



Description détaillée de Azur

1. Historique

À la base, le but du logiciel était de permettre la conversion entre les dates (et donc les calendriers) de différentes planètes, par exemple pour connaître à quoi correspond une certaine durée sur une autre planète, et donc comparer les changements de saisons. Puis, afin d'éviter les calculs « à la main » des paramètres astrophysiques nécessaires, a suivi un simulateur de systèmes planétaires.

Finalement, comme j'utilisais aussi à côté un wiki afin de collecter toutes les informations sur les personnages, les objets, l'histoire des mes romans, je me suis dit qu'il pourrait être intéressant de rassembler tous les outils en un seul, et de créer un logiciel unifié.

Ainsi, bien que le logiciel était clairement orienté vers la science-fiction et la fantasy à la base, sa modularité lui permettait d'être utilisé pour une certaine variété de projets : romans de tout genre, études littéraires (*La Comédie Humaine* de Balzac, *le Seigneur des Anneaux* de Tolkien, *les Rougon-Macquart* de Zola...), univers de jeux de rôles...

Azur est basé sur la notion de *modules* : chaque module a un objectif propre et est presque indépendant des autres. Ce système permet d'étendre facilement les possibilités du logiciel, tout en ne s'encombrant pas de fonctions qui ne seront pas utiles pour le projet. Une partie des modules concerne la description de l'univers (terme générique), tandis que l'autre facilite surtout la tâche de l'écrivain (*a fortiori* s'il s'agit d'un travail de groupe), tant sur le plan de l'écriture que de la communication.

2. Principales fonctionnalités : Univers

Le premier et principal module a trait aux *personnages* : il est possible de créer une fiche pour chaque personnage, afin de détailler ses portraits physique et moral, son histoire... Deux types de *relations* peuvent être mises en place : la *famille* et les *autres relations*. Le premier type permet ainsi de retracer la généalogie du personnage, tandis que le second permet de garder trace de ses amis, de ses ennemis, de ses professeurs, etc. La notion de *groupe* permet d'indiquer de simplifier les réseaux de relations, en indiquant par exemple quelle école ou quelle secte a fréquenté le personnage.

Les *événements* permettent d'ajouter une dimension temporelle aux autres modules, et permettent de donner une dimension temporelle à la plupart des autres objets. Une grande flexibilité est permise, puisque chaque événement peut posséder plusieurs sous-événements (par exemple des batailles dans une guerre) ou être terminé par un autre (traité de paix pour continuer l'exemple). L'utilisateur peut créer ses propres types d'événements ; ceux touchant à la vie (naissance, mort et union) sont gérés un peu différemment. Il est ainsi possible de déterminer quand commence une relation ou encore quand elle finit... Les dates en elle-même et leur affichage sont entièrement

gérées par un module dédié aux *calendriers*.

Un module est consacré aux *lieux* et à la *géographie*, afin de déterminer où vivent les personnages, quelle distance sépare deux endroits... Il sera possible d'afficher des cartes, à deux et trois dimensions de la planète à partir des données saisies.

Un *simulateur de systèmes planétaires* prend en charge tout ce qui a trait aux phénomènes célestes : température et habitabilité d'une planète, passage des saisons, observation du ciel (couleur du ciel, sphère céleste, etc.), marées... Il fournit aussi les données nécessaires aux modules de dates et de lieux.

Tous les modules comportent de nombreux outils de contrôle afin de vérifier la *cohérence* de l'univers ainsi créé. Par exemple, on sera averti si l'on tente d'ajouter un enfant à une femme qui est déjà morte, de marier deux personnages qui ont vécu à des époques différentes, ou encore de faire voyager trop rapidement un personnage entre deux événements.

De même, une place importante est accordée à la représentation du monde tel qu'il perçu par les personnages : par exemple, la pertinence du choix des unités et du format de la date se fera selon le *contexte* des informations consultées (époque, lieu...).

3. Principales fonctionnalités : Conception

Les auteurs peut remplir une *liste des tâches et d'idées*, ou rédiger des *notes personnelles*. En fonction du type de fichier utilisé, un *gestionnaire de source* permet de suivre l'évolution du texte, et donc de retrouver une ancienne version ou de le partager plus facilement entre plusieurs co-auteurs. Dans ce cas, les textes pourront être analysés afin d'améliorer le style et de faire certaines statistiques. Finalement, il sera possible de travailler simultanément à plusieurs sur un même fichier.

De nombreux modules permettent une bonne interaction avec la communauté qui évolue autour du projet : *informations* (blog), *sondages*, *forum*, *salons de discussion*...

4. Extensions

Il est assez aisé de créer de nouveaux modules pour étendre les fonctionnalités du logiciel pour des projets plus spécifiques. Voici quelques outils et concepts supplémentaires qui seront disponibles par défaut :

- linguistique et générateurs de mots, afin de permettre la création de langues avec une certaine consistance ;
- races et d'états spéciaux (vampire, loup-garou...) pour la fantasy ;
- gestion des emplois, des travaux et des statuts.

5. Détails techniques

Le logiciel utilise le langage Python et son framework web Django. Ainsi il se présente sous la forme d'un site internet, cadre naturel pour un travail collaboratif.

À terme, un logiciel client devrait être développé, afin d'administrer et de consulter les données plus rapidement et directement.

Azur repose aussi sur le projet *sympy* (module de calcul formel) pour les simulations de systèmes planétaires, la géodésie et la gestion des unités. *Gobby* permettra l'édition simultanée d'un fichier par plusieurs personnes et *Jabber* servira de logiciel de chat.

6. Sites internet

- Présentation en français : <http://harold.e.free.fr/index.php?page=azur>
- Documentation : <http://harold.e.free.fr/azur/>
- Projet sur Bitbucket : <http://bitbucket.org/Melsophos/azur/>