Projet P2

*Réaliser une animation permettant la résolution d’un système de m équations à n inconnues*

Bastien Burri

Bandelier Matthieu

Gonin Nicolas

Vulliemin Kevin

Équipe 7

2 mars 2015

Table des matières

[1 Introduction 2](#_Toc415441301)

[1.1 Planning 3](#_Toc415441302)

[2 Spécifications des besoins 4](#_Toc415441303)

[2.1 Besoins utilisateurs 4](#_Toc415441304)

[2.1.1 Acteurs 4](#_Toc415441305)

[2.1.2 Cas d’utilisation 4](#_Toc415441306)

[2.2 Scénario 5](#_Toc415441307)

[2.2.1 Séparation 5](#_Toc415441308)

[2.2.2 UC 1.1 Créer un problème 5](#_Toc415441309)

[2.2.3 UC 1.2 Charger un problème 6](#_Toc415441310)

[2.2.4 UC 1.3 Sauvegarder un problème 6](#_Toc415441311)

[2.2.5 UC 1.4 Remplir / Modifier la matrice 7](#_Toc415441312)

[2.2.6 UC 2.1 Résoudre directement 8](#_Toc415441313)

[2.2.7 UC 2.2 Résoudre par étape 8](#_Toc415441314)

[2.3 Exigences fonctionnels et non fonctionnels 9](#_Toc415441315)

[2.4 Maquette 10](#_Toc415441316)

[3 Spécifications techniques 10](#_Toc415441317)

[3.1 Analyse du domaine métier 10](#_Toc415441318)

[3.2 Diagramme de classe complet 10](#_Toc415441319)

[3.3 Dynamique de l’application 10](#_Toc415441320)

[3.3.1 UC X.X 10](#_Toc415441321)

[3.4 Architecture de déploiement 10](#_Toc415441322)

[3.5 Choix des librairies externes 11](#_Toc415441323)

[3.6 Tests fonctionnels 11](#_Toc415441324)

[4 Conclusion 11](#_Toc415441325)

# Introduction

Ce document présente les spécifications des besoins et spécifications techniques du projet « *Réaliser une animation d’un système de m équations à n inconnues*».

L’objectif du projet est de résoudre un système de m équations à n inconnues et de l’illustré à l’aide d’un exemple dans un but pédagogique. L’application sera utilisée pour expliquer la résolution d’un système de m équations à n inconnues à des étudiants de première année ayant des difficultés avec les maths. L’application aura une approche mathématique différente de celles enseignées habituellement en cours, elle contiendra une explication basée sur un problème naturel.

L’utilisateur spécifie à l’application le nombre d’équations ainsi que le nombre d’inconnues. Il devra également spécifier la manière dont il désire que ses équations soient résolues. En effet, ces équations pourront être résolues de deux manières différentes. La 1ère sera une solution ou toutes les étapes de la résolution seront affichées et la 2ème affichera uniquement la solution.

Une fois lancé, le programme résout les équations en fonction des paramètres précédemment saisis. Si les étapes de la résolution sont affichées, alors l’utilisateur aura la possibilité de naviguer entre ces étapes et il y aura une animation sur les lignes.

Les objectifs suivants seront réalisés dans le cadre du projet P2 Java :

* Illustration de la résolution d’un problème naturel à l’aide d’un système d’équations dans un but pédagogique.
* Résolution efficiente avec un algorithme performant sans animation d’un système de *m* équations à *n* inconnus.
* Résolution avec animation d'un système de maximum 5 équations à *n* inconnues.
* Configurations :
  + Entrer le nombre d'équation(s) et le nombre d'inconnue(s) du système d'équations;
  + Résolution étape par étape, avec la possibilité de naviguer entre les étapes, et animations sur les lignes.

## Planning

(screen du planning et mettre la page en paysage si besoin)

# Spécifications des besoins

Ce chapitre est dédié aux besoins du client par rapport au programme voulu. Il doit énumérer les fonctionnalités et service attendu par le programme. Pour ce faire, on commencera par identifier les acteurs sur chaque action, puis on définira ce qu’il peut faire dans notre programme. On pourra alors définir les possibilités des actions en effectuant des scénarios.

## Besoins utilisateurs

La demande du projet était telle que l’utilisateur voulait pouvoir créer des matrices puis résoudre celles-ci dans un cadre pédagogique. L’utilisateur a donc besoin d’avoir une résolution assisté mais afin de rendre ce programme à la fois pédagogique et utile, il a été décidé de créer 2 modes de résolution :

1. Résolution par étape : à des fins pédagogiques, avec affichage de l’historique des opérations ;
2. Résolution directe : à des fins utiles, pour résoudre des équations ou des matrices.

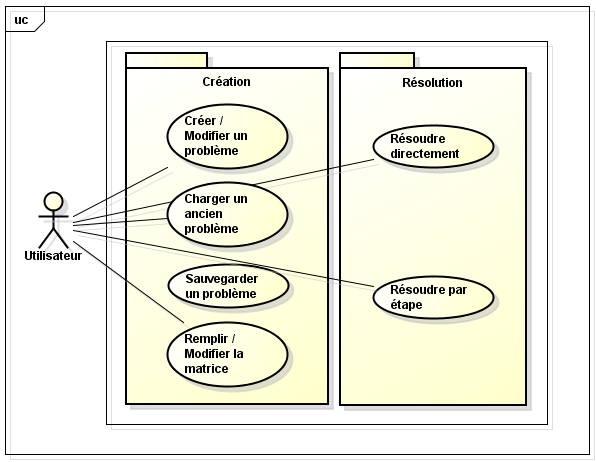
Afin que le programme soit utilisable de manière optimale par un professeur, il est préférable que celui-ci puisse sauvegarder un problème chez lui. Ainsi, celui-ci pourrait préparer son travail à la maison en configurant ses problèmes à sa guise et le présenter le lendemain sans erreur et sans perte de temps.

Pour terminer, la configuration du problème et le remplissage de la matrice ont été placée dans des fenêtres annexes qui peuvent être affichées pour modifier un problème, ou modifier les chiffres de la matrice, à tout instant. L’utilisateur peut donc changer entre le mode directe et par étape sans toucher.

### Acteurs

Le seul acteur dans le programme est l’utilisateur courant. Le programme peut être utilisé par un professeur lors d’une présentation mais également par un élève qui veut essayer de comprendre par lui-même les étapes de résolution d’une équation avec le rendu graphique. Néanmoins, même si les utilisateurs sont différents, ils disposent des mêmes fonctions et des mêmes droits car le programme est purement local.

### Cas d’utilisation



## Scénario

### Séparation

|  |  |
| --- | --- |
| **ID** | Package 1 |
| **Nom** | Création |
| **Description** | * L’utilisateur à la possibilité de créer un problème de n équations à m inconnues à résoudre et remplir les valeurs au moyen de champs générés par les paramètres que l’utilisateur peut spécifier, comme le choix du nombre d’inconnues ou d’équation. * L’utilisateur peut également charger un ancien problème du moment qu’il l’a précédemment sauvegardé. * Le client peut également modifier les valeurs de la matrice s’il a commis une erreur lors de la génération ou que le résultat n’est pas celui qu’il attendait. |

|  |  |
| --- | --- |
| **ID** | Package 2 |
| **Nom** | Résolution |
| **Description** | * L’utilisateur peut résoudre la matrice généré dans le package « Création » selon la méthode spécifié par l’utilisateur. * L’utilisateur peut visualiser les étapes de résolution afin de comprendre comment résoudre à la main cette dernière équation mais il peut également contrôler rapidement un travail effectué à la main avec une résolution directe. |

### UC 1.1 Créer un problème

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | UC 1.1 | **Nom** | Créer un problème / Modifier un problème | | |
| **Date** | 16.03.2015 | **Auteur** | Équipe 7 | **Priorité** | Haut |
| **Description et objectifs** | * Le but de ce cas d’utilisation est de créer un problème à résoudre dans la suite de l’application. On sépare ici clairement la partie de paramétrage du problème, du remplissage de la matrice généré (UC 1.4) ; * L’utilisateur doit choisir les paramètres d’un problème qui se résout sous forme d’équations. Ces détails servent ensuite à générer le canevas de la matrice afin de laisser l’utilisateur la remplir (UC 1.4). | | | | |
| **Acteurs** | Utilisateur | | | | |
| **Déclencheur** | * Clic sur le bouton « Nouveau » sur la page d’accueil ; * Clic sur le menu « Nouveau problème » dans la barre de menu. | | | | |
| **Scénario nominal** | 1 : Affichage de la fenêtre de paramètres ;  2 : Paramétrage et génération de la matrice ;  3 : Confirmation ;  4 : Lancement de l’UC 1.4 : Remplir la matrice | | | | |
| **Scénarios alternatifs** | 3.1 : Détection d’un problème de titre ou de valeurs ;  3.2 : Réaffichage de la fenêtre de paramètres avec champs en rouge (étape 1) | | | | |
| **Préconditions** | Aucune | | | | |
| **Post-Conditions** | Remplissage de la matrice d’équation (UC 1.4) | | | | |
| **Exigences** |  | | | | |

### UC 1.2 Charger un problème

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | UC 1.2 | **Nom** | Charger un ancien problème | | |
| **Date** | 16.03.2015 | **Auteur** | Équipe 7 | **Priorité** | Basse |
| **Description et objectifs** | * Le but de ce cas d’utilisation est de permettre à l’utilisateur de charger un problème qu’il aurait déjà implémenté afin d’éviter l’étape de création si on l’a déjà effectué (UC 1.1). * L’utilisateur doit avoir préalablement sauvegardé son problème afin de pouvoir le charger. | | | | |
| **Acteurs** | Utilisateur | | | | |
| **Déclencheur** | * Clic sur le bouton « Charger » sur la page d’accueil ; * Clic sur le menu « Charger problème » dans la barre de menu. | | | | |
| **Scénario nominal** | 1 : Affichage d’une fenêtre contenant la liste des problèmes sauvegardés ;  2 : Sélection d’un problème ;  3 : Confirmation | | | | |
| **Scénarios alternatifs** | 1.1 : Affichage des problèmes sauvegardés directement dans la fenêtre d’accueil. | | | | |
| 3.1 : Chemin ou projet inexistant ;  3.2 : Réaffichage des projets excluant l’erreur. | | | | |
| **Préconditions** | Fichier disponible / Sauvegarde d’au moins un problème : UC 1.3 | | | | |
| **Post-Conditions** | Aucune | | | | |
| **Exigences** |  | | | | |

### UC 1.3 Sauvegarder un problème

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | UC 1.3 | **Nom** | Sauvegarder un problème | | |
| **Date** | 16.03.2015 | **Auteur** | Équipe 7 | **Priorité** | Basse |
| **Description et objectifs** | Ce cas d’utilisation permet à l’utilisateur de sauvegarder un problème afin de pouvoir le reprendre lors d’une utilisation ultérieur du programme. | | | | |
| **Acteurs** | Utilisateur | | | | |
| **Déclencheur** | * Clic sur le menu « Sauvegarder » dans la barre de menu ; * Clic sur le menu « Sauvegarder sous… » dans la barre de menu ; * Quitter le programme. | | | | |
| **Scénario nominal** | 1 : Confirmation | | | | |
| **Scénarios alternatifs** | 0.1 : Affichage d’une fenêtre de navigation ;  0.2 : Sélection de l’emplacement. | | | | |
| 2.1 : Erreur d’accès ou d’écriture  2.2 : Appel du scénario alternatif 0. | | | | |
| **Préconditions** | Aucune | | | | |
| **Post-Conditions** | Aucune | | | | |
| **Exigences** |  | | | | |

### UC 1.4 Remplir / Modifier la matrice

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | UC 1.4 | **Nom** | Remplir / Modifier la matrice | | |
| **Date** | 16.03.2015 | **Auteur** | Équipe 7 | **Priorité** | Haut |
| **Description et objectifs** | * Ce cas a pour but de laisser à l’utilisateur le choix des valeurs composants la matrice du problème ; * Le client reçoit une fenêtre à trou, basé sur les paramètres saisi à l’UC 1.1, et rempli la matrice ainsi généré. | | | | |
| **Acteurs** | Utilisateur | | | | |
| **Déclencheur** | * UC 1.1 ; * Clic sur le menu « Modifier matrice » de la barre de menu. | | | | |
| **Scénario nominal** | 1 : Affichage de la fenêtre de remplissage ;  2 : Saisi des valeurs ;  3 : Confirmation ;  4 : Lancement de l’UC 2.2. | | | | |
| **Scénarios alternatifs** | 3.1 : Erreur de saisi (autres que nombres) ;  3.2 : Réaffichage de la fenêtre de résolution avec champs en rouge. | | | | |
| 4.1 : Lancement de l’UC 2.1. | | | | |
| **Préconditions** | Problème déjà paramétré : UC 1.1 | | | | |
| **Post-Conditions** | Aucune | | | | |
| **Exigences** |  | | | | |

### UC 2.1 Résoudre directement

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | UC 2.1 | **Nom** | Résoudre directement | | |
| **Date** | 16.03.2015 | **Auteur** | Équipe 7 | **Priorité** | Haut |
| **Description et objectifs** | * Ce cas d’utilisation permet à l’utilisateur de résoudre le problème crée dans le package 1 (Création) de manière directe et optimisé ; * L’utilisateur aperçoit directement la solution du problème et n’a pas accès aux étapes de résolution. | | | | |
| **Acteurs** | Utilisateur | | | | |
| **Déclencheur** | * UC 1.2 ; * UC 1.4. | | | | |
| **Scénario nominal** | 1 : Affichage de la fenêtre de résolution ;  2 : Affichage des résultats. | | | | |
| **Scénarios alternatifs** | Aucun | | | | |
| **Préconditions** | Avoir une matrice remplie : UC 1.1 et UC 1.4 ou UC 1.2 | | | | |
| **Post-Conditions** | Aucune | | | | |
| **Exigences** |  | | | | |

### UC 2.2 Résoudre par étape

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | UC 2.2 | **Nom** | Résoudre par étape | | |
| **Date** | 16.03.2015 | **Auteur** | Équipe 7 | **Priorité** | Haut |
| **Description et objectifs** | * Ce cas d’utilisation permet à l’utilisateur de résoudre le problème crée dans le package 1 (Création) de manière compréhensible et simple ; * L’utilisateur peut avancer dans les étapes en cliquant sur un bouton, idem pour revenir en arrière ; * L’utilisateur à la vue sur l’ensemble des étapes effectuées (logs) pour résoudre le problème et peut sélectionner une étape dans la liste afin de voir ce qui s’y est passé. | | | | |
| **Acteurs** | Utilisateur | | | | |
| **Déclencheur** | * UC 1.2 ; * UC 1.4. | | | | |
| **Scénario nominal** | 1 : Affichage de la fenêtre de résolution ;  2 : Affichage d’une étape de résolution ;  3 : Rappel de l’étape de scénario 2 jusqu’à résolution du problème. | | | | |
| **Scénarios alternatifs** | 2.1 : Clic sur le bouton pour avancer dans les étapes ;  2.2 : Affichage de l’étape suivante de résolution. | | | | |
| 2.1 : Clic sur le bouton pour reculer dans les étapes ;  2.2 : Affichage de l’étape précédente de résolution. | | | | |
| 2.1 : Clic sur un log ;  2.2 : Affichage de l’étape de résolution correspondante. | | | | |
| **Préconditions** | Avoir une matrice remplie : UC 1.1 et UC 1.4 ou UC 1.2 | | | | |
| **Post-Conditions** | Aucune | | | | |
| **Exigences** |  | | | | |

## Exigences fonctionnels et non fonctionnels

Les exigences fonctionnelles fournissent des détails d’implémentation sur les fonctionnalités de l’application. Les exigences non fonctionnelles représentent les facteurs extérieurs imposés à l’application devant être pris en compte lors du développement.

|  |  |  |
| --- | --- | --- |
| ID Exigence | Type | Description |
| EF1 | Fonctionnelle | Pouvoir utiliser très facilement l’application. Elle doit être intuitive. |
| EF2 | Fonctionnelle | Afficher automatiquement les résultats à la fin de la résolution du mode sans animation. |
| EF3 | Fonctionnelle | Afficher automatiquement les résultats à la fin de la résolution du mode avec animation avec la possibilité de naviguer entre les étapes. |
| EF4 | Fonctionnelle | Résolution avec animation d’un système de maximum 5 équations à *n* inconnues. |
| EF5 | Fonctionnelle | Illustration de la résolution d’un problème naturel à l’aide d’un système d’équations dans un but pédagogique. |
| ENF1 | Performance | Résolution efficiente avec un algorithme performant, sans animation, d’un système de *m* équations à *n* inconnus. |
| ENF3 | Ergonomie | Ne pas paralyser l’application lors de la résolution et garantir la réactivité de l’interface en général. |

## Maquette

# Spécifications techniques

## Analyse du domaine métier

## Diagramme de classe complet

## Dynamique de l’application

### UC X.X

## Architecture de déploiement

Le déploiement de l’application se fera sur des PC de l’école ou des PC personnels. Comme l’application est codée en Java, le terminal sur lequel l’application sera exécutée requiert l’installation d’une machine virtuelle Java si elle n’est pas déjà installée.

L’application sera distribuée sous forme de classes Java compilées et zippées dans des librairies JAR (Java Archive).

Deux packages de déploiement seront distribués :

* Package contenant uniquement les librairies JAR du projet, pouvant être exécutées avec une JVM (Java Virtual Machine) déjà installée sur un PC ;
* Package contenant les librairies JAR du projet et une installation du JRE (Java Runtime Environment) contenant une machine virtuelle Java, pour un PC qui ne serait pas encore équipé de Java.



## Choix des librairies externes

## Tests fonctionnels

# Conclusion

Comme il a été expliqué précédemment, le but du projet est de créer un programme de support pour étudier la résolution d’un problème contenant *m* équations à *n* inconnues.

L’utilisateur disposera de deux possibilités pour résoudre ces équations:

* La 1ère  sera une solution ou toutes les étapes de la résolution seront affichées, avec une possibilité de naviguer entre les étapes;
* La 2ème affichera uniquement les solutions.

Comme le programme a premièrement un but pédagogique, les premières itérations de développement porteront essentiellement sur la résolution par étape et la création d’une matrice. Seront ensuite implémentés les fonctions de sauvegardes et de résolution directe.