



# ULTIMAADE

## PROJET ULTIMAADE Bilan Architecture Global

Date: 10/09/2011  
Version: 1.0

## Objectif résumé du document

ULTIMAAGE est une application open source et multiplateformes, de création multimédia. Elle permet à tout utilisateur à partir d'une idée simple, de définir un sujet parmi de nombreux modèles prédéfinis dans une bibliothèque. L'application permet ensuite à l'utilisateur grâce à de nombreux outils, de transformer le modèle choisi en une application multimédia complète, autonome et multiplateformes.

Ce document constitue la première étape de conception du projet, dans laquelle seront posés les fondements du système.

Les objectifs de bases visés pour le système sont :

- la modularité
- la maintenabilité
- La simplicité

Nous commencerons par une capture des besoins qui ont été définis dans le document d'étude détaillée, puis nous entrerons dans la phase d'analyse et conception architecturale.

Ceci nous permettra de construire un pont entre la description purement fonctionnelle et générique des besoins, et la réalisation.

Le présent document est à sa première version, il est appelé à changer à mesure de l'avancement de la conception de l'application. Cette phase est la plus déterminante de la réalisation du projet, et se fera en plusieurs phases intermédiaires. De ce fait, dans cette version du document, nous présenterons une conception beaucoup plus modulaire de l'application. Cette conception suit un modèle objet, et le détail des méthodes spécifiques permettant d'agir sur chaque objet constituera la prochaine phase de notre conception. En termes techniques on parlera des fonctions membres des classes implémentant les modules du système.

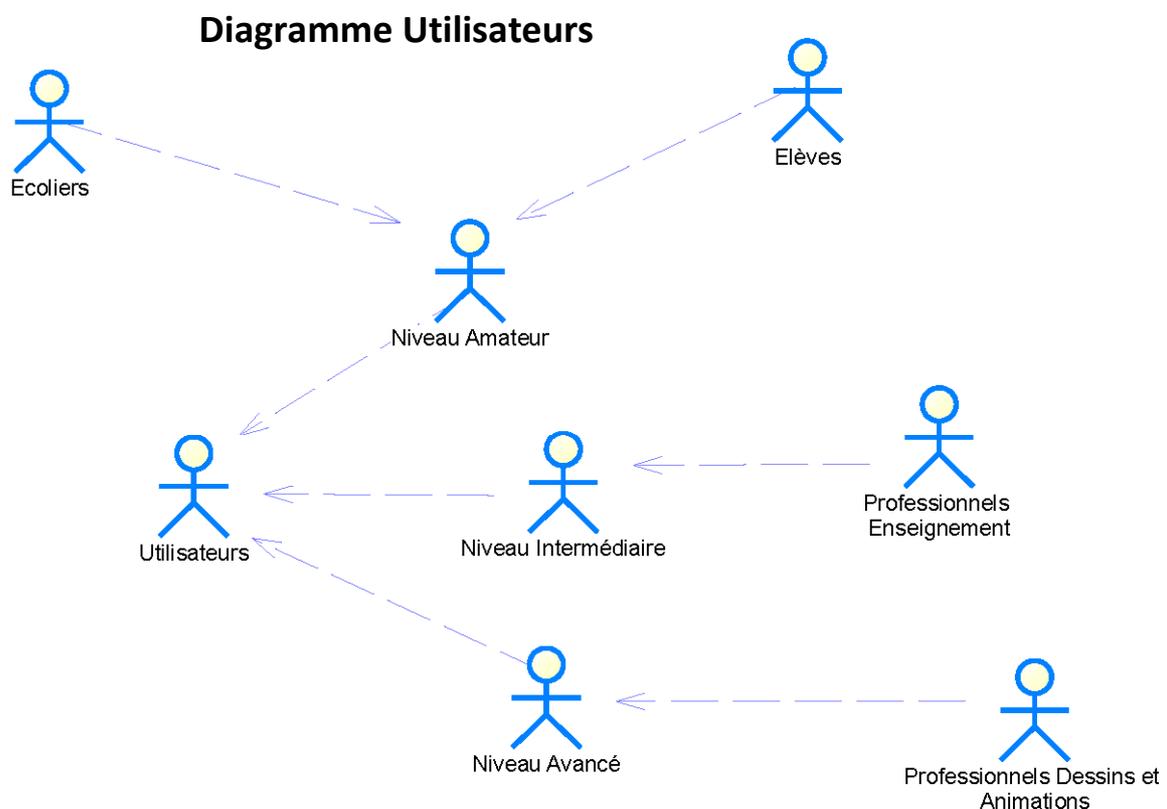
### HISTORIQUE DES REVISIONS

Réf	Objet	Date	Auteurs	Vérificateur	Approbateur
V 1.0	Première phase de conception	10/09/2011	Antoine PIELLARD, Adrien BARRAU, Alexandre DA SILVA, Abdel Moxine SOUMANOU	Eugène NGONTANG	Eugène NGONTANG

## ➤ Identification des acteurs

Nous définissons ici les acteurs qui vont interagir avec le système. Nous entendons par acteur, un humain, une machine, ou un système qui ne fait pas parti de la solution à réaliser mais qui participe au fonctionnement général de la solution par une interaction. Cependant dans notre cas nous avons typiquement des acteurs humains qui pourront être des écoliers/élèves, des enseignants, des professionnels de bandes dessinées, ou tout simplement un utilisateur désirant créer des animations, des enchainements de scènes, chronologiques et interactives. Les acteurs peuvent être regroupés ainsi :

- les plus petits : peuvent créer des albums du style «tourne-page», à branchements multiples
- les enseignants : découverte géographique, scénarios de pédagogie embarqués, enseignement assisté par ordinateur, etc
- les professionnels de bandes dessinées et animation: éditer, importer leurs dessins, puis les animer afin de réaliser un film interactif à partir de plusieurs scènes.
- tout autre type d'utilisateurs: développer une application multimédia, créer des animations, présentations, exporter un projet dans un format de son choix parmi les formats supportés par Ultimaade, etc



### ➤ **Identification des cas d'utilisations**

Après avoir identifié ci-dessus tous les acteurs qui vont interagir avec le système, nous passons à la collecte des besoins fonctionnels. Cette étape est réalisée par une analyse acteur par acteur, afin de vérifier pour chacun qu'il dispose de toutes les fonctionnalités qui lui seront utiles au regard de sa catégorie et du périmètre du projet.

Pour cela, nous avons décomposé les fonctionnalités requises pour notre application en trois catégories d'utilisateurs, à savoir :

- Les amateurs, qui pour nous représentent principalement les élèves du primaire et du secondaire (les plus petits). Ces derniers auront à leur disposition tous les outils leur permettant de créer une histoire chronologique à partir des personnages de leur univers. L'application sera livrée avec une bibliothèque de sprites représentant des exemples de personnages pour cette catégorie.
- Les utilisateurs de niveau intermédiaire : nous pouvons citer dans cette catégorie les enseignants, mais elle regroupe tous les professionnels ou adultes non informaticiens à la base, mais qui se servent de l'informatique au quotidien dans la réalisation de leur projets. Ceux-ci pourront déjà utiliser le logiciel au même niveau que les plus petits, et pourrons réaliser des choses plus intéressantes telles que des montages vidéo, des visites virtuelles.
- Les utilisateurs de niveau avancé, que nous appelons les initiés. Il s'agit ici de personnes ayant une connaissance du monde du jeu vidéo, et capable d'imaginer des scénarios de jeux beaucoup plus complexes. En plus des possibilités offertes aux autres catégories, ceux-ci pourront créer des scènes et histoires interactives, des mini jeux vidéo.

Ci-dessous un diagramme des cas d'utilisation, illustrant mieux l'aspect fonctionnel du projet.



### ➤ **Conception IHM (Interface Homme-Machine)**

La définition des besoins IHM est liée au contexte technique de notre projet, car toutes les fonctionnalités offertes par l'application ne seront accessibles que via son interface graphique.

Nous commençons immédiatement par cette phase après l'identification des cas d'utilisation, car dans l'analyse dynamique qui va suivre avec les diagrammes de séquences il sera très important de connaître au maximum le fonctionnement de notre future IHM.

A cet effet nous avons réalisé une maquette IHM qui sera elle aussi peaufinée lors des prochaines étapes de notre conception. Elle n'est pas le reflet exact de l'interface graphique finale de notre application, mais pourrait déjà donner à un utilisateur un aperçu de la navigation au sein de l'application.

Le document de conception IHM est inséré en annexe.

### ➤ **Les diagrammes des séquences (Chronologie des traitements)**

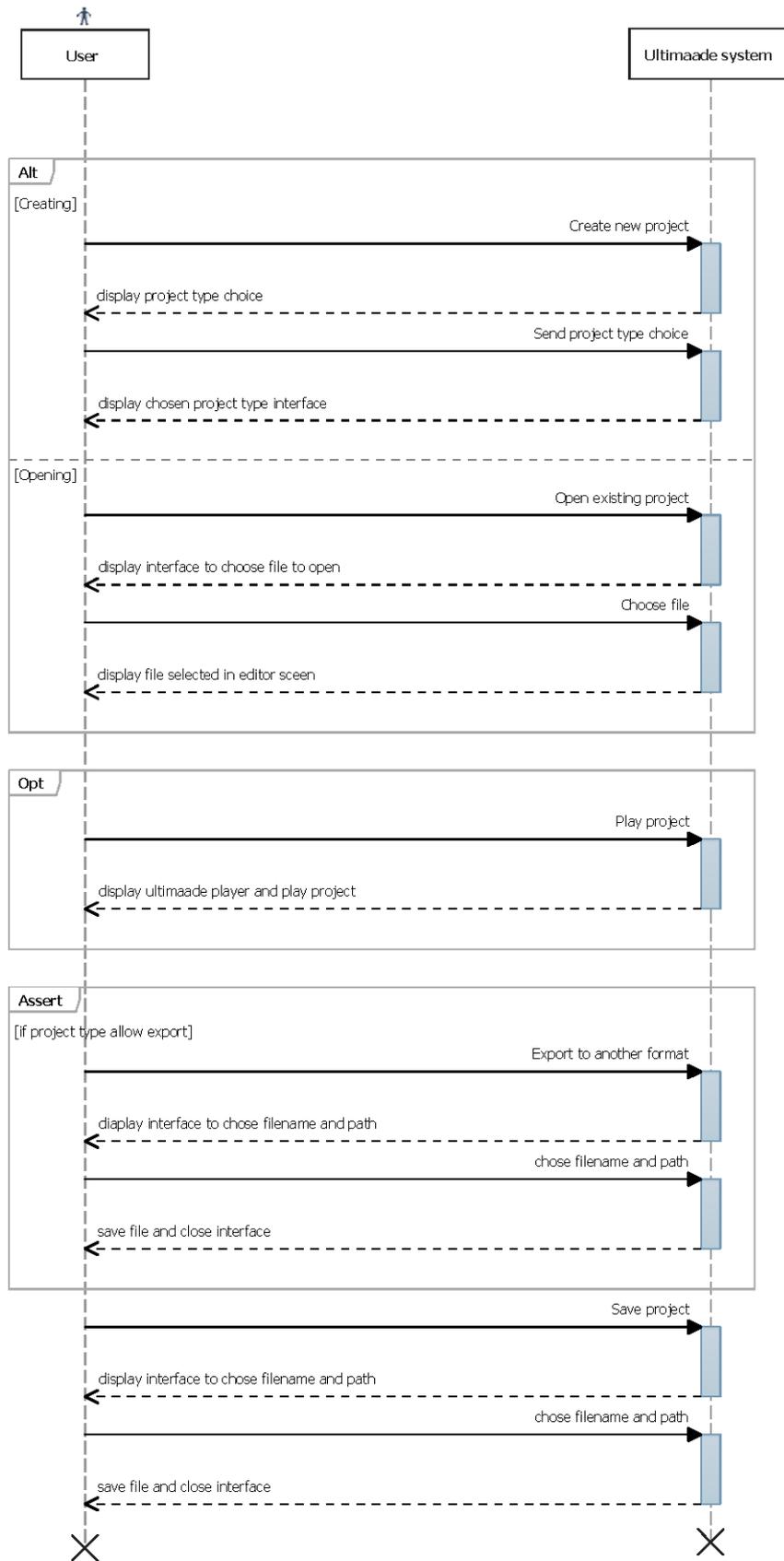
Dans ce paragraphe nous décrivons le processus d'utilisation de l'application.

Nous utilisons pour les diagrammes de séquence l'acteur (petit bonhomme), qui est un acteur générique pour décrire le fait que n'importe quel utilisateur peut utiliser le logiciel. Nous utilisons également le système (le petit rectangle), interfacé par l'IHM.

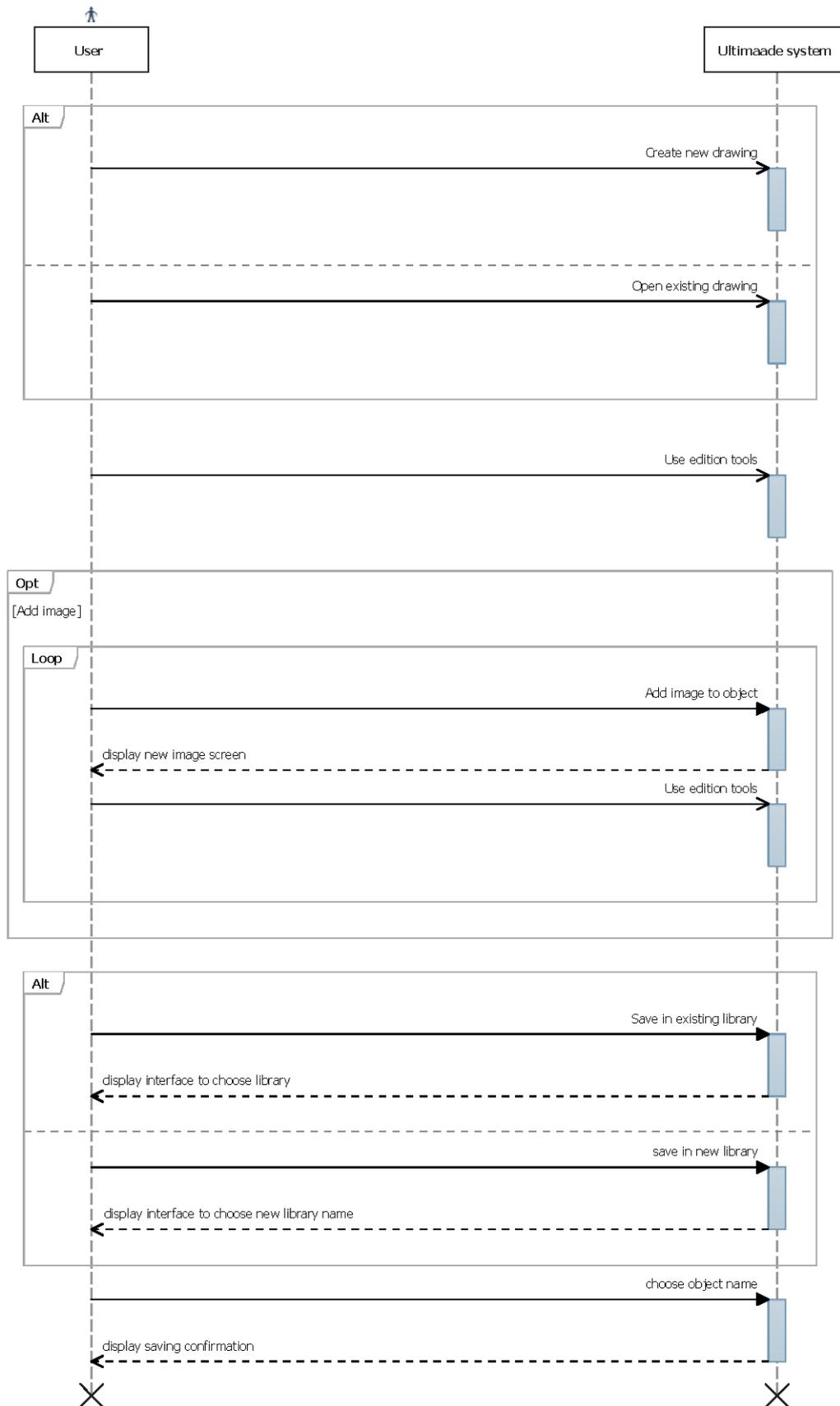
Nous avons pour cela décrit des diagrammes de séquence systèmes correspondant aux caractéristiques principales de l'application telles que décrites dans le cahier de charges.

La version actuelle du document contient en plus du diagramme de séquence système global, les diagrammes de séquence de dessin et d'édition de scène ci-dessous. Les versions futures du document de conception intégreront les diagrammes des séquences des autres cas d'utilisation.

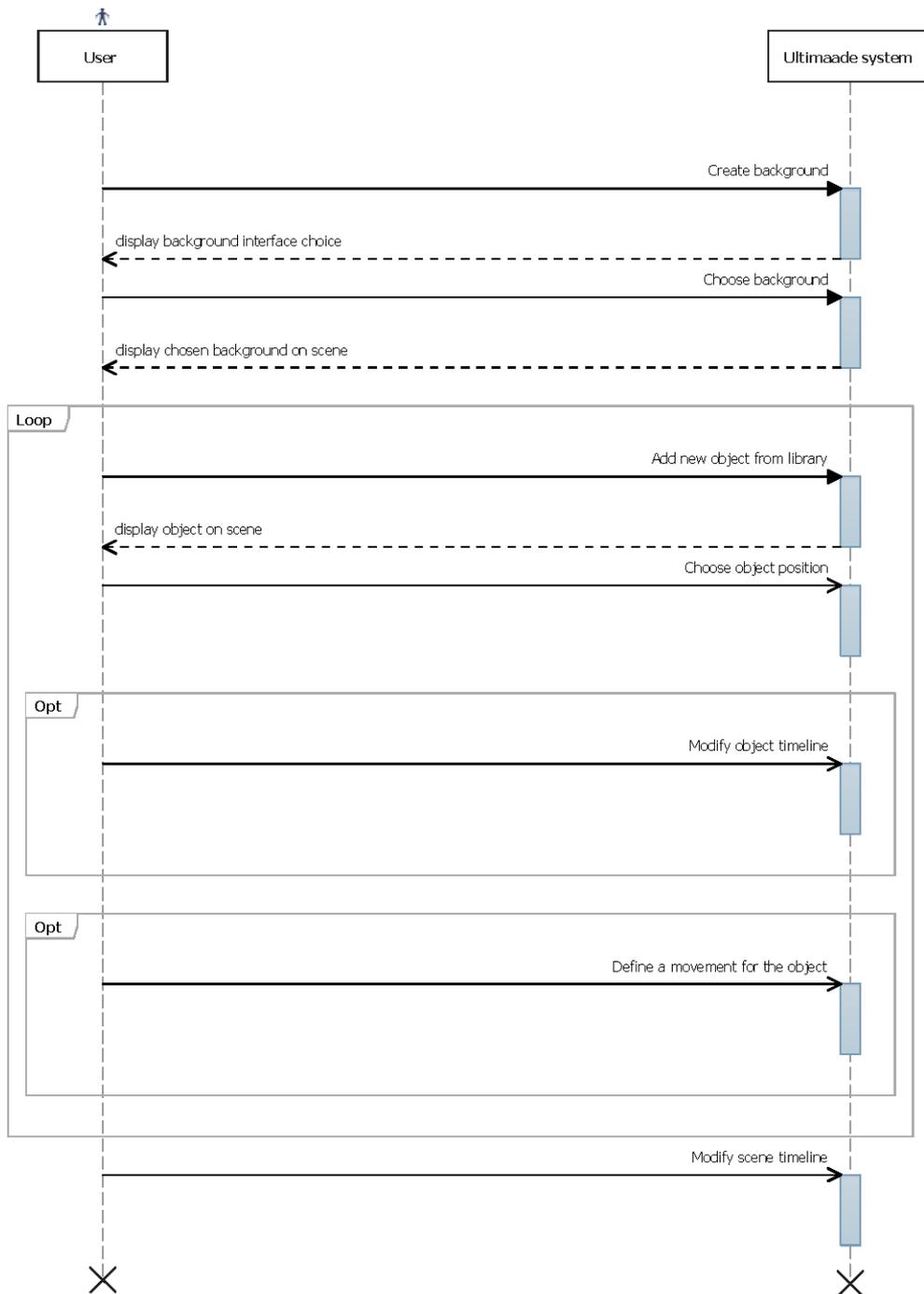
## Diagramme des séquences global



## Diagramme des séquences de dessin



## Diagramme des séquences d'édition de scène



➤ **Les diagrammes d'activités (Flux de données/ressources)**

Nous avons choisi de mettre d'abord cette partie entre parenthèse, car elle découlera directement des diagrammes des séquences dans notre conception. Cette étape étant inachevée pour le moment.

Elle permettra de documenter le fonctionnement de chaque cas d'utilisation et de

mettre en évidence les flots d'information dans le système à développer.

➤ **Les diagrammes d'état (Comportement d'un cas d'utilisation)**

Comme pour la phase précédente, celle-ci sera effectivement abordée dans la suite de notre conception. Elle définira la gestion des événements de façon plus claire, afin de mettre en évidence le traitement des signaux asynchrones que pourra recevoir le système.

➤ **Analyse et conception architecturale : Les classes candidates.**

Dans cette partie nous allons utiliser les différents diagrammes de la phase de capture des besoins et proposer un ensemble de classes qui permettront de matérialiser informatiquement chacune des entités identifiées durant notre capture des besoins.

D'un point de vue beaucoup plus global, l'application sera divisée en trois grosses parties (packages) dans lesquelles seront répartis les différents modules.

Ces trois tiers seront constitués de :

- Une IHM, qui représente le front-end (first tier) du système. C'est la partie interface utilisateur.
- Un Cœur (Core), qui représente le middle tier, et permettra l'interaction des différents plugins avec la GUI. C'est dans cette partie que sera définie l'interface de programmation des plugins Utilimaade.
- Un back-end, qui constitue tout l'environnement d'implémentation de l'API des plugins. Un développeur qui voudra participer au projet en développant son propre plugin Ultimaade travaillera à ce niveau.

A ces trois tiers s'ajoutera un module audio spécifique(Player) que sera livré en tant que lecteur Ultimaade.

De cette manière l'architecture de l'application doit être définie de telle sorte que la modification ou le développement d'un plugin soit essentiellement locale au back-end.

Les changements seront automatiquement reportés à l'interface graphique pour l'utilisateur, sans aucune modification sur le code de cette dernière.

Pour rappeller sur les contraintes de développement telles que décrites dans la présentation du projet en début d'année, voici quelques points :

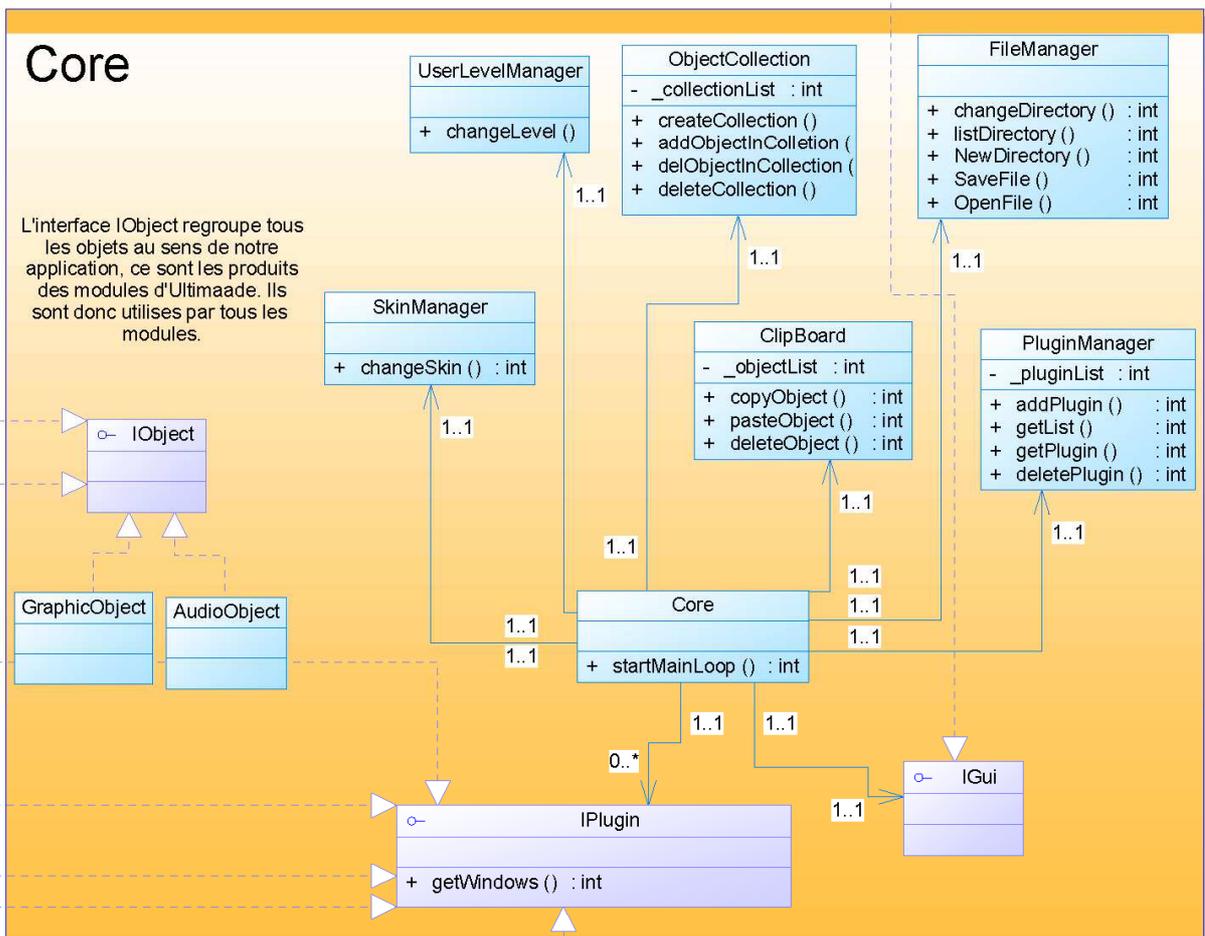
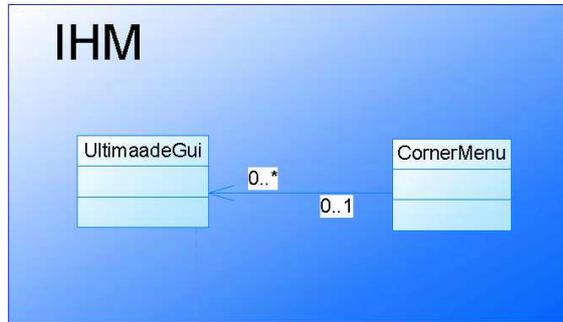
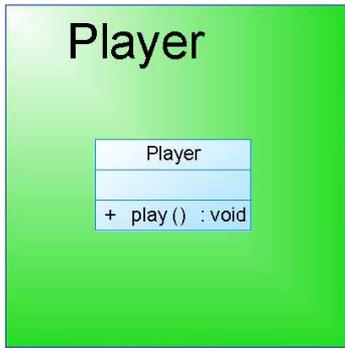
- Licence du logiciel : GNU/GPL v2 ou 3
- Principaux outils/technologies de développement : Framework QT, Langage C++
- Environnement de développement pour les premières versions : Linux, Windows
- Développement collaboratif sur <https://redmine.ryxeo.com/projets/leterrier-ultimaade>
- Gestionnaire de version : Un dépôt svn (svn) local (EPITECH), et un dépôt bazzard (bzd) distant

Petit point sur le fonctionnement de la GUI, et l'affichage pour les différents modules/plugins. Le logiciel sera muni d'une interface principale, et chaque module implémentera une mini-interface/gui représentant l'espace de travail associé. Cependant tous les affichages se feront dans l'interface principale, et la GUI aura la seule charge de faire les rendus graphiques.

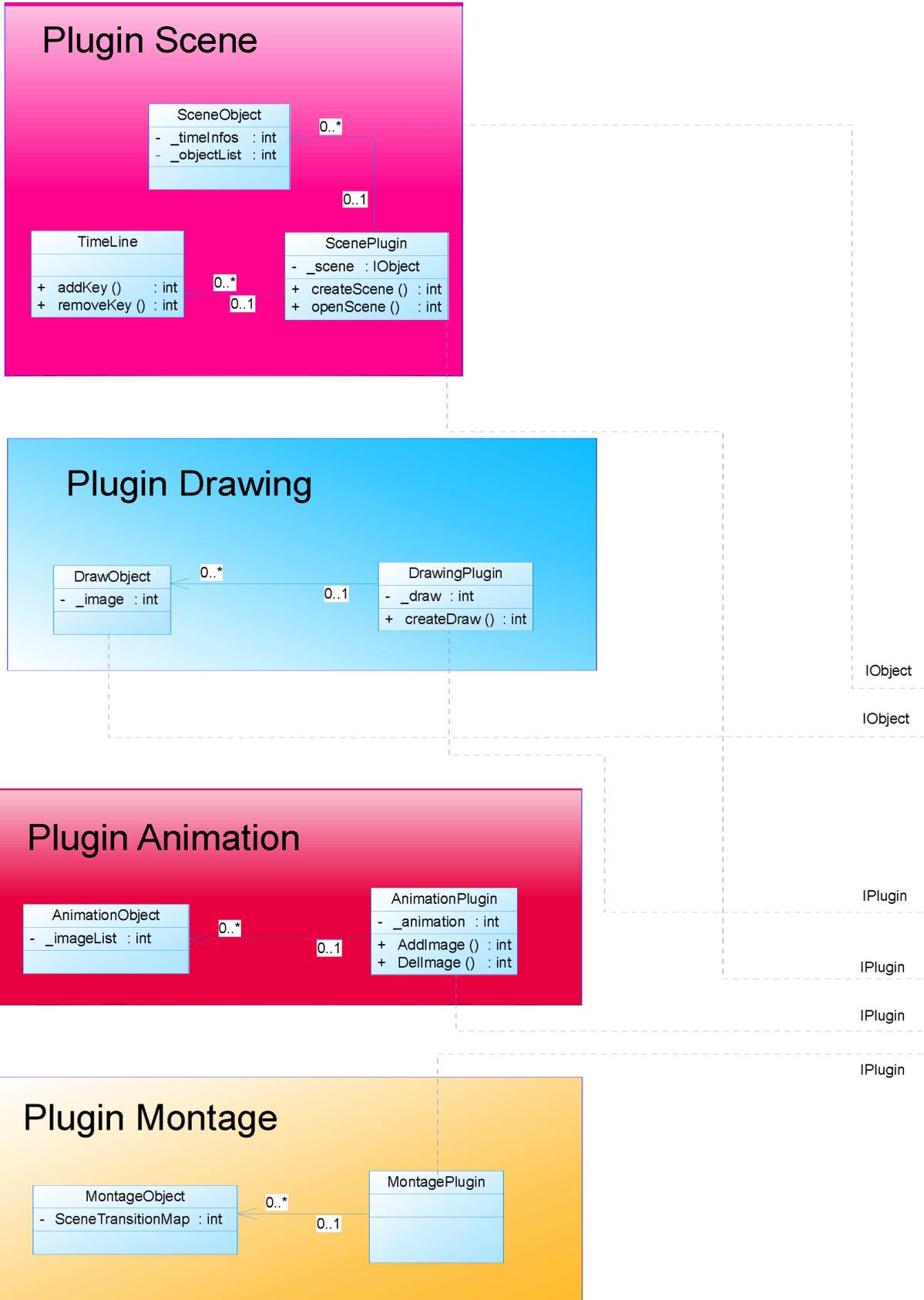
Le diagramme des classes ci-dessus illustre mieux cette dernière partie. Il est constitué des différents packages qui ont été définis jusqu'ici, et son développement entrera en première ligne des prochaines priorités.

Afin de faire rentrer le diagramme dans le document, nous l'avons divisé en deux pages. Mais ce sont deux parties du même diagramme.

**Diagramme des classes**

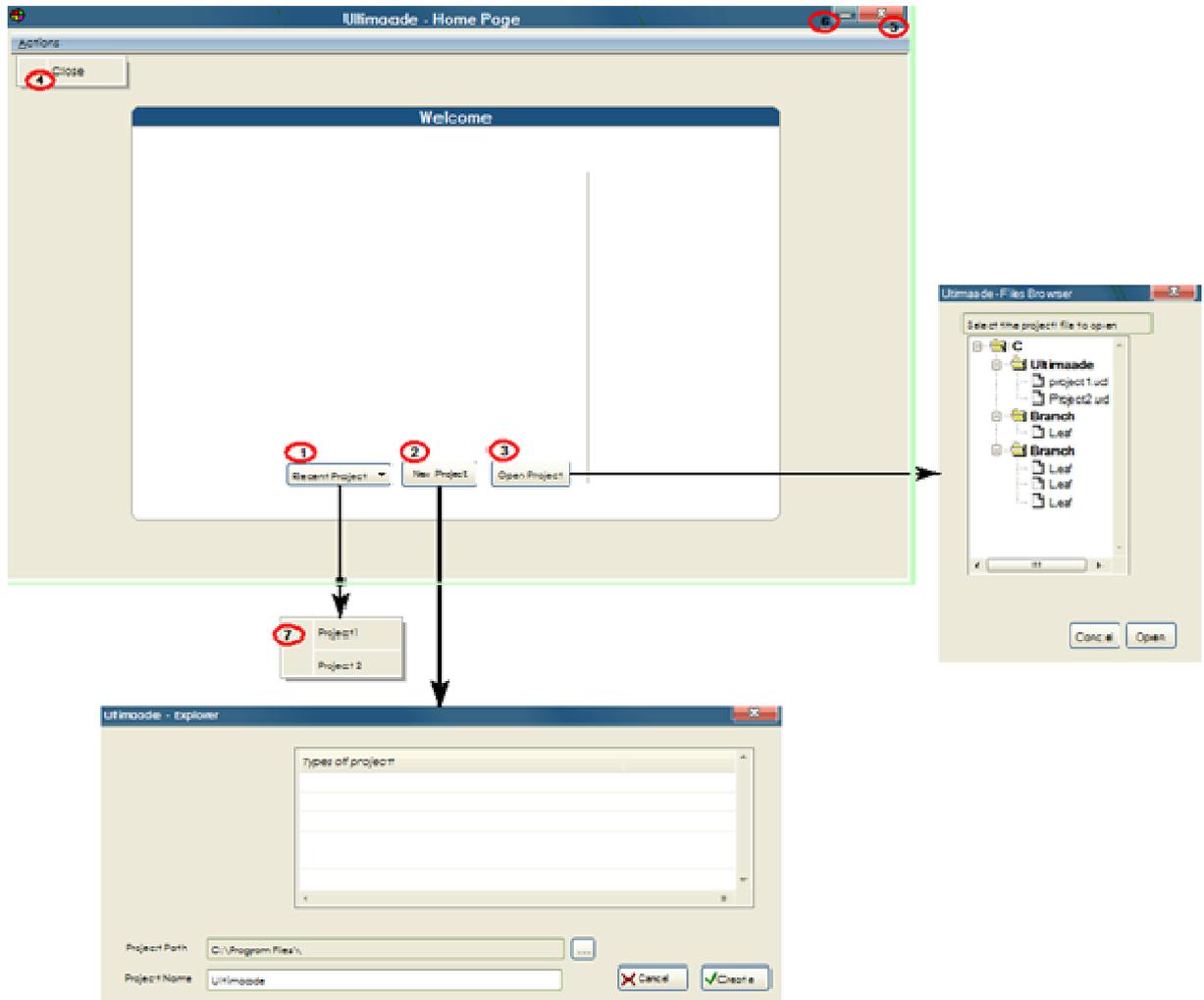


## Diagramme des classes Suite



➤ **Annexe : Document de conception IHM**

## 1. Ultimaade Page d'Accueil



Cette figure représente la page de démarrage d'ULTIMAIDE et les différentes interactions avec les boutons.

### 1.1. Intitulé des boutons

1. Recent Project
2. New Project
3. Open Project
4. Close

## 1.2. Interactions IHM

1. **Le bouton n° 1** est en relation avec le **Pop-up n° 7** qui affiche les projets récents de l'utilisateur.
2. **Le bouton n° 2** comme l'indique la flèche qui lui est associée, est en relation avec la fenêtre intitulée **Ultimaade – Explorer** et permet de créer un nouveau projet.
3. **Le bouton n° 3** comme l'indique la flèche qui lui est associée, est en relation avec la fenêtre **Ultimaade – File Browser**.
4. **Les boutons n° 4 et 5** ferment l'application.
5. **Le bouton n°6** réduit l'application.

Une fois un projet créé ou un projet récent ouvert, l'application vous mène vers l'espace de travail approprié ; en l'occurrence l'environnement de dessin dans ce cas.

## 2. Ultimaade – Environnement de dessin

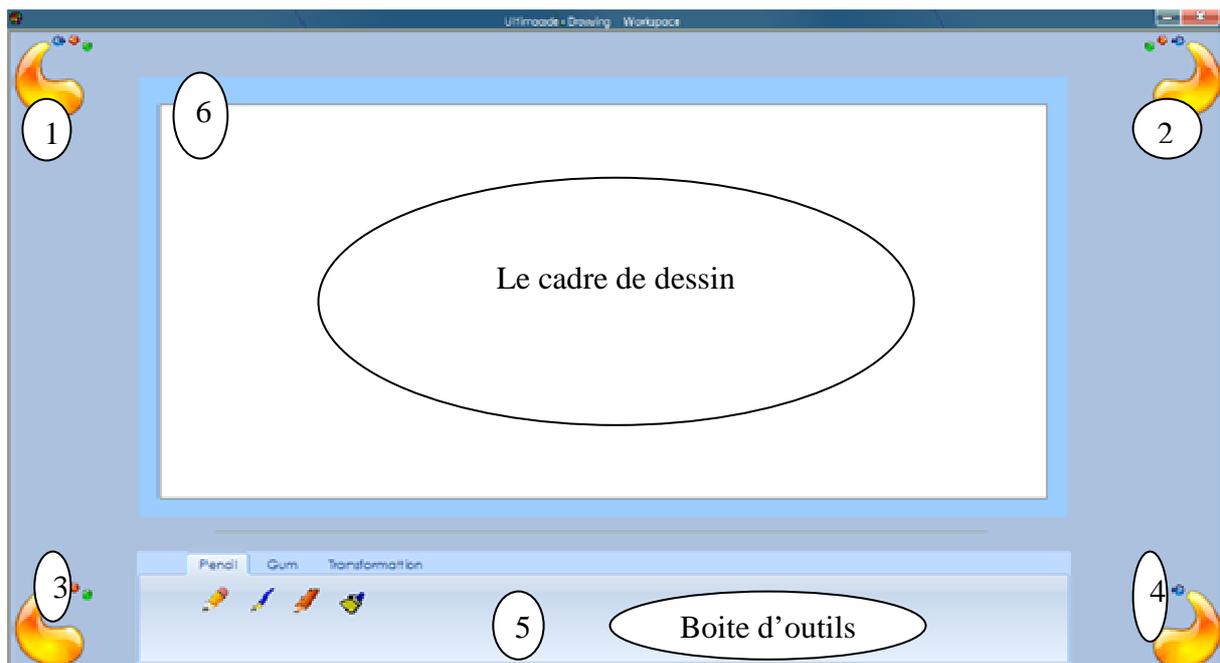
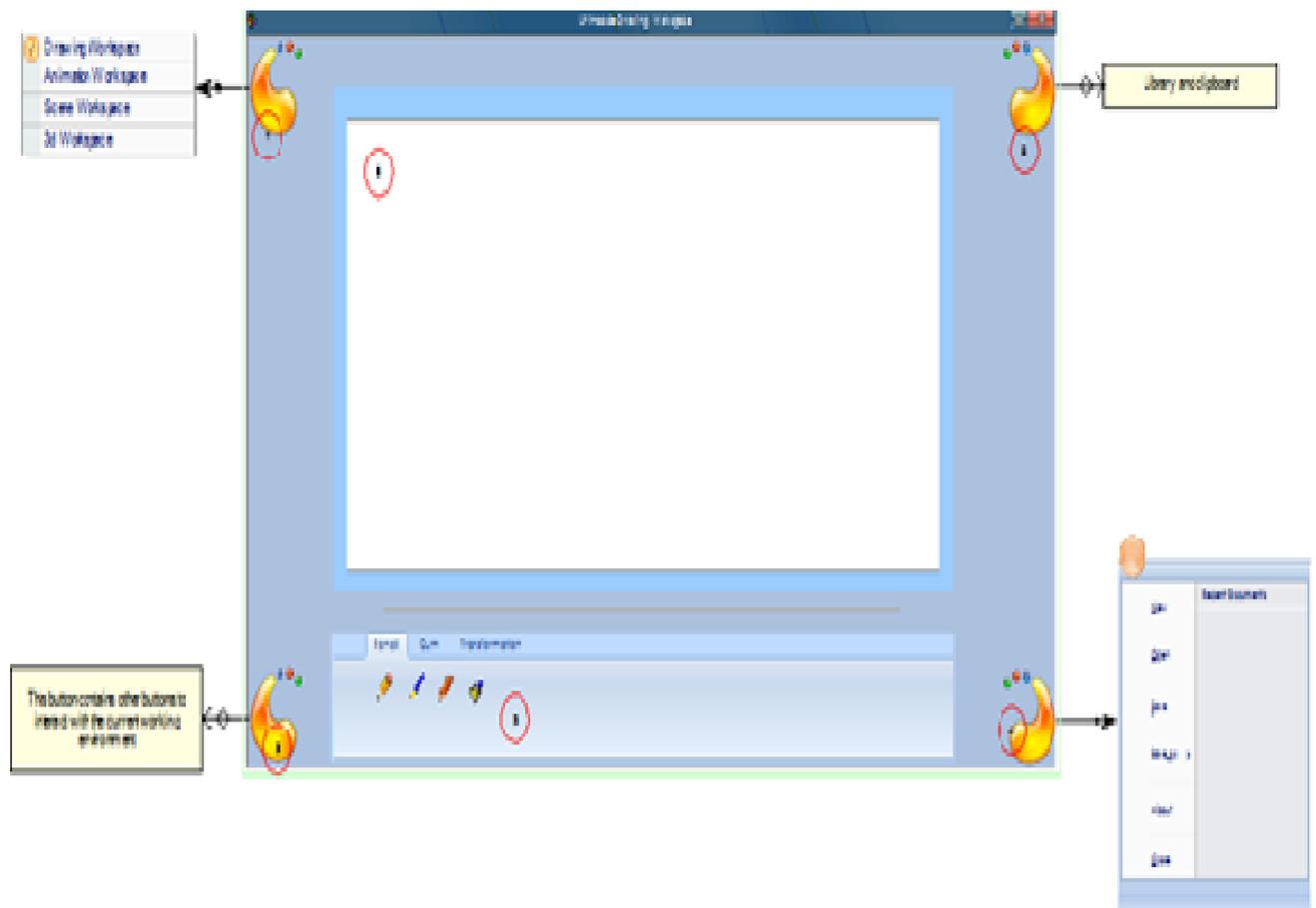


Figure 1 Environnent de travail



**Figure 2** Schéma des interactions (plus de détails ci - dessous);

Les boutons 1, 2,3 et 4 constituent les plasmoides (gadgets) de notre interface, leur présence est continue durant tout l'exécution du programme. Ils seront dans toutes les interfaces représentées par les figures des paragraphes suivant.

### 2.1 Interaction Plasmoides

1. Une interaction avec le plasmoides 1 permet de changer l'environnement de travail de l'utilisateur. Par défaut

<input checked="" type="checkbox"/>	Drawing Workspace
<input type="checkbox"/>	Animation Workspace
<input type="checkbox"/>	Scene Workspace
<input type="checkbox"/>	3d Workspace

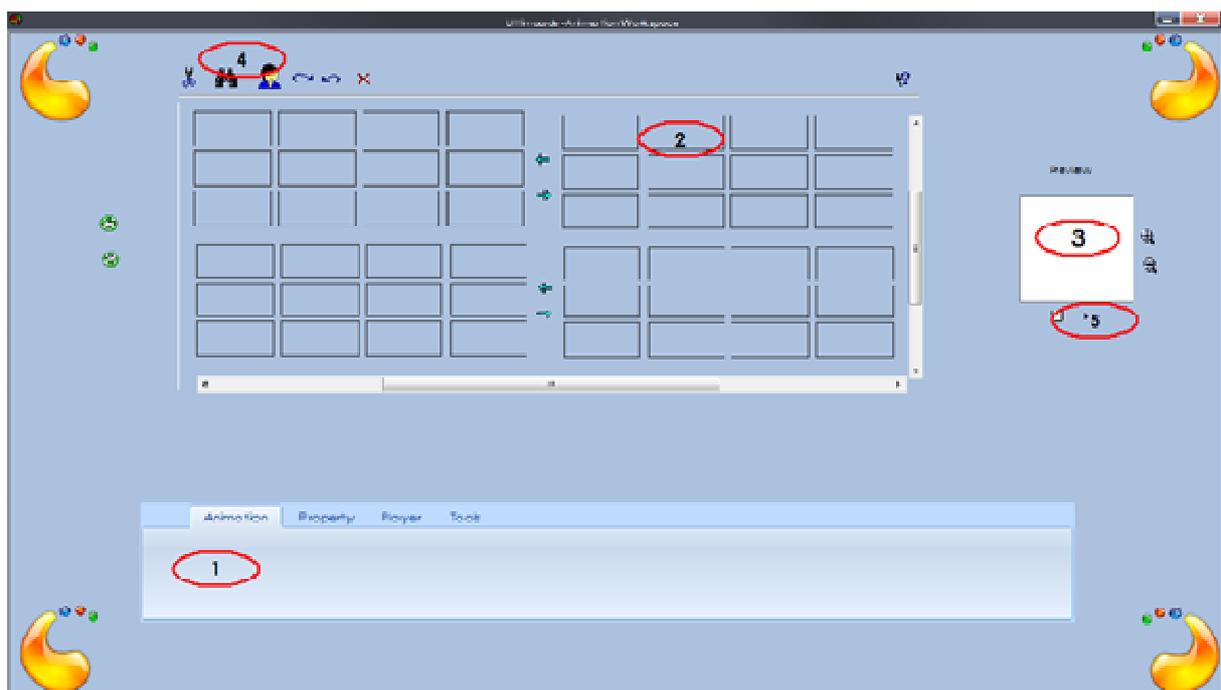
2. Une interaction avec le plasmode 2 définit la bibliothèque et le presse-papier, ce qui permettra à l'utilisateur un gain de temps dans la récupération de ses données. Il s'agit d'un système de glissement **drag and drop** intuitif.
3. Une interaction avec le plasmode 3 définit les paramètres de l'environnement de travail actuel.

Les fonctionnalités affichées/visibles dépendent de l'environnement d'édition.

4. Une interaction avec le plasmoïde 4 définit les fonctionnalités standards commune à toutes les applications, telles que sauver, ouvrir un fichier etc...



### 3. Ultimaade – Environnement d'animation



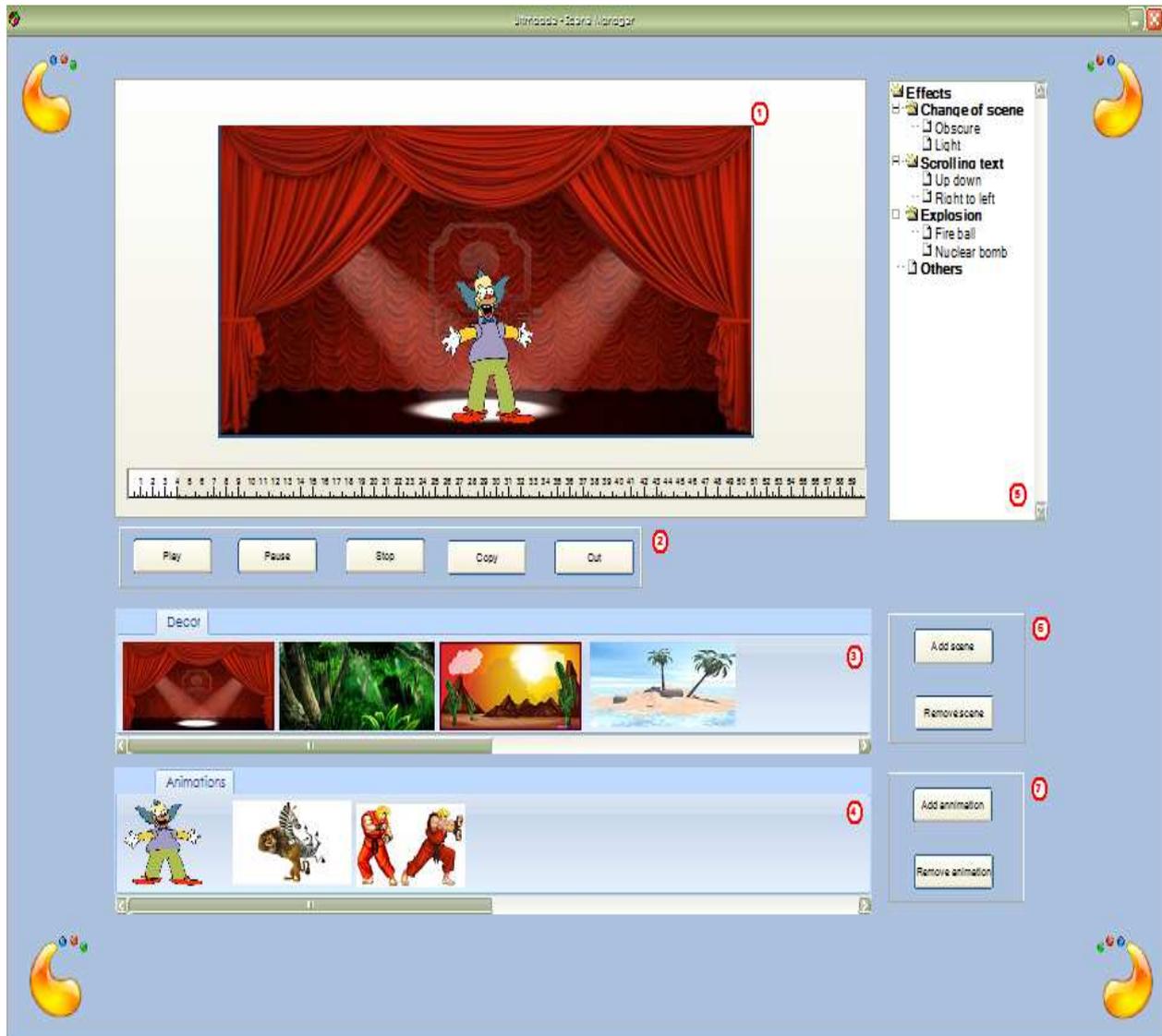
#### 3.1 Description de l'environnement

L'environnement d'animation se compose de 5 zones décrites ci-dessous respectivement.

1. Zone outils. C'est ici que l'utilisateur va interagir avec les sprites, il pourra dérouler ses animations, y ajouter quelques effets (transparences, limiter les Frame Per Second, etc..) ce n'est en aucun cas du dessin.

2. Cadre de sélection et enchainements de sprites, possibilités de switcher entre sprites, de les glisser via la bibliothèque accessible par la plasmoïde.
3. Pré visualisation de l'animation
4. Boîtes à outils limitées, disponible aussi via la zone outils.
5. Le Player, on pourra lire et arrêter les animations à travers ce menu.

#### 4. Ultimaade – Environnement de mise en Scène



**Description :**

**Élément 1:**

Il s'agit de la zone de gestion de la scène, on insère des objets (sprites, animations, etc.).

**Élément 2:**

La zone de bouton 2 permet de manipuler la visionneuse de scène ou de modifier une scène.

Ainsi, on pourra visionner une scène, la mettre sur pause ou stopper la lecture.

**Élément 3:**

Liste des décors.

**Élément 4:**

Liste des animations.

**Élément 5:**

Liste des effets qu'on peut intégrer.

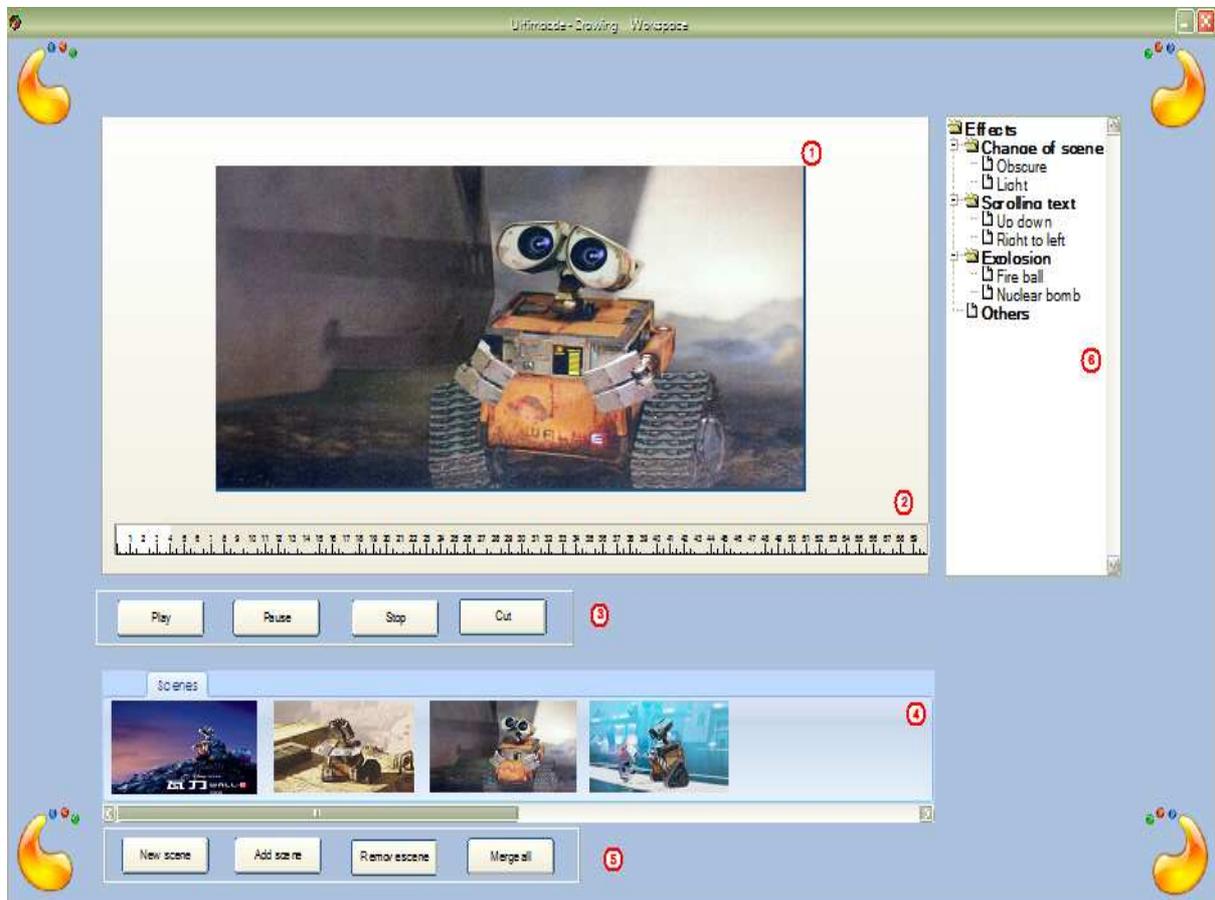
**Élément 6:**

Boutons de gestion des décors.

**Élément 7:**

Boutons de gestion de des animations.

## 5. Ultimaade – Environnement de Montage



### **Description :**

#### **Zone 1:**

Il s'agit de la zone de lecture de la scène.

#### **Zone 2:**

Chronologie des différentes séquences entre les scènes. On pourra modifier, remplacer des scènes dans le temps.

#### **Zone 3 :**

La zone de bouton 3 permet de manipuler la visionneuse de scène ou de modifier une scène.

Ainsi, on pourra visionner une scène, la mettre sur pause ou stopper la lecture.

#### **Zone 4:**

Liste des différentes scènes.

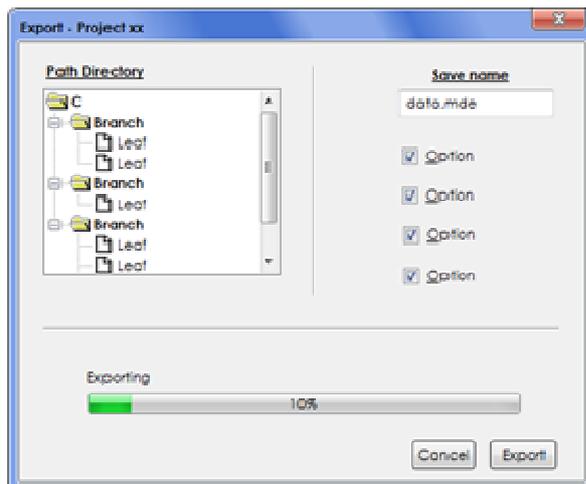
### Zone 5:

La zone de bouton 5 permet de gérer l'ajout, la suppression, la création et la fusion des différentes scènes.

### Zone 6:

L'élément 6 regroupe un panel d'effets pouvant être intégrés aux scènes.

## 6. Ultimaade – Environnement d'Export



La fonctionnalité export est disponible depuis les plasmoides comme on l'a précédemment dit.

Voici un aperçu de la fenêtre d'exportation simple de prise en main.