Année 2015/2016
Jean-Marc Salottti, Edwige Clermont, Olivier Chator

Projet informatique du module génie logiciel

Résumé : L'objectif de ce projet est de réaliser un logiciel pour une équipe de planification des activités des astronautes lors d'une simulation de mission à la surface d'une autre planète.

Figure 1 : Salle de Mission Control de la NASA (JPL).

Dans une salle de contrôle de mission dédiée au domaine spatial, illustrée figure 1, on réunit de nombreux experts de différentes spécialités. Ils suivent en temps réel la mission, sont attentifs à la moindre difficulté et apportent leurs conseils sur les décisions à prendre. Il y a par exemple des experts qui sont en charge du bon déroulement de la mission du point de vue technique (position, orientation, état des batteries, etc.), d'autres de la communication avec l'engin spatial, d'autres encore qui planifient les activités heure par heure pour les prochaines 48 heures, ainsi que des scientifiques qui analysent les données reçues. S'il s'agit d'une mission habitée, il y a en plus une équipe chargée de surveiller les astronautes, que ce soit du point de vue physiologique, psychologique ou relationnel.

Plaçons-nous dans le contexte d'une mission habitée martienne. On s'intéresse dans ce projet à un outil de planification qui serait utile à l'équipe en charge de planifier les activités des astronautes. Notez que cet outil pourrait être exploité ultérieurement lors d'une simulation organisationnelle effectuée dans une salle de simulation de l'ENSC. Les spécifications sont les suivantes :

Gestion des activités

Les activités sont classées en 6 catégories et plusieurs sous-catégories, selon la hiérarchie indiquée figure 2. L'application est consacrée à la gestion des activités. Pour cela, on gère un planning sur 500 jours, qui correspond à la durée de la mission en surface. Il faut 3 niveaux de visualisation sur 3 fiches différentes.

- Activity
  - Living
    - Eating
    - Sleeping
    - Entertainment
    - Private
    - Health control
    - Medical act
  - Science
    - Exploration
      - Space suit
      - Vehicle
    - Briefing
    - Debriefing
    - Inside experiment
    - Outside experiment
  - Maintenance
    - Cleaning
    - LSS air system
    - LSS water system
    - LSS food system
    - Power systems
    - Space suit
    - Other
  - Communication
    - Sending message
    - Receiving message
  - Repair
    - LSS
    - Power systems
    - Communication systems
    - Propulsion systems
    - Habitat
    - Space suit
    - Vehicle
  - Emergency

Figure 2 : Liste des activités

- Au premier niveau, le planning indique les 500 jours numérotés de 1 à 500 (à vous de choisir la solution la plus ergonomique pour présenter le calendrier). Les jours déjà passés doivent apparaître en gris, le jour courant doit être coloré en bleu et les jours suivants en vert. Afin de distinguer les jours où il y a une sortie en extérieur, que ce soit dans le passé ou programmé dans le futur, on affichera un petit astronaute habillé en scaphandre. On doit pouvoir cliquer sur n'importe quel jour pour passer au deuxième niveau de détail.
- Au deuxième niveau, on affiche les activités de la journée, sur 24 heures 40' (durée du jour martien), avec une précision de 10 minutes. La première fois qu'on arrive sur une journée donnée, aucune activité n'a été programmée. Toutefois, par défaut, on indiquera la journée type suivante :
  - 0h à 7h : Sleeping
  - 7hà 8h : Eating
  - 8h à 12h : Private
  - 12h à 14h : Eating
  - 14h à 19h : Private
  - 19h à 21h : Eating
  - 21h à 23h : Private
  - 23h à 24h40 : Sleeping

  Il doit être possible d'insérer une nouvelle activité ou d'en modifier une existante par un simple clic, ce qui conduit à la fiche de niveau 3. Il doit également être possible de supprimer une activité. Un bouton doit permettre de passer au jour précédent, un autre au jour suivant, et un autre de revenir au calendrier global.
  Enfin, il faut également associer à cette journée un compte-rendu, sous la forme d'un texte de 1000 caractères maximum. A noter qu'il doit être impossible de modifier le planning des activités des jours déjà écoulés. Normalement, le planning peut être mis à jour en temps réel, il est donc théoriquement juste mais si pour une raison quelconque il était erroné et non mis à jour, cela serait simplement indiqué dans le compte-rendu de la journée.

- Au niveau 3, il y a un descriptif de l'activité. Il faut pouvoir choisir dans la liste présentée figure 2. A cela, on doit ajouter les informations suivantes (à remplir par l'utilisateur si non disponibles) : numéro du jour, heure de début (précision de 10'), heure de fin (précision de 10'), la liste des astronautes concernés (on indiquera les noms possibles), le lieu, ainsi qu'un texte descriptif limité à 400 caractères. En ce qui concerne le lieu, on dispose d'une carte de la zone. Pour l'exemple fourni, il s'agit d'une image de taille 1095 par 2053 pixels représentant la région de Nanedi Vallis. L'habitat est situé à l'origine du repère (0,0) et correspond en pixels à la position (700, 1000) de l'image. Un lieu est défini par un nom et une position (x,y) exprimées en mètres relativement à l'habitat selon l'axe ouest-est (abscisses) et sud-nord (ordonnées). La résolution de l'image est de 5 mètres par pixel en première approximation selon les 2 axes. Si l'activité est en extérieur, l'utilisateur doit pouvoir entrer les coordonnées directement ou sélectionner sur la carte une position par simple clic. L'utilisateur doit pouvoir enregistrer la nouvelle activité dans le fichier des activités (voir plus loin) ou annuler, ce qui ramène à la fiche de niveau 2.

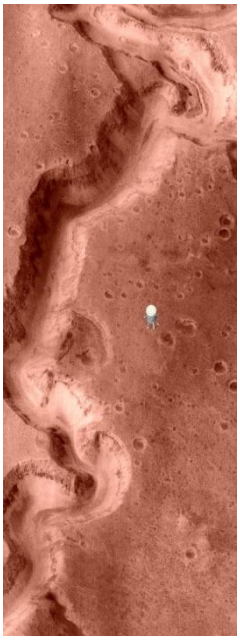

Figure 3 : Image de Nanedi Vallis

Analyse de l'activité : Depuis la fiche présentant le calendrier, il doit être possible d'analyser les activités passées. Pour cela, l'utilisateur doit pouvoir effectuer une recherche des jours concernés par une activité donnée. Par exemple, en entrant l'activité « Medical act », le programme doit renvoyer la liste des jours où un acte médical a été effectué. Il peut s'agir

également d'une activité générique, par exemple « Repair ». Il doit également être possible d'effectuer une recherche en fonction d'un mot présent dans le texte descriptif de l'activité ou du compte-rendu. Ensuite, l'activité doit pouvoir être sélectionnée pour visualiser les informations détaillées. Enfin, l'utilisateur doit pouvoir sélectionner une période pour affiner sa recherche.

En ce qui concerne l'activité d'exploration, l'analyse se fait avec la carte, éventuellement en utilisant une autre fiche (à vous de proposer la solution la plus ergonomique). L'utilisateur doit pouvoir visualiser tous les lieux où sont allés les astronautes, ainsi que les lieux qui sont programmés pour les jours suivants. Cela concerne l'exploration en scaphandre, avec un véhicule, ou une expérience en extérieur. On affichera sur la carte des icônes de différentes formes ou/et couleur pour représenter ces activités, en effectuant une distinction importante entre le passé et le futur. Ces icônes seront « cliquables », c'est-à-dire que sur le clic, on doit afficher les informations de l'activité, notamment la date et la description détaillée. Vous devez également proposer à l'utilisateur de sélectionner une période définie par une date de début et une date de fin (numéros de jour), afin que ne soient affichées que les activités en extérieur qui se sont déroulées pendant ladite période.

Fichiers : Des informations importantes seront stockées dans 2 fichiers XML distincts.

Le premier fichier concerne les informations générales, notamment le nom du fichier image servant de carte pour l'exploration, la position de l'habitat, ainsi que le nombre et le nom de chaque astronaute. Ce fichier inclut également la hiérarchie des activités présentée figure 2, qui pourrait éventuellement être modifiée ultérieurement, ainsi que la description des activités de la journée par défaut. A vous de définir la structure de ce fichier XML.

Le 2ème fichier XML concerne le détail des activités planifiées, passées et futures. Au lancement de l'application, ce fichier sera lu de manière automatique. La création, la modification ou la suppression se fait à partir de la fiche de niveau 3. Le nom de ce fichier n'est pas connu de l'utilisateur.

A rendre :

Vous rendrez un rapport comportant les chapitres suivants :

1. Gestion de projet

Vous indiquerez votre méthode de travail, les principaux jalons de votre gestion de projet ainsi que la répartition du travail. NB : Un prévisionnel pourra vous être demandé en début de projet.

2. UML

Vous expliquerez, au sein de cette partie, l'analyse que vous avez réalisée dans les premières phases de votre projet et qui a servi de base à l'architecture finale du logiciel. Pour ce faire, vous vous appuierez sur les diagrammes UML, statiques ou dynamiques, qui vous semblent les plus pertinents et significatifs dans le cadre de cette analyse.
Si certains de vos diagrammes ont beaucoup évolué entre les premières phases de l'analyse et la livraison finale, vous le ferez apparaître dans votre rapport et vous en expliquerez les raisons.
NB : vos diagrammes doivent être expliqués et justifiés.

3. Description du programme

3.1 Définition de chaque classe

Vous indiquerez le rôle de chaque classe.
Précisez dans quelles autres classes ou parties du programme sont déclarées des instances de la classe en question. S'il n'existe qu'une seule instanciation, précisez à quel moment elle est faite et par quelle méthode.
NB : La partie 2.1 peut éventuellement être écrite directement dans les fichiers sources Csharp, sous forme de commentaires clairs et précis. Le code devra de toute façon être bien documenté.

3.2. Fiches Form

Présentez chaque fiche par une image (éventuellement plusieurs si vous le jugez nécessaire) résultant d'une copie d'écran de la fiche en mode exécution et expliquez le rôle de chaque élément. Décrivez également les principaux événements associés à la fiche.

4. Tests unitaires et tests fonctionnels

4.1 Tests unitaires

Indiquez les tests unitaires que vous avez réalisés.

4.2 Tests fonctionnels

Vous proposerez dans un premier temps des protocoles de test. Exemples :

Test 1 : On crée une activité d'exploration avec scaphandre (Space suit) le 10ème jour, de 14h à 17h pour l'astronaute « John Glenn » aux coordonnées (500,600), avec la description suivante « Visite du site Rocky 4 », alors qu'aucune activité n'avait été programmée ce jour-là, l'activité « private », par défaut, ayant été supprimée au préalable. Après enregistrement, on quitte l'application. Puis on la relance. On vérifie que cette activité est bien présente pour le jour et l'heure indiquée.

Test 2 : On utilise le fichier x33 pour initialiser la liste d'activités (voir détail en annexe). L'utilisateur tente de visualiser sur la carte toutes les activités d'exploration entre le 20ème et 30ème jour. Théoriquement, on doit obtenir le résultat graphique de la figure 16. On vérifie que les 7 éléments sont bien présents aux endroits indiqués.

Test 3 : Le fichier UML des activités est déplacé. On lance le programme, qui doit nous avertir de l'absence du fichier des activités.

Etc etc.

A vous de proposer une batterie de tests pertinente qui permet de tester les parties les plus critiques.

4.3 Résultats des tests

Vous procéderez à chacun de ces tests de manière méthodique et indiquerez le résultat dans un tableau.[1]

NB : on ne demande pas de tests utilisateurs pour tester la convivialité.

5. Résultats

5.1 Bilan

Les fonctionnalités ont-elles été toutes prises en compte. Le programme fonctionne-t-il parfaitement ou subsiste-t-il des erreurs ? L'interface est-elle fonctionnelle et ergonomique ? Dressez un bilan du travail effectué.

5.2 Evolutivité

En quelle mesure peut-on faire évoluer le programme pour l'améliorer ?

Livrable

Vous déposerez sur Moodle (rubrique Génie Logiciel) ce rapport ainsi qu'une copie du répertoire complet contenant votre programme.

Les livrables doivent être constitués d'une **SEULE** archive nommée avec **les noms des membres du groupes**, contenant :

- la solution complète du projet (ainsi que les éventuels fichiers d'entrée/sortie),
- la documentation.

Date limite pour rendre le projet : le vendredi 18 décembre 2015, 23h59. Passée cette date, la plateforme de dépôt sera fermée et le dépôt impossible.

Critères d'évaluation

La note tiendra compte à la fois du rapport et du logiciel livré.

Pour le rapport :

- sur le fond : analyse UML, justification des choix, pertinence et validité des tests
- sur la forme : respect des consignes, présentation du programme, gestion de projet, clarté

Pour le logiciel :

- fonctionnement du programme, respect du cahier des charges, robustesse,
- qualité de la programmation, clarté des commentaires,
- convivialité de l'IHM.

[1] Si le résultat à un test est marqué positif alors qu'il s'avère qu'il y a bien une erreur, le projet sera sévèrement pénalisé