

- Complétion de code : la saisie des mots du langage est complétée automatiquement.
- Indentation intelligente : le passage à la ligne positionne le curseur au bon endroit.
- Raccourci vers les définitions : un raccourci permet d'atteindre la définition

---

(4) Un compilateur crée directement un programme exécutable qui fonctionne sur le système d'exploitation choisi (Windows, iOS, Linux, Android, ...) ou à travers une machine virtuelle (Java) ; un interpréteur lit les instructions au fur et à mesure et les exécute immédiatement : au début de l'informatique le BASIC était interprété. On peut convertir un programme interprété en exécutable.

(5) Voir la doc ici <https://wiki.inria.fr/sciencinfoycee/ClefISN> ainsi que sur le forum SILO (<https://science-info-lycee.fr/>). Réservé aux experts chevronnés...

(6) En général les administrateurs refusent l'implantation des distributions type WAMP utilisant le PHP pour des raisons de sécurité. Il vaut mieux travailler en Javascript si on veut des applications web. Notez que les langages de description type HTML, XML sont interprétés mais ne sont pas des véritables langages de programmation.

et/ou le code d'une fonction, d'une classe, etc.

- Soulignement des erreurs en temps réel : les erreurs de syntaxe sont automatiquement marquées.
- Gestion des versions : lorsqu'on travaille à plusieurs sur un projet il faut utiliser des versions bien repérées sous peine de voir son code écrasé par les autres participants...

Et ces outils devraient être libres de droits et gratuits si possible ..

En fait pour débiter on pourra choisir une des solutions suivantes :

- Java et JavaScool (éditeur/tutoriel proposé par l'INRIA pour apprentissage), <http://javascool.gforge.inria.fr/>
- Java et Processing (éditeur/compilateur destiné au multimédia), <http://fr.flossmanuals.net/processing/>
- Python et AmiensPython (éditeur PyScripter pour Windows adapté), <http://amienspython.tuxfamily.org/>

Pour ma part je n'avais pas vraiment d'a priori sur un langage mais quand j'ai vu la facilité d'apprentissage et d'utilisation, la très riche bibliothèque, la qualité de Python, j'ai craqué pour AmiensPython et je n'ai qu'à m'en féliciter ... pour le moment.

Voici un exemple commenté de code Python sur le temps de vol maximum de la suite de Syracuse pour ceux qui ne connaissent pas :

<pre>from __future__ import division from lycee import *</pre>	<p>Utilisation de bibliothèques externes, lycee est une librairie spéciale Amienspython.</p>
<pre>def suivant(x) :     if reste(x,2) == 0 :         return quotient(x,2)     else :         return 3*x+1</pre>	<p>Fonction qui retourne le nombre qui vient après x dans la suite de Syracuse (mot clé def). reste et quotient sont définies dans la bibliothèque lycee. Les blocs de code sont indiqués par les indentations successives. Notez les deux points (:) avant chaque nouvelle indentation.</p>
<pre>def vol(x) :     L=[ ]     while x != 1 :         L.append(x)         x = suivant(x)     return L</pre>	<p>vol renvoie une liste L contenant toutes les valeurs de la suite de Syracuse en partant de x. L est initialisée à « vide » : [] On suppose que la suite finit bien à 1... L est un objet : on appelle la méthode append qui rallonge la liste L avec x. La sortie est la liste L du vol de x.</p>
<pre>tmax=0 for n in range(1,100001) :     t = len(vol(n))     if t &gt; tmax :         print "Temps de vol de"         ,t, "pour n =" ,n         tmax = t</pre>	<p>Programme principal : initialisation du temps de vol maximum à 0. La fonction len renvoie la longueur de la liste L obtenue à partir de la fonction vol. On pourrait fabriquer une liste des n donnant les temps de vol maximum successifs...</p>

Comme on le voit la syntaxe est « naturelle » et les principaux termes sont faciles à assimiler ; par ailleurs Python se charge de gérer le typage de données, la gestion de la mémoire, le type de retour des fonctions ; on peut même retourner des objets composites : par exemple `vol(x)` pourrait retourner **L**, **haut** où haut serait la valeur maximale atteinte en vol ; à ce moment `vol(n)` donnerait une liste avec `L = vol(n)[0]` et `haut = vol(n)[1]`.

C'est vraiment bien plus confortable pour les élèves et leur professeur qui ne passe pas son temps à chasser les petites erreurs bien cachées dans un coin de code !

C'est vrai que Java ou C# sont les langages les plus couramment utilisés dans les applications mais l'objectif n'est pas de former des professionnels dans l'immédiat<sup>(7)</sup>. Le temps gagné en remplaçant ces langages par un outil plus facile d'emploi sera mieux utilisé dans la méthodologie, l'algorithmique, etc. Par ailleurs il existe des outils permettant de convertir le code Python interprété en code C#, en Java, ou en exécutable ... sans oublier la richesse des bibliothèques comme Pygame, Scipy, Mathplotlib, etc.

## 5. Projets

Lorsque j'ai préparé ma première année d'ISN, cet été, j'ai envisagé un certain nombre de situations propices à la mise en oeuvre de projets intéressants et porteurs... J'avais donc préparé surtout des projets « de prof ».

Après trois séances de découverte-initiation, je me suis rendu compte de la grande hétérogénéité des élèves et de la difficulté à gérer mes groupes sans devenir un coureur de marathon pendant 2 heures... J'ai donc proposé aux élèves de réfléchir à un projet, soit à partir de mes suggestions, soit à partir de leurs désirs. Voilà ce que j'ai eu comme projets : jeu de ping-pong, jeu de Marienbad, fourmis de Langton, méthodes de Monte-Carlo, démineur, tableau de tournoi, jeu du pendu, jeu de plateau type Mario, jeu de tir au but, base de données sur des symptômes de maladie et les médicaments, super-balles, jeu de bonneteau, agenda, lancer de fléchettes, le voyageur de commerce, pierre-papier-ciseaux, simulation d'une montagne russe, test de QI, suivi de l'évolution de ses notes, lancer de balles, calculatrice.

Remarquez que c'est varié au moins, je n'ai pas vraiment le temps de m'ennuyer... Globalement il y a 2/3 de jeux, 1/3 d'activités autres dont deux orientées maths expérimentales. Un tiers des projets sont gérés par des élèves seuls, les autres sont par groupes de 2 ; j'ai formellement interdit les groupes de 3 ou plus à la lumière de mon expérience des TPE. J'ai également veillé à ce que les groupes ne soient pas trop hétérogènes : un élève très habile ne fonctionnera pas bien si on lui adjoint un camarade totalement inexpérimenté. Pour les débutants/intermédiaires j'ai insisté lourdement pour que les projets ne soient pas trop ambitieux, quitte à développer d'autres aspects une fois le projet initial terminé ; pour les experts qui arrivent avec des centaines ou des milliers de lignes de code j'essaie de cibler une situation et leur faire faire des modifications : ça me permet dans un premier temps de vérifier qu'ils

(7) Ceci dit Python ou Ruby qui lui est proche, sont de plus en plus utilisés dans l'industrie et la recherche ; il existe même un framework web (django) qui remplace avantageusement PHP.

n'ont pas copié du code bêtement (surtout quand c'est du C ou du Java) et de les obliger à mettre le nez dans des choses qu'ils ne me semblent pas maîtriser ; ces experts me prennent malheureusement plus de temps que les autres groupes. Certains groupes ont essayé de prendre des applications déjà réalisées mais comme ils n'y comprenaient pas grand-chose, ils ont vite abandonné cette idée.

Donc les élèves essaient de comprendre ce qu'ils doivent faire pour réaliser leur projet et quels outils utiliser : pratiquement tout est à démarrer à zéro, même sur des questions algorithmiques élémentaires, mais c'est cet aspect formateur qui est vraiment intéressant, aussi bien pour eux que pour l'enseignant !

En pratique, après quelques tâtonnements, j'ai décidé d'utiliser trois fiches techniques : la première, évolutive, décrit le planning des activités à effectuer pour mener à bien le projet ; la deuxième liste divers éléments devant impérativement être présents ou utilisés dans le projet, ces items seront évalués et me servent à préciser mes attentes aux élèves : il est évident que les élèves ont tout intérêt à jouer le jeu par rapport à cette fiche en vue de l'évaluation finale... Enfin une fiche de suivi à compléter à chaque séance et à m'envoyer systématiquement avec leur code ; ces fiches sont notées de manière à obtenir des évaluations « au fil de l'eau ». Cela me permet également de contrôler a minima ce qui se passe dans les groupes que je n'ai pas eu le temps de voir.

Pour l'instant tout ceci est en phase expérimentale : il y a eu deux séances où j'ai utilisé ce système et j'ai mis environ une demi-journée à la maison à chaque fois pour tout gérer ; avec un peu d'expérience ça devrait aller plus vite.

### Planning

Activités	Outils	Durée
...		
	Total	

### Ce que devra comporter votre projet

	Proposé	Accepté avec modifications	Accepté définitivement	Evaluation/ Note max
Descriptif détaillé				
Référence à une réalité/Contexte				
Commentaires du programme				
Références / Site-Bibliographie				
Algorithme principal				
Utilisation d'une bibliothèque				
Utilisation et création de fonctions				
Gestion de fichiers				
Structure de données				
Interface utilisateur				

*Pour l'utilisation de fonctions externes vous devrez être capables de donner systématiquement les paramètres d'entrée, l'action de la fonction, les paramètres de sortie et un exemple d'utilisation contextualisée.*

### Fiche de suivi

Groupe	
Date	N° fiche
Étape du projet : (selon le planning établi)	
Bilan : (où en est le projet ?)	
Éventuelle(s) difficulté(s) : (le projet est-il freiné ou bloqué, pourquoi ?)	
Solutions proposées : (comment surmonter cette difficulté ? Qui fait quoi ?)	
Prochaine étape du projet : (ce que chacun doit faire et pour quand)	
Observations particulières	

Les conseils donnés dans le cadre institutionnel étaient plutôt de démarrer avec un mini-projet couplé avec des activités disciplinaires puis de passer à un projet plus conséquent avec de nouveaux apports : cela signifierait arriver à mener deux projets dans l'année, ce qui peut être possible avec peu d'élèves (une dizaine maximum ?) mais certainement pas avec un groupe de 20...

## 6. Et finalement...

En fait je me suis aperçu au bout de cinq séances de ce type qu'il y avait trop de décalage entre ce que les élèves souhaitaient faire et ce qu'ils étaient capables de faire, et ceci pratiquement quel que soit leur niveau : j'ai donc interrompu brutalement le travail sur projets pour revenir à des activités plus scolaires d'apprentissage des bases de la programmation... Mes vacances de Noël ont donc été consacrées à la mise au point d'une batterie d'exercices et d'activités indispensables, de difficultés variées ; je fais également des contrôles régulièrement. Je prévois de travailler ainsi jusqu'aux vacances de Février ou même un peu plus tard, puis de repartir sur les projets, espérant que les élèves arriveront à être plus autonomes dans leurs apprentissages.

## 7. Informatique et société

*« Le professeur peut s'appuyer sur la mise en place d'exposés suivis de débats au sein de la classe pour introduire des questions sociétales liées à la généralisation du numérique. »*

Là encore j'avais préparé plusieurs questions sociétales en récupérant divers articles de journaux afin d'aider les élèves dans leurs choix d'exposés ce qui a été plus efficace que pour les projets... Il y a un quart d'heure par séance réservé à un exposé d'une dizaine de minutes + questions. Je commente immédiatement en essayant de ne pas me faire embarquer par le sujet...

Les sujets choisis par les élèves sont les suivants : téléchargement illégal, vol d'identité sur Internet, la e-réputation, la cyberdéfense, les grands bugs, la propriété

intellectuelle, ACTA et Hadopi, Big Brother, le « cloud computing », Samsung contre Apple, les supercalculateurs, les dangers liés à internet, les hackers, les Anonymous, l'évolution des ordinateurs, la création d'internet, les virus informatiques, les métiers de l'informatique, etc. Dans l'ensemble les exposés sont corrects, les élèves étant visiblement intéressés par cet aspect des choses.

## 8. Quelques conclusions très provisoires

Voilà, j'ai donc parlé de mon expérience et de ce que je vis depuis le début de l'année mais il m'a semblé important de faire partager mes doutes, mes interrogations et mes choix à d'autres : le forum organisé sur le site du SILO reste assez pauvre en retour d'expériences et est plutôt orienté vers la technique ; je n'ai pas trouvé d'autre forum, aussi j'imagine que la plupart des enseignants concernés sont à peu près dans la même situation que moi : pas mal d'enthousiasme, de la créativité, une relation aux élèves globalement enrichissante mais des questions assez importantes sans vraies réponses et un programme ainsi qu'une organisation plutôt mal adaptés aux contraintes temporelles et scolaires ... même si notre liberté pédagogique nous laisse une marge de manœuvre importante.

Il ne me semble pas que cette spécialité risque de poser trop de problèmes à la spécialité mathématique : les élèves ciblés ne sont pas les mêmes et les problèmes organisationnels (salles, emplois du temps, recrutement des enseignants) limiteront forcément les effectifs. Mais si le ministère décide d'étendre cet enseignement à d'autres classes (par exemple en TES/L ou en Première) les questions qui se posent déjà à l'occasion de la mise en place de ISN seront remplacées par d'autres, bien plus ouvertes et plus importantes par les conséquences des réponses qui seront apportées.

## Sitographie

Programme :

[http://www.education.gouv.fr/pid25535/bulletin\\_officiel.html?cid\\_bo=57572](http://www.education.gouv.fr/pid25535/bulletin_officiel.html?cid_bo=57572)

Profils d'enseignants :

[http://www.education.gouv.fr/pid25535/bulletin\\_officiel.html?cid\\_bo=57998](http://www.education.gouv.fr/pid25535/bulletin_officiel.html?cid_bo=57998)

Épreuve terminale :

[http://www.education.gouv.fr/pid25535/bulletin\\_officiel.html?cid\\_bo=57489](http://www.education.gouv.fr/pid25535/bulletin_officiel.html?cid_bo=57489)

Ressources Eduscol : <http://eduscol.education.fr/cid60671/ressources-isn.html>

Association EPI : <http://www.epi.asso.fr/revue/iticsom.htm>

Société Informatique de France : <http://www.societe-informatique-de-france.fr/>