

Listes et idées de Projets pour la NSI

QK

Listes et idées de Projets pour NSI

Extrait du programme :

Une part de l'horaire de l'enseignement d'au moins un quart du total en classe de première doit être réservée à la conception et à l'élaboration de projets conduits par des groupes de deux à quatre élèves.

Les projets réalisés par les élèves, sous la conduite du professeur, constituent un apprentissage fondamental tant pour la compréhension de l'informatique que pour l'acquisition de compétences.

En classe de première comme en classe terminale, 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 ; dans la mesure du possible, il convient de laisser le choix du thème du projet aux élèves eux-mêmes. Il peut s'agir d'un approfondissement théorique des concepts étudiés en commun, d'une application à d'autres disciplines telle qu'une simulation d'expérience, d'un travail sur des données socioéconomiques, du développement d'un logiciel de lexicographie, d'un projet autour d'un objet connecté ou d'un robot, de la conception d'une bibliothèque implémentant une structure de données complexe, d'un problème de traitement d'image ou de son, d'une application mobile, par exemple de réalité virtuelle ou augmentée, du développement d'un site Web associé à l'utilisation d'une base de données, de la réalisation d'un interprète d'un mini-langage, d'un programme de jeu de stratégie, etc.

Mode d'examen

Selon le mode d'examen retenu l'enjeu des projets varie

- Soutenance orale : le projet devient l'examen
- Écrit : le projet est un simple élément d'apprentissage

Comment évaluer ?

Il faut envisager la préparation **d'un écrit**

- Questions théoriques
- Problème *@informatique pour tous* de CPGE ?
- Mise en situation et passage de code papier ?
- Évaluation de connaissances ?

Répondre aux deux modes d'examen

4H / semaine

- 1H de cours
- 2h de TP
- 1h de projet

Path to success

Que des projets courts

1. **Toussaint** - *thème imposé* - initiation à la méthode - présentation
2. **Noël** - *thème imposé* - Refonte et équilibrage des groupes - application
3. **Hiver** - *thème au choix* - objet connecté, multiple technologies, communication
4. **Printemps** - *thème au choix* - peut déborder fin d'année - multiple techno, pur logiciel,

Sujets de projets par période

difficulté : de facile * à très difficile ****

Les projets sont le terrain d'expression des meilleurs. Il y en a toujours eu en ISN/ICN, il y en aura en NSI. C'est l'occasion aussi pour les novices d'accomplir quelque chose de gratifiant.

1. **Toussaint** - Python : control flow, tableaux, dictionnaires
 - Lexicométrie (stats dans un doc) * se poursuit en Distance de Levenshtein ***
 - Génération de fractales (arbre Pythagoricien ?) ***
2. **Noël** - un JV Pygame : tableaux, librairies, événements, couleurs, graphismes
 - Snake ou Timberman basique ** avec sprites, animations ***
 - Flappy Bird ou Mario like basique ***
 - 2048, Othello... liste infinie en fait ****
3. **Hiver** - IHM, réseau, stockage/utilisation de données
 - Objet connecté : interface pour RPI Sense Hat : changer d'animation en pressant les touches, afficher les données à l'écran*

- Générateur aléatoire d'art (Tkinter, PIL) ****
 - Système d'alarme avec capteur, NFC, alerte, photo automatique et transmission réseau ***
 - Présentation de relevés Météo OpenWeatherMap (flask, bdd, js, graphe) ***
 - Science : analyse statistique et présentation de données ???
4. **Printemps** - Algorithmique, UI, réseau
- Langton / Game of Life dans pygame avec controles et dépos de cellules **
 - Comparaison de tris, autre tris, sound of sort etc. ** **** ****
 - Mouvement Brownien / Choc élastique ** ****
 - Sandpiles (dimension modeste) **, extensions nombreuses
 - Générateur de labyrinthe + comparaison de recherche de sortie ***
 - Jeu de stratégie ***
 - Jeu vs ordinateur + approche bourrinne @ reversi ***
 - IHM : annoying chat bot telegram / discord *****
 - Simulation de circuits logiques (ouch) *****
 - ML : NEAT *****

TP : thème abordé et sujet

P : au programme

C : complément

✓prêt avec Colab

✍brouillon

⊘pas - prêt du tout

1. ✍- P : **Python Basic** : =, control flow (if, for, while), def
 2. ✍- P : **Python str, list, tuple, dic** : manipulations simples : atspace
 3. ✍- P : **Python, Données en tables Manipuler un CSV** - gdrive
 4. ✓- P : **Python, Données Convertisseur de base**
 5. ⊘- P : **Python, Données Compléments binaire (somme, produit, complément à 2)**
 6. ✓- P : **Python, Données Flottants**
 7. ⊘- P : **Python, Données Représentation d'un texte en machine (ascii, iso-8859-1, unicode). Identifier l'intérêt, convertir**
 8. ✓- P : **Python, Algorithmique Recherche dichotomique**
 9. ⊘- P : **Python, Algorithmique Algorithmes gloutons : problème du Sac à dos** :
10. ✓- P : **Python, Algorithmique, librairies Tri insertion Tri sélection**
C (bulle, rapide, merge - terminale)

11. ✓- P : Python, Algorithmique *Initiation au Machine Learning : KNN*
:
12. ✓- C : Python, Données, Librairies *Manipulation d'images avec PIL / pillow*
13. ⓪- C Python *itérer sur n'importe quoi* :
14. ⓪- C Python (terminale) *Classes* :
15. ⓪- C Python, Données en tables : *Manipuler un JSON* :
16. ⚡- P Python, Algorithmique *Gloutons : problème du Sac à Dos, rendu de monnaie cas super croissant*
17. ⚡- C Python, Algorithmique *Gloutons : problème du Sac à dos - cryptographie avec Pb du Sac à dos*
18. ⚡- C Python, Algorithmique *Gloutons : problème du Sac à dos - recherche solution hors cas super croissant*
19. ⓪- C Python, Algorithmique *Approfondissement : KNN en Sklearn et autres algos (réseau de neurones ?) pour reconnaître manuscrits* :
20. ⓪- C Python, Algorithmique *Pygame, automate cellulaire (feu de forêt)* :
21. ⓪- C Python, Algorithmie, librairie *Créer une bibliothèque de fonctions vectorielles (somme, produit par un scalaire, norme scalaire, translation, réflexion, rotation etc.)* :
22. ⓪- C : Python, librairie *Matplotlib : initiation graphe, trucs bidons* :
23. ⓪- C : Python, librairie *Créer des fenêtres avec Tkinter* :
24. ⓪- C : Python, librairie, réseau *Créer un serveur Socket* :
25. ⓪- C : Python, librairie, réseau *Transmettre des données avec MQTT, les stocker* :
26. ⓪- C : Python, librairie, IHM web *Créer un serveur web avec Flask, utiliser un template HTML* :
27. ⓪- C : Python, librairie, tables *sqlite3 (basique) : des données dans une BDD SQL* :
28. ⓪- C : Python, IHM, événements *Pygame, découverte* :
29. ✓- P : Langages *Suite de Fibonacci en Python, C, JS, langage machine*
:
30. ✓- P : Architecture *Découverte de Linux sur une VM ou un Raspberry Pi : fichiers, droits, paquets, ssh*
31. ⓪- P : Architecture *Les constituants d'un PC moderne, Smartphone, Camera IP, Ordinateur ~1970 / 1980 / 1990 (que leur faire faire ?) (terminale)*
32. ⓪- P : Architecture, données *Raspberry Pi : étude de quelques composants, créer un circuit simple* :
33. ⓪- P : Réseau *Consulter des configurations, réparer un réseau simple (sûrement sur RPI)* :
34. ✓- P : Réseau *Découverte d'un simulateur de réseau, Mise en place d'un réseau avec le simulateur*
35. ✓- P : Réseau *Analyser des trames avec WireShark*
36. ⓪- C : Réseau *Ethical hacking avec WireShark (pourquoi https ou sftp ?) - version poussée avec Kali Linux sur WM ou RPI + mode*

37. **Ø- C : Réseau** Réseau complet élaboré CF fin Tuto OpenClassRooms :
38. **Ø- P : IHM sur le Web : P** HTML, CSS : les bases, prise en main de la console Chrome / Firefox, décrire des éléments de code :
39. **Ø- P : IHM sur le Web : P** JS : repérer, analyser quelques scripts simples, modifier sur le web quelques scripts. Programmer un événements (*clavier*) :
40. **Ø- P : IHM sur le Web : P** Comprendre les codes HTTP (implémenter dans FLASK ?) :
41. **Ø- P : IHM sur le Web : P** REST : comprendre GET et POST :
42. **Ø- C : Veille technologique : C** Les serveurs web d'hier et d'aujourd'hui : WAMP, NGINX, node.js, Django,