Une liste de projets ISN.

Deux catégories de projets sont présentes principalement : les projets de programmation pure, et les projets robotique. Il est également possible de faire un projet sur papier uniquement (logisim), pas plus simple que les autres. Attention il n'y a que deux robots (dont un au professeur), donc il sera très compliqué de faire plus de deux groupes avec les robots. Les projets « robotique » rajoutent l'intérêt et la difficulté du réel (mesures des capteurs variant légèrement). Sur ces projets, vous aurez plus d'aide en conséquence. **Je vous préciserai le cahier des charges, une fois les projets choisis.**

- Les projets se font par groupe de deux. Pour les projets plus difficiles (***), il est possible de faire le projet à 3.
- L'interface graphique doit être séparée de la partie algorithmique (ou « métier »), dans une ou plusieurs fonctions à part. Cela signifie que la partie logique doit fonctionner de manière autonome. Les seules fonctions importées de la partie algorithmique dans la partie graphique sont les « get » et les « set ». On utilisera un minimum de variables globales (seulement les listes/tableaux de grande taille).
- Vous pouvez éventuellement programmer dans d'autres langages, à voir avec le professeur (Java, C et ses variantes, php, SQL en prime pour les bases de données). Attention dans ce cas : si dans un groupe un des participants ne connaît pas le langage en question, il est censé l'apprendre pendant le projet ! Certains projets se prètent particulièrement bien à la programmation objet, cela peut être une occasion de l'apprendre. En ce qui concerne la bibliothèque graphique, l'accent est mis sur l'usage de tkinter, qui est le standard sous python 3. De même, si vous utilisez une bibliothèque, par exemple pour les images, parlez-en au professeur pour éviter d'utiliser une bibliothèque dépréciée (ie qui n'est plus utilisée), ou bien pour éviter de l'utiliser tout court si c'est inutile. On n'utilisera pas la bibliothèque Pygame. On n'utilisera pas les langages de programmation propres aux jeux vidéos (comme celui de Warcraft 3 don't j'ai oublié le nom). Vous pouvez programmer en objet, pour certains sujets c'est vraiment plus simple d'ailleurs.
- Pour certains sujets (chiffrement, labyrinthes, graphes), votre honoré professeur vous fournira des exemples sous la forme adaptée afin que vous puissiez tester vos programmes.
- La notation au bac ne se fait pas sur la difficulté des sujets réalisés, mais sur votre maitrise de ce que vous avez fait (dans le respect du cahier des charges au minimum). La qualité de l'oral est indispensable (présentation maîtrisée, avec une très forte préférence pour une présentation de type diaporama ou prezi).

Exemple de progression version 1 :
Semaine 1 : étude du cahier des charges, imaginer le programme (lister tout ce qu'il doit faire)
Semaine 2 : déterminer les différentes fonctions de la partie métier/partie graphique, répartition des tâches dans l'équipe. Il est fortement déconseillé de programmer avant d'avoir fait les étapes des semaines 1 et 2.
Semaines 3 à 6 : version fonctionnelle de la partie algo (doit être fini à la fin des vacances de printemps, éventuellement début Mai si la rentrée de ces vacances est tôt)
Semaines 4 à 7 : version fonctionnelle avec la partie graphique. Suivant la répartition des tâches, se fait entièrement en parallèle ou bien un peu décalé de la partie algo
Semaine 8 : correction des derniers petits bugs, ajout des derniers commentaires.
Semaine 8/9 (suivant les années) : remise du rapport et du programme
Exemple de progression version 2 :
Idem pour les semaines 1,2 8 et 9. Pour les autres semaines, la partie algo est faite entièrement avant la partie graphique

- Le Jeu de la vie *
  Pour aller plus loin
  - Faire fonctionner une grille à la façon d'un tore (on peut penser à une chambre à air comme représentation d'un tore)
    - Tout ce qui disparaît par le bas réapparaît par le haut et inversement
    - Tout ce qui disparaît par la gauche réapparaît par la droite et inversement

- Le compte est bon *

Sujet avec une interface graphique minimale. Sujet pour ceux qui aiment réfléchir, les algorithmes de résolution sont de ceux qui apportent le plus au niveau programmation, parmis tous les sujets proposés ici. Peut permettre un approfondissement sur la récursivité (non obligatoire).

Pour aller plus loin

- Concevoir un algorithme résolvant le problème en utilisant le plus petit nombre de plaques possible
- Donner toutes les solutions possibles

- Le tic-tac-toe (2 ou 3 élèves) *(*). Variante puissance 4

Sujet très classique. L'intelligence artificielle est obligatoire, elle est plus dure à faire pour le puissance 4 (pas de stratégie gagnante à coup sûr me semble-t-il).

Pour aller plus loin

- Permettre à deux joueurs de jouer plusieurs parties d'affilée l'un contre l'autre en mémorisant les points gagnés par chacun des adversaires

- Bataille navale **

L'interface graphique n'est pas obligatoire. Moins simple qu'il n'y paraît.

On utilisera les règles suivantes : jeu humain contre ordinateur (idiot qui joue au hasard, mais pas deux fois au même endroit), les bateaux ne doivent pas se toucher.

Pour aller plus loin

- Faire une IA.

- Algorithme de Dijkstra : calcul d'un trajet le plus court possible par un GPS *

Facile mais pas fascinant.

- Générateur d'exercices de calcul mental *

Le sujet du projet

Cahier des charges de base

- Ecrire un programme qui génère aléatoirement un opérateur (les quatre opérations usuelles) et deux entiers. Les divisions doivent « tomber juste », les soustractions doivent donner un résultat positif.
- Ecrire un programme vérifiant que la réponse donnée par l'utilisateur est juste.
- Ecrire un programme permettant de choisir la bonne réponse parmi 4, une seule étant juste, les autres étant aléatoires (version QCM)
- Ecrire un programme permettant de faire plusieurs calculs et de donner un score en fonction de la difficulté
- Programmation d'une interface :
  - Choix de la difficulté (ou absence de choix)
  - Choix de la version : QCM ou non.
  - Affichage des questions, des réponses et du score final

Pour aller plus loin

- Dans la version QCM, donner la réponse juste et trois réponses plausibles.

- Calcul en valeur exacte sur des fractions *

La partie « pour aller plus loin » est la bienvenue, sinon le projet est un peu juste.

Présentation du problème.

Le but est de créer un programme permettant de faire du calcul sur des fractions, en donnant les résultats sous forme de fraction réduite.

Le sujet du projet

Cahier des charges de base

- Proposer un type de données pour représenter une fraction.
- Ecrire un programme qui étant donné deux fractions rentrées sous la forme ci-dessus, et un opérateur, renvoie le résultat.
- Ecrire un programme utilisant la méthode de Héron pour calculer la valeur approchée d'une racine carrée sous forme de fraction, la précision étant donnée.
  http://fr.wikipedia.org/wiki/M%C3%A9thode_de_H%C3%A9ron

Pour aller plus loin

- Compléter par un programme donnant l'écriture en fraction continue d'un nombre rationnel, ainsi que l'interface correspondante

- Algorithme de Pledge **

L'algorithme est simple à programmer. Par contre l'interface graphique donne plusieurs challenges

Présentation du problème.

L'algorithme de Pledge permet de sortir d'un labyrinthe, lorsque l'on part d'un point quelconque à l'intérieur. Un labyrinthe peut toujours être représenté sous forme de graphe, il est donc assez simple de programmer cet algorithme.

http://interstices.info/jcms/c_46065/l-algorithme-de-pledge

http://en.wikipedia.org/wiki/Maze_solving_algorithm#Pledge_algorithm

Le sujet du projet

Cahier des charges de base

- Programmer l'algorithme de Pledge étant donné un graphe représentant un labyrinthe, on considèrera que tous les murs sont à angle droit.
- Interface :
  - Etant donné un graphe représentant un labyrinthe, donner le sommet représentant la position initiale ainsi que le sommet représentant la sortie
  - Afficher la suite des directions permettant de sortir du labyrinthe

Pour aller plus loin

- Génération automatique, ou par "point and click " du labyrinthe

- Stéganographie et traitement des images **

L'algorithme est assez simple à programmer, une fois qu'on l'a compris... Très beau sujet.

Présentation du problème.

La stéganographie est « l'art de la dissimulation ». Il s'agit de cacher une image dans une autre (ou un son dans un autre etc…). Soit pour jouer aux espions, soit pour mettre un copyright sur une image

Le sujet du projet

Cahier des charges de base

- proposer un algorithme qui, étant donné deux images dans un format simple non compressé, l'une étant « lourde » et l'autre « légère » (en taille), cache l'image légère dans la lourde. On utilisera la technique du bit de poids faible (cf http://fr.wikipedia.org/wiki/St%C3%A9ganographie). On réfléchira sur la taille relative des images.
- Il est impératif de travailler à partir des outils proposés par le professeurs (programme « utilitaires images » et format ppm de préférence). Sinon cela devient vite compliqué. Par ailleurs, travailler dans un format compressé (comme le jpg) détruit l'image cachée.
- Proposer un autre algorithme qui extrait une image cachée dans une autre (ie l'opération inverse du précédent)

Pour aller plus loin

- compléter par d'autres effets sur une image (par exemple le contour avec des masques sophistiqués).
- Utiliser une autre méthode de stéganographie

- Cryptanalyse du chiffre de Vigénère *

Pas d'interface graphique, sinon minimale. Le décrytpage se fera suivant la méthode de Kasiski et Babbage décrite ici : http://fr.wikipedia.org/wiki/Cryptanalyse_du_chiffre_de_Vigen%C3%A8re

Pour aller plus loin : Ecrire un programme permettant transfromant un texte standard en texte sans signes diacritiques.

- Pendu *

Pour aller plus loin ***(*)*: chercher des mots au hasard dans un dictionnaire sur internet

- Solitaire *

- Snake **
  Se prète bien à la programmation objet.
  Pour aller plus loin
  - Rajouter les fruits qui font changer de sens, ainsi que d'autres effets amusants (vitesse par exemple)

- Casse-brique **
  Se prète bien à la programmation objet.
  Pour aller plus loin
  - Rajouter des effets sur certaines briques (taille de la raquette, accélération, ralentissement de la balle)

- 2048 **(*)
  Pour aller plus loin
  - Jouer avec des images plutôt que des nombres, sur interface graphique.

- Algorithmes génétiques, IA et jeux vidéos***
  L'idée est de programmer une IA qui apprend toute seule de ses erreurs, avec un algorithme génétique, pour aller le plus loin possible dans un jeu vidéo. On choisira un jeu très simple (exemple flappy bird, ou un infinite runner avec juste deux types d'obstacles). Ce sujet donne le double de travail : jeu vidéo + alogrithmes génétiques.

- Tétris ***
  Tout est dur dans ce projet. A commencer de préférence directement en programmation objet.

- Tower défense **
  A commencer de préférence directement en programmation objet.

- Robotique : coordination des applaudissements ***
  Sujet original.
  Présentation du problème.
  Lors d'un concert ou d'un spectacle, vous avez tous constaté qu'une foule peut se coordonner pour applaudir en rythme. L'idée de ce projet est de modéliser le phénomène avec arduino, et des signaux lumineux plutôt que sonores. On pourra utiliser une dizaine de modules arduino, chacun représentant une personne. Ce projet fait partie des problèmes dits « d'intelligence collective ».
  Le sujet du projet
  Cahier des charges de base
  - Chauqe arduino doit avoir le même programme.
  - Programmer une génération d'un signal lumineux aléatoire périodique.
  - Acquérir les intensités émises régulièrement (par exemple 10 fois par seconde), sur un temps assez long.
  - Extraire les deux signaux les plus « proches », en tenant compte des possibles variations de lecture de la lumière, et de leur période inconnue a priori
  - Coordonner le signal émis avec ces deux signaux reçus, en faisant la moyenne.
  - Recommencer indéfiniment les trois étapes précédentes

  Pour aller plus loin
  - Rajouter une perturbation aléatoire après un certain temps, lorsque tous les signaux sont coordonnés

- Robotique : algorithme de Pledge ***
  Le labyrinthe sera représenté par les chemins et non par les murs. En effet il existe déjà un programme permettant de suivre des traces noires sur un fond blanc.
  Présentation du problème.
  L'algorithme de Pledge permet de sortir d'un labyrinthe, lorsque l'on part d'un point quelconque à l'intérieur.
  http://interstices.info/jcms/c_46065/l-algorithme-de-pledge

http://en.wikipedia.org/wiki/Maze_solving_algorithm#Pledge_algorithm
Pour que ce projet soit plus simple, les murs seront représentés par des surfaces noires au sol. Le robot roulera sur le bord de ces surfaces, ce qui est bien plus facile que de rester à distance constante d'un mur.

Le sujet du projet

Cahier des charges de base avec le robot arduino, similaire avec les autres robots)

- Comprendre le fonctionnement du robot arduino : servomoteurs, boussole, comment suivre une ligne.
  http://arduino.cc/en/Guide/Robot pour les moteurs et la boussole, ainsi que le tutoriel « line following ». Voir aussi la commande « updateIR »
- Programmer l'algorithme de Pledge sur le robot arduino (votre honoré professeur ou bien le lycée vous fournira les planches pour faire le sol du labyrinthe, ainsi que du scotch noir)

Pour aller plus loin

- Choisir aléatoirement si l'on « longe » les murs par la droite ou la gauche
- Rajouter des indicateurs (diodes) pour les actions en cours

- Robot télécommandé * ou ** suivant le type de robot télécommandé

Présentation du problème.

Comme son nom l'indique, le but est de commander un robot à distance.

Le sujet du projet

Cahier des charges de base

- Comprendre le fonctionnement du robot parallax/arduino : servomoteurs, et éventuellement compléments.
  http://learn.parallax.com/ShieldRobot pour la partie servomoteurs, en comprendre l'essentiel dans le chapitre 2
- Comprendre le protocole de communication xbee et le logiciel X-CTU (livre du professeur, en particulier p 467)
- Réaliser un « mappage » qui permet de passer des deux axes x/y du joystick Tinkerkit, aux deux moteurs droite/gauche du robot.
- Programmer le robot et la télécommande afin qu'il communiquent

Pour aller plus loin

- Rajouter une sécurité avec les détecteurs de contact, pour que le robot ne heurte pas les obstacles.

- Logisim (2 ou 3 élèves)

Présentation du problème.

Le but est la création d'une UAL 2 ou 3 bits, 4 opérations, sur papier, à l'aide du logiciel Logisim.

- Robotique et réseaux : afficheur météo (2 ou 3 élèves).

Présentation du problème.

Le but est de créer un gadget connecté qui donne des informations récupérées sur le web. Ce genre d'objet est sans fil, mais on ne transmettra pas les données sans fil dans la version de base. Attention sur ce projet, l'aide du professeur sera minimale, vu les faibles compétences de celui-ci en réseaux...

Le sujet du projet

Cahier des charges de base

- Comprendre le fonctionnement du module lcd http://www.tinkerkit.com/lcd/
- Comprendre le shield ethernet (livre du professeur)
- Comprendre comment on récupère des données sur le web (livre du professeur)
- Afficher des données météo d'un endroit spécifié.

Pour aller plus loin

- Rajouter une alarme « vous avez reçu un message » en surveillant une boite e-mail.
- Rajouter une connexion xbee sans fil

Quelques indications pour la rédaction du rapport :

Ce rapport doit impérativement être réalisé avec un outil informatique (pas à la main ni à la machine à écrire !). En cas d'absence du rapport, ou de rapport fait "à la main", vous êtes notés sur 12. Utilisez Word, OpenOffice ou un de ses forks (fork = fourchette en anglais, siginife variante d'un programme en informatique).
Vous pouvez le faire de manière classique, sous la forme d'un rapport écrit. Dans ce cas il doit faire au minimum 5 pages hors annexes, sommaire, bibliographie. Au total, il doit faire de 5 à 10 pages, toujours hors annexes. Vous verrez qu'il est difficile de faire peu de pages. Vous pouvez faire un peu plus si nécessaire, mais pas trop (15 pages me paraissent déjà être trop). Vous voudrez bien me fournir une copie papier de ce rapport.
Vous pouvez également sous la forme d'un site web, que je récupèrerai. Sous cette forme, il faudra conserver les parties importantes de la structure « papier », en modifiant la présentation. Il n'y aura pas de résumé de 4ème de couverture bien sûr.
Les deux formats ont leurs avantages et leurs inconvénients. Vous ne serez pas notés sur le choix d'un format para rapport à l'autre. Le format « web » est plus long à faire, mais montre que vous avez des bases de html, et évite de faire une copie papier. Par contre il est plus dur de faire apparaître la structure du document, cela peut vite être brouillon.

Une possibilité pour la structure du document papier:

- Couverture
- Sommaire
- Introduction
- Cahier des charges (c'est à dire ce que vous devez faire).
- Rapport technique. C'est le coeur du rapport.
- Manuel d'utilisation
- Conclusion
- Glossaire, bibliographie
- Résumé

Structure du site web:

- Page de présentation :
  - introduction
  - hyperliens pour le sommaire, avec indications de ce à quoi corrrespondent ces hyperliens.
  - Il n'y a pas de couverture bien sûr, mais doit figurer l'équivalent sur le site
- Cahier des charges (ce que vous devez faire)
- Rapport technique (comment vous l'avez fait, faites notamment apparaître la manière dont vous avez collaboré dans le groupe)
- Manuel d'utilisation
- Conclusion
- Glossaire, bibliographie
- Résumé

Quelle que soit la forme du rapport/site, le cahier des charges et le rapport technique sont indispensables.

Couverture (pour la version papier ; dans la version web ces informations figurent d'une manière ou d'une autre sur la page d'accueil):

- Doit figurer le titre du projet, le nom des auteurs, le lieu (Lycée Jean Jaurès Saint Clément de Rivière Académie de Montpellier), la période/date, éventuellement le nom de l'enseignant, « spécialité ISN terminale S », « présenté pour l'obtention du Baccalauréat », « soutenu en … (mois à préciser, très probablement Mai) ».

Sommaire (version papier, correspond aux hyperliens dans la versin web):

- Ne pas oublier la numérotation des pages. Tous les éditeurs de texte modernes ont automatiquement cette fonctionnalité. Si vous ne la trouvez pas faites le sommaire à la main, ici c'est un petit rapport. Si votre rapport comporte de nombreuses figures (y compris les algorithmes, photos, etc…) il est d'usage de faire également une table des figures, celles-ci pouvant être numérotées de manière automatique, et

ainsi la table est crée par le logiciel.

Introduction :

- expose un résumé du sujet et du résultat, annonce le plan. Évitez les clichés (« j'ai fait ce sujet parce qu'il m'intéresse très beaucoup …») même s'ils correspondent à une réalité. Dans ce cas trouvez une formulation plus élégante ou détournée.

Cahier des charges :

- Présente le sujet et son contexte
- Liste des fonctionnalités attendues (ce qui est demandé dans le projet),

Eventuellement un schéma sur l'articulation desdites fonctionnalités,

- Eventuellement des exemples d'utilisation.
- Vous pouvez également préciser ce que vous avez fait en plus / différemment de ce qui était demandé dans l'énoncé du sujet.
- Le lecteur potentiel ne cherche pas à connaître le côté technique (par exemple il n'y connait rien en informatique, ou bien il veut refaire votre projet, sans se spoiler le côté fun du technique). Il s'agit dans cette partie de bien expliquer tout ce que le programme fait / est censé faire, sans expliquer comment il le fait.

Rapport technique :

- Choix technologiques (langages, explication de la programmation objet, bibliothèques particulières,...)
- Algorithmes. Les algorihtmes ne sont pas du Python, même traduit en français. Ils sont écrits en « langage naturel », comme ce que vous faites en mathématiques. Seuls les algorithmes importants sont écrits, et on n'y met que les étapes importantes du calcul. Par exemple, on ne met pas les fonctions d'affichage, de création de matrice, ni un algorithme pour tirer une valeur aléatoire dans une liste, on peut écrire « échanger les valeurs de *a* et *b* sans détailler, etc...
  Les algorithmes détaillant une fonction importante sont spécifiés comme en programmation, afin que l'on en comprenne bien le fonctionnement.
  Un algorihtme est commenté. S'il est très complexe, on peut le présenter en préambule en quelques lignes, en faisant une liste des étapes importantes.
  Il est impératif de respecter l'indentation pour la présentation des algorithmes.
- Les choix de structures de données utilisés, avec éventuellement une justification, si ces choix peuvent paraître biscornus.
- architecture du programme (liste des fichiers et découpage en fonctions), articulation des fonctions (un schéma peut être le bienvenu). En annexe du rapport, on précise la liste des fonctions et leurs spécifications, sans aucun détail. Si vous avez respecté le format des spécifications en « docstrings », il existe des programmes qui les extraient automatiquement. Mais vous passerez plus de temps à comprendre comment fonctionnent ces programmes qu'à faire des copier/coller.
- Faites également figurer dans cette partie la répartition du (et/ou la méthode de) travail dans le groupe
- Le lecteur potentiel est un de vos camarades, qui a parfaitement maîtrisé tout ce qu'on aura vu cette année, qui veut par exemple améliorer ou corriger votre programme.

Manuel d'utilisation :

- on imagine que le lecteur est cette fois-ci l'utilisateur de votre programme, et de plus il ne connaît rien à l'informatique. Il faut donc lui expliquer comment installer puis utiliser le programme. Cette partie est en général courte. Si vous ne voyez pas quoi écrire, pensez à votre grand-père qui rame dès qu'il touche à un ordinateur (ou toute autre personne si votre grand-père est un boss de l'informatique), et faites ce mode d'emploi pour lui.

Conclusion (dans la version web, cette conclusion peut paradoxalement figurer en fin de page d'accueil):

- Rappelle le sujet et le programme réalisé. Présente des perspectives : reste-t'il des bugs à corriger ? Peut-on rajouter des fonctionnalités ?

Glossaire :

- uniquement sur le vocabulaire compliqué, si nécessaire (inutile de définir ce qu'est une souris)

Bibliographie :

- donne toutes les références citées à l'intérieur du rapport, et uniquement celles-ci. Si ce sont des sites précise la date de consultation des pages web.

Résumé (version papier uniquement) :

- en quatrième de couverture : décrit le problème et le logiciel réalisé.

Annexes:

- notamment les spécifications de toutes les fonctions. Les annexes sont faites uniquement pour ceux qui veulent aller loin dans la compréhension de votre programme ; elles ne sont pas rédigées (un titre et le contenu suffisent)

Indications sur la soutenance (en général 3ème ou 4ème semaine de Mai)
Les projets (rapports et programmes) devront être impérativement rendus le jeudi d'avant.
La soutenance est individuelle. Les élèves de chaque projet passent les uns après les autres, vous pouvez échanger l'ordre de vos convocations si nécessaire à l'intérieur d'un groupe (mais pas entre groupes).
Première partie 8 minutes : présentation orale du projet. Si l'élève ne présente pas/n'a pas fait de dossier, ou que le dossier fait moins de 5 pages, ou a été écrit à la main, etc…, l'élève a 0 à cette partie.
Deuxième partie 12 minutes : dialogue/questions. D'abord sur le projet, puis sur le reste du programme.

Lors de la première partie (celle où vous présentez tout seul, sans que l'on vous interrompe)

- Faites nous éventuellement une démo de vos programmes
- Coordonnez-vous dans le groupe pour que chacun présente une partie du programme. Vous devez quand même comprendre le reste du programme, mais moins en détail.
- Je vous conseille une présentation de type diaporama ou Prezi (qui est à power point ce que l'internet est au minitel).

Deuxième partie (celle où on vous pose des questions)

- Pas de panique, on vous posera des questions pour éclaircir des points de votre présentation, ou des choix que vous avez fait. Ce ne sont pas des questions pièges.
- On peut aussi vous poser des questions moins précises sur les parties de votre projet présentées par vos camarades.
- On vous posera probablement une ou deux questions/exercices portant sur le reste du programme. Révisez tous vos cours depuis le début de l'année (codage machine, architecture, calcul booléen, problèmes sociétaux)

- La durée de 20 minutes est indicative à 1 ou 2 minutes près en plus ou en moins. Les élèves bénéficiant du 1/3 temps ont 27 minutes au total.
- Arrivez 10 minutes en avance, surtout pas en retard le planning étant très tendu.