PX 212 : Projet développement

- Administratif:
  - Enseignants:
    - Quentin Giorgi,
    - Oum-el-kheir Aktouf
  - 3 ECTS (soit plus de 80h de travail en tout par personne)

    http://ec.europa.eu/education/ects/users-guide/assets/ects_users-_guide_web.pdf
  - Evaluation:
    - QCMs
    - Soutenance et présentation
    - Comportement durant les séances (engagement, assiduité, dynamisme, initiative, etc.) (en bonus/malus)

PX 212 : Projet développement

- Objectifs pédagogiques :
  - Conforter son niveau de maîtrise du langage C
  - Concevoir et réaliser un programme conséquent
  - S'approprier les outils : compilateur, debugger, (gdb,ddd), analyse de code (valgrind), etc.
  - Réaliser un travail de groupe (en trinôme) en respectant délais et objectifs.
  - L'objectif est avant tout d'acquérir de bonnes méthodes et de bonnes pratiques.
  - Il faut donc privilégié du code « propre » au-delà du résultat final (développement de toutes les fonctionnalités)

PX 212 : Projet développement

- Sujet :
  - Sokoban est un casse-tête qui demande logique, réflexion et ténacité. Il ne requiert ni réflexes ni rapidité, ni agilité, ni agressivité (il n'y a rien à tuer !)
  - Le jeu comprend plusieurs centaines de tableaux tous différents et de difficulté variable. Les tableaux sont dessinés sur une grille carrée où un manutentionnaire peut se déplacer d'une case dans chacune des quatre directions (haut, bas, gauche, droite). Les déplacements en diagonale et les sauts sont exclus.
  - Le but est pour le manutentionnaire, que vous dirigez, de ranger tous les caisses du tableau à la place qui leur est destinée. Les caisses n'ont pas de poignées, ce qui fait qu'on ne peut que les pousser et non les tirer. De plus, elles sont trop lourdes pour qu’on puisse en pousser deux à la fois.

PX 212 : Projet développement

- Analyse fonctionnelle :
- Fonctions de base
  - Lecture tableaux depuis un fichier au format texte standard (voir levels.lvl)
  - Affichage d'un tableau
  - Gestion des entrées (touches) clavier
  - Gestion des mouvements et poussées par l'utilisateur
  - Passage au tableau suivant quand il est terminé
  - Possibilité de recommencer un tableau depuis le début (souvent utile !)
  - Comptage du nombre de mouvements et de poussées
  - Possibilité d'annuler un nombre quelconque des coups précédents
  - Possibilité de sauvegarder pour reprendre la partie plus tard exactement au point où on l'a laissée
  - Enregistrement de la solution quand on réussit un tableau.

PX 212 : Projet développement

- Analyse fonctionnelle :
- Fonctions avancées
  - Dans les tableaux complexes, la résolution devient laborieuse. Les fonctions suivantes permettent au joueur de s'économiser et de se concentrer sur les difficultés du tableau.
    - déplacement du manutentionnaire par sélection sur la case destination : le programme doit déterminer un chemin possible (s'il existe) pour le manutentionnaire (sans déplacement de caisses) et lui faire suivre ce chemin.
    - déplacement d'une caisse par sélection : « clic » sur la caisse puis sur la case destination. Le programme détermine les déplacements et poussées du manutentionnaire pour amener la caisse à sa destination (bien entendu, il ne doit déplacer aucune autre caisse).

PX 212 : Projet développement

- Analyse fonctionnelle :
- Fonctions optionnelles (pour ceux qui ont tout fini)
  - Utilisation d’une bibliothèque graphique (par exemple SDL) pour gérer les affichages et les contrôles de déplacements.
  - Réalisation d’un éditeur de tableaux

PX 212 : Projet développement

- Analyse fonctionnelle :
- Fonctions optionnelles (pour ceux qui ont tout fini)
  - Utilisation d’une bibliothèque graphique (par exemple SDL) pour gérer les affichages et les contrôles de déplacements.
  - Réalisation d’un éditeur de tableaux

PX 212 : Projet développement

- Modalités pédagogiques :
  - Pas d'aide de l'enseignant tant que pas d'utilisation du debugger.
  - Règles / bonnes pratiques de codage (exemple):
    - Définir des règles de nommage entre vous
    - Une fonctionnalité = un fichier source (compilation séparée), attention aux contenus des fichiers d'entête. Identifier les fonctions et variables exportées de celles qui ne le sont pas.
    - Effectuer des fonctions de tests unitaires.
    - Indiquer les commentaires au format doxygen et générer la documentation.
    - Obligation d'avoir un Makefile.
    - Pas de VLA, pas de déclaration de variables au milieu du code, limiter l'usage des variables globales à celles strictement nécessaires.
    - Option de compilation « minimum » : -Wall -g -fsanitize=address

PX 212 : Projet développement

Des Questions ?

PX 212 : Projet développement

Des Questions ?

DEFINIR et JUSTIFIER la structure de données représentant les niveaux d'un jeu de sokoban, en tenant compte de la manière dont les niveaux vont être utilisés dans le reste du programme.

INDIQUER comment vous allez TESTER vos fonctions et VERIFIER qu'elles se comportent comme vous l'aviez prévu, gère les cas particuliers (par exemple fichier de niveaux mal formé, etc)

DEFINIR et JUSTIFIER la portée des structures de données, la méthode d'allocation de la mémoire, en quoi cela impose des contraintes ou hypothèses particulières, comment elle sera utilisée dans les différentes fonctions.