

les Cahiers du **Programmeur**

Swing

Java SE 5 • AWT/Swing • Java 3D • Java Web Start
SWT/JFace • JUnit • Abbot • Eclipse • CVS • UML • MVC • XP

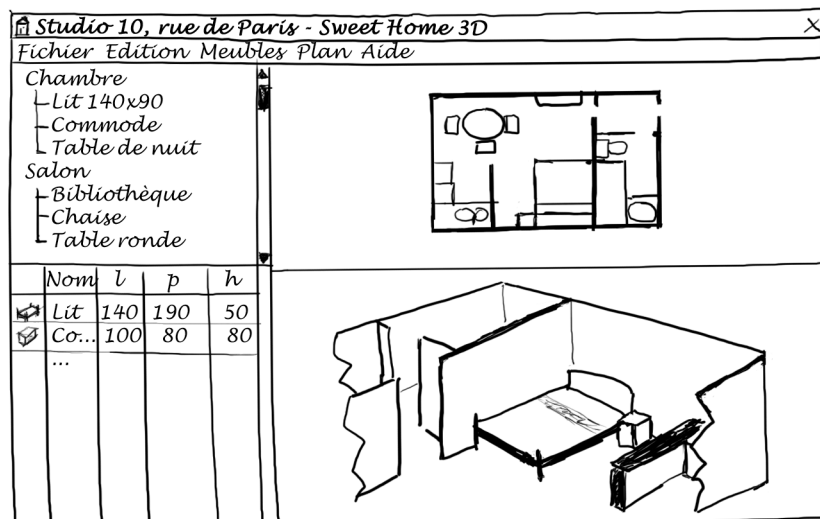
Emmanuel Puybaret



EYROLLES

1

chapitre



L'étude de cas : Sweet Home 3D

Ce premier chapitre présente le cahier des charges de l'application Sweet Home 3D, notre étude de cas. Y sont décrits les fonctionnalités, le mode de développement et de distribution de ce logiciel qui nous accompagnera tout au long de cet ouvrage, ainsi que les rôles des membres de l'équipe chargée de créer ce projet.

SOMMAIRE

- ▶ Sweet Home 3D
- ▶ Cahier des charges
- ▶ Spécifications générales
- ▶ Méthodologie XP
- ▶ Planification des scénarios

MOTS-CLÉS

- ▶ Open Source
- ▶ Cas d'utilisation
- ▶ eXtreme Programming

Les circonstances

Matthieu, Margaux, Thomas et Sophie sont des informaticiens convaincus de l'intérêt de l'Open Source. Particulièrement intéressés par le développement des interfaces utilisateurs graphiques et par l'ergonomie logicielle, ils décident de lancer un nouveau projet qui leur permettra de comprendre le fonctionnement de cet univers.

REGARD DU DÉVELOPPEUR **L'Open Source, nouvel eldorado de l'informatique ?**

La contribution au très riche gisement de l'Open Source peut se faire de différentes façons, depuis le volontariat dans ce qu'il a de gratuit voire d'idéaliste, au développement selon un modèle économique donné. Chaque individu ou société n'a pas les mêmes contraintes au même moment ; chacun peut tenter sa chance et trouver des contreparties économiques, publicitaires ou de réputation adaptées.

Sweet Home 3D

Sweet Home 3D est un logiciel d'agencement des meubles dans un logement. Il offre à l'utilisateur la possibilité de dessiner le plan de son logement, d'y disposer des meubles et de visualiser finalement le résultat en 3D.

B.A.-BA MVC

L'architecture MVC (Modèle Vue Contrôleur) sépare les classes d'une application dotée d'une interface utilisateur en 3 catégories :

- celles du *modèle*, qui structurent et gèrent les données du programme ;
- celles de la *vue*, qui affichent les données du modèle dans l'interface utilisateur ;
- celles du *contrôleur*, qui gèrent les actions de l'utilisateur sur la vue pour mettre à jour le modèle.

Ce type d'architecture permet notamment de réutiliser les classes du modèle dans des vues différentes.

REGARD DU DÉVELOPPEUR **Pourquoi choisir une telle étude de cas ?**

Bien qu'il existe de nombreux logiciels pour aménager les meubles dans un logement, ce type de programme a été retenu comme étude de cas pour les raisons suivantes :

- Il n'en existe apparemment aucun d'abouti en Open Source, et probablement pas écrit en langage Java.
- Il permet de se concentrer sur le développement d'une application côté client sans se soucier des aspects serveur.
- Ce type d'application permet de montrer comment créer un composant graphique dans lequel l'utilisateur dessine.
- Il permet de montrer un cas concret de mise en œuvre de l'architecture MVC en offrant notamment une vue 2D et une vue 3D d'un même modèle.
- L'organisation thématique des meubles d'un logement justifie la mise en œuvre des composants graphiques affichant des arbres et des tableaux. Comme ces types de composants sont généralement parmi les plus complexes des bibliothèques graphiques, ce sera l'occasion d'étudier leur utilisation.

Cahier des charges

L'utilisateur type du logiciel est une personne en cours de déménagement qui n'a que peu de temps à sa disposition. Tout sera donc mis en œuvre pour l'aider à dessiner le plus rapidement possible le plan de son logement et à y disposer ses meubles. Pour atteindre ce but, l'équipe décide des grandes fonctionnalités de l'application et des étapes que l'utilisateur devra suivre :

- 1 Choisir ses meubles dans un catalogue affiché dans une liste organisée par catégories (meubles de cuisine, de salle d'eau...), avec saisie de leurs dimensions si les valeurs proposées par défaut ne conviennent pas.
- 2 Dessiner en 2D les murs du plan de son logement réalisés à l'aide de la souris. L'utilisateur n'étant pas a priori architecte, des guides visuels, comme l'affichage d'une grille ou de certaines dimensions, devront l'aider à accélérer la saisie des pièces de son logement à partir d'un plan existant.
- 3 Placer ses meubles sur le plan de son logement, soit à partir de la liste des meubles constituée au cours de la première étape, soit à partir de la liste thématique des meubles. Chaque meuble pourra être déplacé ou pivoté autour de lui-même n'importe où dans les pièces du logement. Pour obtenir un rendu réaliste, sera affichée une vue en 3D reflétant l'aménagement des pièces avec leurs meubles. Elle sera également mise à jour à chaque manipulation dans la vue 2D du logement.
- 4 Enregistrer son travail dans un fichier afin d'y revenir plus tard. Les fichiers créés devront pouvoir être lus par n'importe quelle personne qui aura installé le logiciel sur son ordinateur.

Le logiciel disposera d'un catalogue de base de meubles, constitué de fichiers 3D existants et libres de droits. L'utilisateur pourra enrichir ce catalogue à partir de fichiers 3D qu'il créera avec le modèleur de son choix ou qu'il aura récupéré sur le Web.

L'application devra fonctionner au minimum sous Windows, GNU/Linux et Mac OS X. Son interface utilisateur sera disponible en français et en anglais, en fonction de la langue de l'utilisateur, avec la possibilité d'ajouter facilement le support d'autres langues ultérieurement.

Spécifications générales

Les spécifications générales sont synthétisées sous la forme du diagramme UML des cas d'utilisation de la figure 1-1 : ce schéma résume les actions que peut effectuer l'utilisateur du logiciel.

DANS LA VRAIE VIE

Choisir le nom d'une application

Le choix d'un nom pour un projet ou une application est toujours une épreuve. Faut-il un nom évocateur ou non la fonction du logiciel ou la technologie qu'il utilise (par exemple, *Home Planer* ou *JHome*) ? Est-ce préférable de choisir un nom en français ou en anglais pour espérer toucher un public plus large ? Les noms auxquels vous pensez sont-ils déjà utilisés ? Si oui, concernent-ils une activité informatique ? Es-ce que tel ou tel nom est une marque déposée ou un nom de domaine ? Voilà en tout cas un beau sujet pour un *brainstorming* ! Une fois que vous aurez déterminé une liste de noms, soumettez-la à votre entourage et recevez leurs critiques comme autant de bonnes idées à exploiter...

Pour en savoir plus sur le brainstorming :

- ▶ http://egov.wallonie.be/boite_outils_methodes/pa0305.htm

Pour vérifier l'existence d'une marque :

- ▶ <http://www.icimarques.com/>

B.A.-BA Modèleur 3D

Un modèleur est un logiciel qui permet de créer des formes en trois dimensions, soit à partir de formes simples (parallépipède, sphère...), soit à partir de courbes et de surfaces dessinées par l'utilisateur.

CONVENTIONS Linux et GNU/Linux

Il ne faut pas oublier que Linux représente uniquement le noyau du système d'exploitation GNU/Linux, bâti autour des outils du projet GNU. L'usage courant veut actuellement que Linux désigne ce système d'exploitation. Par commodité, nous adopterons cette convention dans la suite de l'ouvrage.

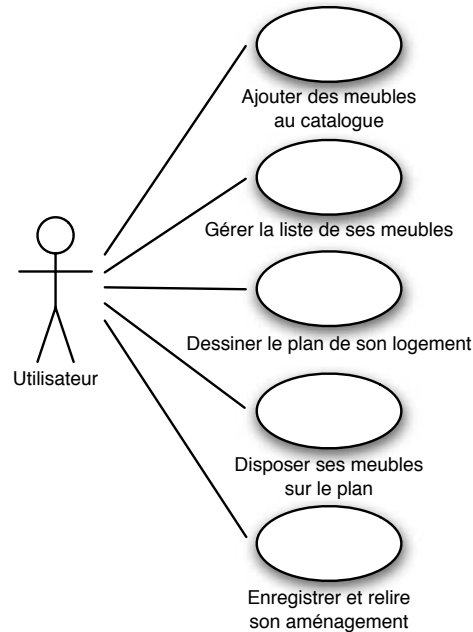


Figure 1-1
Cas d'utilisation de l'application

Maquette papier du logiciel

L'équipe dessine ensuite grossièrement sur papier une maquette du logiciel et aboutit à la figure 1-2. L'application se présente sous la forme d'une fenêtre, avec menus, découpée en 4 zones librement redimensionnables les unes par rapport aux autres :

- 1 En haut à gauche, le catalogue des meubles disponibles.
- 2 En bas à gauche, le tableau des meubles de l'utilisateur qu'il enrichit à partir de la liste précédente.
- 3 En haut à droite, le plan de son logement que l'utilisateur dessine et dans lequel il place ses meubles.
- 4 En bas à droite, une vue en trois dimensions du logement aménagé dans lequel l'utilisateur peut uniquement naviguer.

Cette présentation a pour avantage de présenter sur un seul écran, toutes les données manipulées par l'utilisateur ; aussi bien celles préexistantes comme le catalogue de meubles, que celles qu'il crée, comme la liste de ses meubles et le plan du logement. S'il a besoin d'agrandir une zone particulière, il lui suffira d'agir sur les barres de division qui séparent les quatre zones à l'écran ou d'utiliser le menu correspondant. Par exemple, s'il veut agrandir le plan du logement pour bénéficier de plus d'espace pendant la saisie du plan, il pourra temporairement faire disparaître les trois autres zones ou uniquement la zone de la vue en 3D.

ERGONOMIE **Wizard ou fenêtre avec menus**

Un *wizard* (quelque fois traduit par assistant ou expert) est un enchaînement de boîtes de dialogue auxquelles l'utilisateur répond étape par étape pour effectuer une tâche prédéfinie, comme pour le programme d'installation d'un logiciel. Ce système a pour avantage de montrer explicitement les différentes étapes à effectuer pour atteindre un but, mais n'est conseillé que pour réaliser des tâches simples et courtes que l'utilisateur doit réaliser dans un ordre déterminé. Comme le choix des meubles et le dessin du logement sont des opérations assez longues, nous choisirons une fenêtre avec des menus, ce qui permettra à l'utilisateur d'enregistrer son travail et d'y revenir plus tard, quelles que soient les actions déjà réalisées. Il faudra donc prévoir un système d'aide qui lui suggérera les différentes étapes à effectuer.

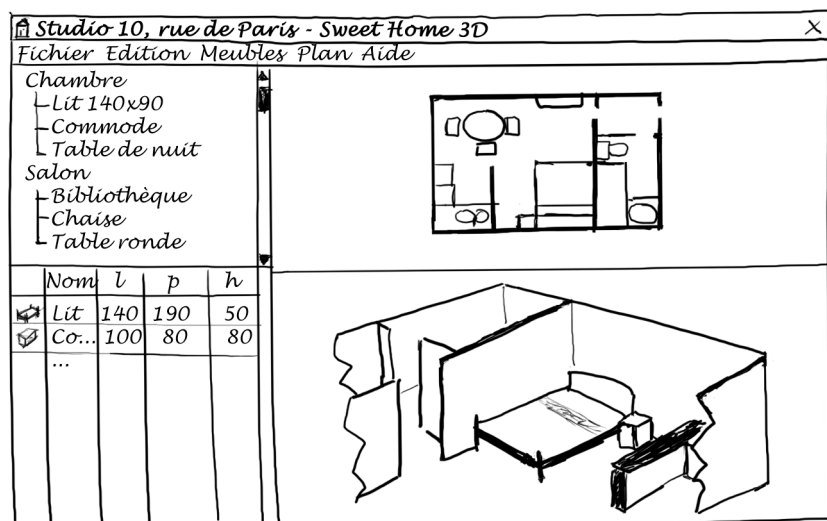


Figure 1-2
Maquette papier de l'application

Principaux menus de l'application

L'équipe complète la maquette précédente avec une première liste des premiers menus qui seront offerts par l'application.

| Menu (anglais / français) | Élément (anglais / français) | Description |
|---------------------------|--|---|
| File / Fichier | New / Nouveau | Création d'un nouveau logement. |
| | Open... / Ouvrir... | Ouvrir un logement / aménagement existant. |
| | Close / Fermer | Ferme le logement ouvert. |
| | Save / Enregistrer | Enregistre le logement en cours d'édition avec l'aménagement de ses meubles. |
| | Save as... / Enregistrer sous... | |
| | Preferences / Préférences... | Modifie les préférences (catalogue des meubles, unité de longueur...). |
| | Exit / Quitter | Quitte l'application |
| Edit / Édition | Undo / Annuler | Annule / Refait la dernière opération effectuée sur le logement en cours d'édition. |
| | Redo / Refaire | |
| | Cut / Couper | Coupe / Copie / Colle la sélection courante. |
| | Copy / Copier | |
| | Paste / Coller | |
| | Delete / Supprimer | Supprime la sélection en cours. |
| Furniture / Meubles | Add / Ajouter | Ajoute le meuble sélectionné dans le catalogue à la liste des meubles du logement. |
| Plan / Plan | Create walls / Créer les murs | Passes en mode création de murs. |
| | Import image... / Importer image... | Importe l'image affichée sous le plan. |
| Help / Aide | Sweet Home 3D help / Aide de Sweet Home 3D | Lance les pages d'aide. |
| | About... / À propos de... | Boîte de dialogue des copyrights. |

ERGONOMIE Critères d'ergonomie d'une interface utilisateur

Le document mentionné ci-dessous regroupe en détail les différents critères d'ergonomie qui doivent guider la conception d'une interface utilisateur. Il est en anglais, suivi d'une version en français.

▶ <ftp://ftp.inria.fr/INRIA/publication/publi-pdf/RT/RT-0156.pdf>

Le choix entre les bibliothèques graphiques Java Swing et SWT, sera décidé suite à une évaluation des avantages et des inconvénients de ces deux outils (voir le chapitre 3, « Choix techniques : Swing ou SWT »).

B.A.-BA Java Web Start

Java Web Start offre un mode de diffusion qui simplifie l'installation et les mises à jour d'applications Java sur des postes client. Cette technologie se met simplement en place en diffusant sur un site web un fichier d'extension `jnlp` qui décrit en XML les composants nécessaires à l'application.

▶ <http://java.sun.com/products/javawebstart/>

Intégration du logiciel dans le système d'exploitation

Plutôt que d'offrir de nombreuses fonctionnalités peu utilisées, l'équipe de développement propose de mettre surtout l'accent sur l'ergonomie du logiciel. Par ailleurs, un utilisateur apprécie surtout un logiciel pour les fonctionnalités qu'il offre et son intégration dans le système. Il ne se soucie guère du fait que l'application qu'il utilise soit disponible sous plusieurs systèmes, et n'excusera pas les défauts d'un programme pour la simple raison que, développé en Java, il soit portable et donc disponible sous un autre système. L'équipe s'attachera donc à assurer la portabilité de son programme mais aussi son intégration correcte dans le système. Ils devront par exemple prendre en compte le fait que la barre de menus sous Mac OS X est toujours en haut de l'écran, ou que la plupart des utilisateurs Windows s'attendent à pouvoir naviguer dans les menus d'une application uniquement au clavier.

Choix du langage

Il existe très peu d'outils pour développer facilement une application graphique portable recourant à de la 3D : on peut citer Java et REALbasic[®], pour les plus connus. Les membres de l'équipe choisissent Java pour sa maturité, la richesse de sa bibliothèque et la gratuité de ses outils de développement. Ce choix leur permettra aussi d'améliorer leurs compétences sur un des langages de programmation les plus utilisés sur le marché.

Distribution

Afin de faciliter son installation et sa diffusion, Sweet Home 3D sera mis à disposition sur un site web sous forme d'une application Java Web Start.

Méthodologie XP

Trop souvent, de bonnes idées de logiciels n'aboutissent pas, du fait d'une mauvaise organisation. Pour améliorer leurs chances de réussite, l'équipe choisit la méthode de développement de logiciels XP (eXtreme Programming). Cette dernière leur semble mieux adaptée à leur projet qu'une organisation qui débute par une phase d'analyse suivie d'une phase de programmation, et ce pour les raisons suivantes :

- XP permet de produire des logiciels plus fiables, notamment grâce aux tests unitaires.
- Le développement par itération, prôné par XP, permet d'obtenir régulièrement des versions fonctionnelles.

- Les rôles XP attribués aux membres d'une équipe favorisent la communication entre eux et leur permettent d'être occupés dès le démarrage du projet.
- XP semble bien adapté aux projets basés sur le bénévolat qui, par essence, ne permet pas de garantir une disponibilité à 100 % des membres de l'équipe.

B.A.-BA eXtreme Programming

XP est une méthodologie de développement de logiciels surtout connue pour sa notion de tests unitaires. Ces derniers sont des programmes simples, développés pour tester les fonctionnalités d'une classe ou d'un programme. Lancés automatiquement de façon régulière, ils évitent à un logiciel de subir d'éventuelles régressions au cours de son évolution.

Mais XP propose en fait d'aller plus loin, notamment en prônant le développement d'un logiciel par itérations successives. Chaque itération débute par l'écriture d'un scénario qui décrit la fonctionnalité ajoutée. Ce scénario est traduit ensuite sous la forme d'un programme de test qui fait appel à des classes et des méthodes développées en dernier. Ce mode de programmation assure que les programmes de tests sont développés systématiquement et minimise le développement de fonctionnalités non demandées dans les classes. Au cours du développement, les programmeurs doivent organiser aussi des séances de revue de code qui aboutissent, si nécessaire, à un *refactoring* des classes dans le but d'en améliorer la qualité (par exemple, en factorisant du code dupliqué dans des méthodes ou en introduisant des super-classes regroupant des concepts communs à plusieurs classes).

En ce qui concerne l'organisation d'une équipe, XP définit six rôles différents qui, idéalement, doivent être assurés par des membres différents. Au rôle de programmeur, assuré par binômes, s'ajoutent les rôles de client, de testeur, de tracker, de manager et de coach :

- Le *client* rédige et ordonne les scénarios et les tests de recette prévus à chaque itération.
- Le *testeur* met en place les tests et les outils qui s'y rattachent.
- Le *tracker* suit le travail des programmeurs, afin de prendre les dispositions nécessaires au plus tôt en cas de difficulté.
- Le *manager* s'occupe de l'organisation humaine et matérielle de l'équipe.
- Le *coach* coordonne et aide les membres de l'équipe tout en veillant à la mise en place correcte de la méthode XP.

Enfin, XP propose d'organiser régulièrement des réunions de planification et d'avancement pour favoriser l'échange d'informations. Le développement par binômes de programmeurs et les rencontres fréquentes entre membres d'une équipe qui applique la méthode XP implique par conséquent que ceux-ci doivent travailler dans un même lieu, ouvert si possible. Cette contrainte ne pourra pas s'appliquer à l'équipe de Sweet Home 3D dont les membres sont répartis sur plusieurs sites, mais ceux-ci feront de leur mieux pour communiquer le plus souvent possible entre eux, par exemple grâce à des outils de messagerie instantanée.

📖 *L'eXtreme Programming* - J.-L. Bénard, L. Bossavit, R. Médina et D. Williams, Eyrolles 2002

Répartition des rôles de l'équipe

Les quatre personnes de l'équipe se répartissent les rôles comme décrit dans le tableau suivant.

Tableau 1-1 Rôles des membres de l'équipe

| | Rôle XP | Mission particulière |
|----------|-----------------------|---|
| Matthieu | Manager / Coach | Choisit les outils de développement et le référentiel. Supervise et conseille les développeurs pour la conception du logiciel. Gère la communication entre les membres de l'équipe et avec l'extérieur. |
| Margaux | Programmeur / Tracker | Développe les classes liées à l'interface utilisateur. Gère la configuration du référentiel et des versions. |
| Thomas | Programmeur / Tracker | Développe les classes du modèle et les tests automatisés. |
| Sophie | Client / Testeur | Prend en charge l'infographie 2D et 3D du logiciel, rédige les pages d'aide. Rédige les scénarios des itérations. Teste et fait tester l'application par des bêta testeurs. |

B.A.-BA Référentiel

À la base, un référentiel (*repository*) est un outil permettant d'archiver et de partager les fichiers d'un projet entre les membres d'une équipe. Par extension, les référentiels de projets sur Internet proposent aussi d'autres outils nécessaires à la gestion d'un projet : outils de diffusion (documents présentant le projet, flux RSS, gestion des téléchargements), de suivi de projet (mailing list, forums, gestion de bogues)...

Cet ouvrage traite uniquement les scénarios 1 à 10 : ceux-ci couvrent la plus grande partie des besoins des développeurs pour créer une application dotée d'une interface graphique.

Planification des scénarios

Sophie et Matthieu décomposent les fonctionnalités du logiciel décrites dans le cahier des charges et les cas d'utilisation, puis définissent les titres des scénarios associés à chaque itération du développement. Ils attribuent ensuite à chacun d'eux un numéro d'ordre, en tenant d'abord compte de leur importance pour obtenir le plus vite possible une version fonctionnelle de Sweet Home 3D, puis de l'intérêt de la fonctionnalité qui leur est associée dans la version finale du logiciel.

À RETENIR Découper un projet en scénarios

Voici quelques critères qui vous aideront à découper et ordonner en scénarios les fonctionnalités à développer dans un logiciel :

- Difficulté : si la conception d'une fonctionnalité est complexe ou nécessite une mise à niveau importante des connaissances, isolez-la dans un scénario séparé autant que possible, pour ne pas augmenter la durée et la difficulté de réalisation du scénario.
- Intérêt : intégrez dans les scénarios les tâches moins intéressantes comme la localisation.
- Architecture : prévoyez dans les premiers scénarios les fonctionnalités qui ont une influence importante sur l'architecture du logiciel.
- Nombre de programmeurs : si l'équipe est composée de plusieurs binômes de programmeurs, créez si possible des scénarios qui peuvent être développés en parallèle.
- Compétences : répartissez les tâches dans les scénarios en fonction des compétences disponibles.
- Dépendance : programmez en premier les scénarios dont la fonctionnalité est nécessaire pour débiter un autre.
- Degré d'importance : reléguez dans les dernières itérations les fonctionnalités qui ne permettent pas d'obtenir le plus vite possible une version fonctionnelle du logiciel.

Bien que ce ne soit pas généralement le cas dans un vrai projet, le critère pédagogique peut rentrer aussi en compte, ce qui est bien entendu le cas pour le développement de l'application développée dans cet ouvrage.

D'après ce tableau, l'équipe devrait obtenir une application en grande partie fonctionnelle à la dixième itération.

Tableau 1–2 Scénarios de développement

| Titre du scénario | Itération | Localisation | Annuler / refaire |
|--|-----------|--------------|-------------------|
| Création de l'arbre du catalogue des meubles | 1 | x | |
| Ajout / suppression de meubles au catalogue | 17 | x | |
| Création du tableau des meubles du logement | 2 | x | |
| Ajout / suppression des meubles du logement | 3 | | |
| Mise en place des opérations Annuler / refaire | 4 | x | x |
| Tri des meubles du logement | 11 | x | |
| Modification du nom, des dimensions et de la couleur des meubles | 12 | | x |
| Ajout / suppression des meubles par copier-coller et glisser-déposer | 9 | | x |
| Dessin du plan du logement | 5 | x | x |
| Modification de l'épaisseur et de la couleur des murs | 13 | x | x |
| Choix de l'image affichée sous le plan | 15 | x | x |
| Gestion du zoom et des guides visuels pour le dessin | 14 | x | |
| Disposition des meubles dans le plan | 6 | | x |
| Vue 3D de l'aménagement | 7 | | |
| Déplacement de l'observateur dans la vue 3D | 16 | x | |
| Enregistrement / lecture d'un aménagement | 8 | x | |
| Modification des préférences utilisateur | 10 | x | |
| Intégration des pages d'aide | 18 | x | |

Par ailleurs, Matthieu a inclus dans ce tableau les tâches transversales de gestion de la localisation et des opérations annulables. Si ces tâches sont effectuées tardivement, il craint qu'elles n'entraînent la modification de trop nombreuses classes et provoquent ainsi des régressions. Comme ces deux tâches sont assez répétitives, il conseille les développeurs de les intégrer au fur et à mesure à chaque itération quand cela est nécessaire. Chaque fois que les tests d'un scénario x seront validés, le logiciel sera marqué d'un numéro de version $0.x$ avant de passer aux développements du scénario suivant.

En résumé...

Ce premier chapitre nous a permis de présenter Sweet Home 3D, l'étude de cas étudiée dans cet ouvrage, ainsi que l'équipe qui prend en charge le développement de ce projet. Ce type d'application nous permettra tout au long de ces pages d'aborder la plupart des composants graphiques Swing disponibles, des plus simples aux plus complexes, la création d'un composant de toute pièce et l'assemblage de ces composants pour former l'interface utilisateur de l'application.

B.A.-BA Localisation et internationalisation

La localisation et l'internationalisation (aussi désignées respectivement par les sigles L10N et I18N) consistent à adapter l'interface utilisateur d'un logiciel en fonction de sa langue ou de son pays. La bibliothèque standard Java dispose de classes qui facilitent grandement la mise en place d'un tel système.

► <http://java.sun.com/j2se/corejava/intl/reference/faqs/>