
Cachier des charges : PicExplorer

LANVIN Elyan - MARCAIS Thomas

RAMOLET Arthur - GUENVER Loïc

AYDIN Emre - FOUCAULT Antoine

24 février 2014

©2014 Groupe C - Tous droits réservés

Sommaire

1	Présentation du projet	3
1.1	Contexte	3
1.2	Étude de l'existant	3
1.3	Objectifs	4
1.4	Calendrier	4
1.5	Critères d'acceptabilité du produit	5
2	Analyse des besoins	5
2.1	Liste des acteurs	5
2.2	Expression des besoins	5
2.3	Besoins fonctionnels	6
2.4	Besoins non-fonctionnels	7
2.5	Fonctionnalités	7
2.6	Cas d'utilisation	10
3	Livrables	12
4	Contraintes	13
4.1	Documentation	13
4.2	Délais	13
4.3	Contraintes techniques	13
5	Organisation du projet	13
5.1	Diagramme WBS	14
5.2	Diagramme de Gantt	15

1 Présentation du projet

1.1 Contexte

Ce projet se déroulera dans le cadre d'un projet tutoré lors du semestre 6 de L3 SPI Informatique. Ce document définit les exigences et les besoins du client, ainsi que les fonctionnalités qui seront présentes dans la future application.

1.2 Étude de l'existant

Hanjie-star : Jeu de picross en ligne gratuit ¹

Picross Nintendo© : Jeu de picross sur Nintendo© DS

Zelda Picross : Jeu de picross amateur open source ²

La partie ci-dessous décrit les différentes fonctions communes et spécifiques des trois applications :

- **Fonctions communes**

- Cocher case valide
- Cocher case non valide
- Cocher en continu (en faisant glisser la souris)
- Timer
- Vérifier la grille

- **Hanjie-star**

- Éditeur
- Recherche de grille dans la base (sur différents critères de description de grilles)

- **Picross Nintendo©**

1. Disponible sur la page suivante : www.hanjie-star.fr

2. Téléchargeable sur la page suivante : <http://www.zeldaroth.fr/dlpicross.php>

- Correction temps réel (pénalité de temps et signalement en cas d'erreur)
- Pas de remplacement
- Tactile

- **Zelda Picross**

- Mode aventure

1.3 Objectifs

Ce projet doit aboutir à la création d'une application de type jeu de picross (ou henjie), destinée à tout public. Un système d'aide permettra de terminer une grille, peu importe son expérience et la difficulté de la grille.

L'application comprendra :

- La résolution d'une grille de picross
- Un éditeur pour créer ses propres grilles
- Un système d'aide à la résolution
- Un système de sauvegarde/chargement (persistance des données)
- Un tutoriel (sous forme de pop-up/fenêtre d'aide)

1.4 Calendrier

La soutenance est prévue pour le vendredi 16 mai à 13h30, le projet est à rendre le même jour au plus tard. Notre projet étant organisé selon le modèle de cycle de développement en spirale, voici les dates de fin des différentes spirales (date de présentation au client) :

- 14/03/2014
- 11/04/2014
- 25/04/2014
- 05/05/2014

— 12/05/2014

1.5 Critères d'acceptabilité du produit

L'application doit répondre aux critères suivants :

- Validation du produit via un dossier de tests réalisé par notre groupe
- Respect des contraintes client (choix technologique, fonctionnalités de l'application)

2 Analyse des besoins

2.1 Liste des acteurs

L'utilisateur :

À l'aide de l'application, l'utilisateur pourra éditer ses propres grilles de Picross en noircissant les cases de la grille comme voulu.

Il peut également générer un fichier de sauvegarde de sa grille qui sera exportable avec d'autres instances de l'application. L'utilisateur pourra également se servir de l'application pour jouer au Picross. Il pourra sélectionner une des grilles à résoudre par défaut ou il pourra importer des grilles éditées au préalable.

Lors de la résolution des grilles, il pourra faire appel à une aide de la machine. Il peut également consulter les statistiques et classements du jeu.

2.2 Expression des besoins

Le client souhaite disposer d'une application réalisée sous Ruby. Celle-ci sera décomposée en deux grandes parties :

- L'édition de grilles de Picross via une interface simple permettant de noircir la grille voulue, générer une grille aléatoirement ou à partir d'une image (option)
- La résolution des grilles de Picross, selon les règles traditionnelles du jeu, avec possibilité de demander de l'aide au jeu si le joueur est bloqué. L'application doit également être capable de charger, au démarrage, la dernière grille non résolue si le jeu a été quitté précédemment lors de la résolution de celle-ci.

2.3 Besoins fonctionnels

Édition des grilles de Picross :

- Création d'une nouvelle grille de Picross à partir d'une grille vierge
 - L'utilisateur devra choisir la taille de grille puis noircir les cases tel qu'il le souhaite
- Création d'une grille de Picross aléatoirement
 - L'utilisateur doit avoir la possibilité, via un bouton par exemple, de noircir aléatoirement une grille dans l'éditeur. Il lui est également possible de la modifier manuellement par la suite
- Exportation d'une grille
 - Après avoir édité sa grille, l'utilisateur peut générer un fichier de sauvegarde de sa grille qu'il pourra partager avec d'autres utilisateurs
- Modification d'une grille existante
 - L'utilisateur peut modifier une grille qu'il a créé préalablement
- Importation d'une grille à partir d'une image (OPTION)
 - L'utilisateur peut générer une grille à partir d'une image qu'il spécifiera

Résolution des grilles de Picross :

- Démarrer une partie
 - L'utilisateur doit choisir entre deux modes : « Grilles par défaut » ou « Grilles importées ». Ensuite il sélectionne la grille à laquelle il souhaite jouer
- Aide à la résolution automatique
 - Lors d'une partie, le joueur peut demander l'aide du jeu s'il se voit bloqué. Différents niveaux d'aide sont disponibles : résolution (d'une/d'un ensemble) de case(s) (joker) ou conseils

- Sauvegarder la grille en cours
 - L'application doit pouvoir gérer différentes manières de sauvegarder ses parties en cours. Soit l'application garde la dernière partie en cours en mémoire et propose de la reprendre au démarrage, soit on peut sauvegarder la partie en cours via une option dans la barre de menu
- Importation d'une grille
 - L'utilisateur doit pouvoir importer une grille sous forme de fichier de sauvegarde pour pouvoir y jouer

2.4 Besoins non-fonctionnels

- Sérialisation des objets pour persistance
 - L'application a la possibilité, à son extinction, de sérialiser les grilles, les statistiques et les options afin de pouvoir les récupérer au redémarrage de l'application

2.5 Fonctionnalités

Le jeu

Référence	Fonctionnalité
	Traitement/Persistance des données
PE100	Sauvegarder/Charger les options
PE101	Sauvegarder/Charger/Supprimer une partie
PE102	Sauvegarder/Charger les scores et les statistiques
PE140	Crypter les données (OPTION)
	Jeu
PE110	Jouer une partie
PE101	Système d'aide

- **Jeu :**

Aider le joueur : L'utilisateur doit pouvoir recevoir une aide du logiciel dans le cas où il est bloqué. Il pourra interagir sur le plateau de deux

différentes manières. Aider l'utilisateur en noircissant une case correcte (joker) ou simplement lui donner un indice sur une ligne/colonne où jouer

Jouer une partie : L'utilisateur doit pouvoir jouer une partie de Picross à l'aide de la souris. Les différentes interactions disponibles seront définies lors de la phase de conception

- **Traitement et persistance des données :**

Sauvegarder/Charger les options : Les choix de l'utilisateur concernant les options du jeu sont sauvegardées entre les différents lancements du jeu

Sauvegarder/Charger/Supprimer une partie : L'utilisateur peut sauvegarder sa partie en cours afin de la reprendre ultérieurement. Lorsqu'il quitte le jeu en pleine partie et qu'il revient, le jeu lui propose de reprendre la partie précédente ou de retomber sur les menus. L'utilisateur peut également supprimer ses sauvegardes

Sauvegarder/Charger les scores et les statistiques : Le jeu charge automatiquement les scores et statistiques des différents joueurs au démarrage

Crypter les données (OPTION) : Toute donnée sauvegardée sur le système est cryptée de façon à assurer l'intégrité des sauvegardes

L'éditeur

Référence	Fonctionnalité
	Traitement/Persistance des données
PE103	Sauvegarder/Charger un plateau
	Éditeur
PE120	Créer un plateau aléatoirement
PE121	Créer un plateau
PE122	Modifier un plateau
PE141	Importer à partir d'une image (OPTION)

- **Éditeur :**

Créer un plateau : L'utilisateur doit pouvoir créer un plateau personnalisé en remplissant un plateau vide

Créer un grille aléatoirement : L'application doit pouvoir créer un plateau aléatoire pour l'utilisateur

Modifier un plateau : L'utilisateur doit pouvoir modifier un plateau déjà existant

Importer à partir d'une image (OPTION) : L'utilisateur doit pouvoir créer une grille à partir d'une image locale ou distante (web) qu'il spécifiera

- **Traitement et persistance des données :**

Sauvegarder/Charger un plateau : L'utilisateur doit pouvoir sauvegarder ses plateaux en cours et si l'application quitte sans avoir sauvegarder, le dernier plateau est sauvegarder automatiquement

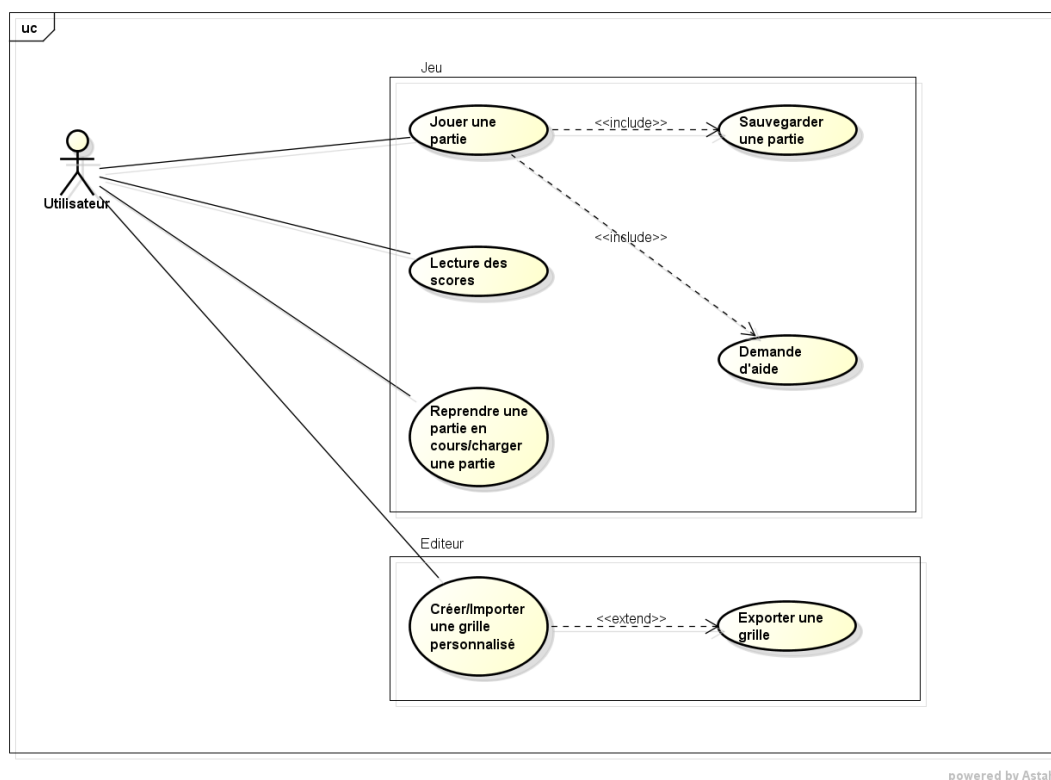


FIGURE 1 – Diagramme de cas d'utilisation

2.6 Cas d'utilisation

Cas d'utilisation n° 1

Nom : Résolution picross

Description : L'utilisateur tente de résoudre une grille de picross

Acteur : L'utilisateur de l'application

Préalables : L'application doit être téléchargée

Conséquents : L'utilisateur peut exécuter l'application

1. Choix des options :

- L'utilisateur clique sur l'option "nouvelle partie"
- Le système demande si l'on souhaite une grille choisie ou aléatoire, dans le premier cas on préselectionne la première grille non résolue
- Si l'utilisateur demande une grille aléatoire :
 - Le système propose différentes tailles de grille (Peut être encombrant s'il y en a beaucoup, mais évite une entrée manuelle qui pourrait être incorrecte)
- L'utilisateur choisit une taille
- Le système sélectionne une grille
- Ajout d'un malus par joker utilisé
- Dans le cas où le nombre de joker est limité :
 - Ajout d'un bonus en fonction du nombre de jokers non utilisés (optionnel)
- On considère que le timer est toujours mis en place

2. Résolution du picross :

- L'utilisateur essaye de résoudre la grille
- L'utilisateur peut utiliser l'option "joker / aide" pour se débloquer
- Une fois la grille complète, le programme vérifie automatiquement celle-ci
- Le système compare les valeurs de chaque ligne / colonne avec les cases

noircies

- Affichage d'un message indiquant s'il y a correspondance (réussite) ou non
- S'il y a correspondance :
 - Enregistrement du temps mis pour terminer la grille, suivant le profil
 - Affichage des records pour cette grille.
 - Si l'image obtenue est différente de celle stockée :
 - Le système en informe l'utilisateur et lui montre l'image en question
- Le système propose soit de tenter une nouvelle grille, soit de retourner au menu principal
- L'utilisateur choisit
- Sinon :
 - L'utilisation d'un joker peut être proposé dans le message, puisque le joueur s'en servira probablement pour repérer une erreur.

3. Exceptions :

- L'utilisateur veut utiliser un joker alors qu'il n'en a plus.

Cas d'utilisation n° 2

Nom : Édition picross

Description : L'utilisateur crée une grille de picross via l'éditeur

Acteur : L'utilisateur de l'application

Préalables : L'application doit être téléchargée

Conséquents : L'utilisateur peut exécuter l'application

1. Création de la grille :

- A l'ouverture de l'application l'utilisateur arrive sur une nouvelle grille d'édition
- Le système permet de modifier le nombre de lignes et de colonne soit avec une saisie manuelle
- Un "pointeur" situé sur une barre de longueur 25
- L'utilisateur sélectionne le nombre de lignes et de colonnes (non modifiable par la suite et multiple de 5 entre 5*5 et 25*25)
- L'utilisateur dessine son image en cliquant sur les cases de la grille pour les noircir, une 2e fois pour annuler.
- L'utilisateur clique sur l'option "enregistrer la grille".

2. Enregistrement de la grille :

- Le système demande de saisir un nom de grille (obligatoire) ainsi qu'une description (optionnel, peut aider à résoudre la grille).
- L'utilisateur saisit le nom, et éventuellement la description.
- Le système compte, pour chaque ligne et ensuite pour chaque colonne, les séries de cases noires et enregistre ces valeurs avec la grille.
- Le système propose soit de créer une autre grille, soit de retourner au menu principal.
- L'utilisateur choisit.

3 Livrables

- Un cahier des charges validé par le client
- Un logiciel qui respecte le cahier des charges
- Les cahiers d'analyse et de conception
- Un manuel utilisateur

4 Contraintes

4.1 Documentation

Document	Objectif
Cahier des charges	Spécifier les besoins du client, les fonctionnalités attendues de l'application, ainsi que les contraintes liées au projet
Dossier d'analyse et de conception	Présenter les outils et technologies utilisées dans le cadre du projet et préciser son architecture
Manuel utilisateur	Indiquer à l'utilisateur comment utiliser les différentes fonctionnalités présentes dans l'application

4.2 Délais

Tous les livrables qui accompagnent l'application, ainsi que l'application elle-même doivent être rendus le 16 Mai au plus tard.

4.3 Contraintes techniques

Le développement de l'application se fera avec ces moyens techniques listés ci-dessous :

- Compatibilité avec le langage Ruby dans sa version 1.9.3 et inférieur
- Interface graphique avec la bibliothèque logicielle GTK
- Mise en œuvre de la persistance des données à l'aide de la sérialisation
- Environnement technique (système d'exploitation, langage de programmation ...)

5 Organisation du projet

5.1 Diagramme WBS

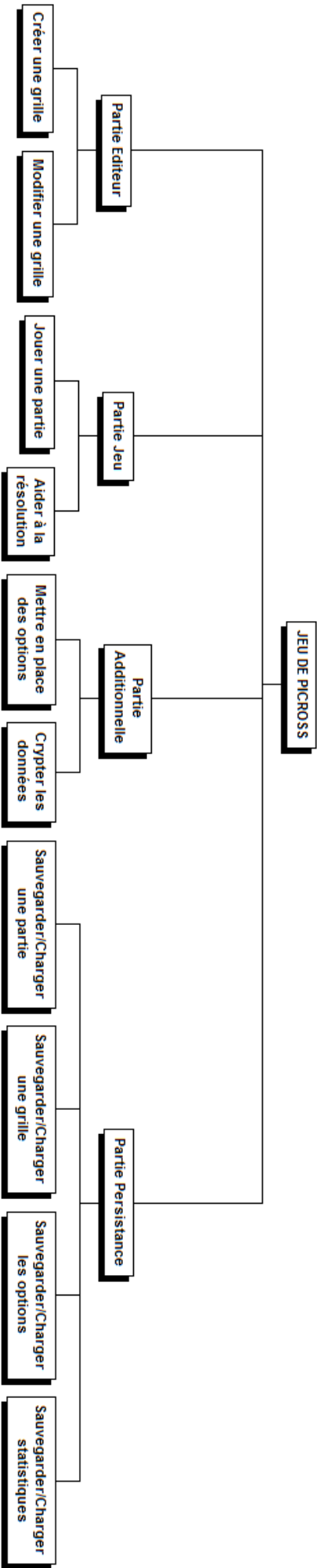


FIGURE 2 – Diagramme WBS du projet

5.2 Diagramme de Gantt













		Nom	Durée	Début	Fin	Prédécesseurs
1		Présentation du projet	0j	31/01/2014	31/01/2014	
2		Préparation du questionnaire au client	3j	01/02/2014	04/02/2014	1
3		Entretien avec le client	0j	05/02/2014	05/02/2014	2
4		Rédaction du cahier des charges	14j	06/02/2014	20/02/2014	3
5		Validation du cahier des charges par le client	0j	21/02/2014	21/02/2014	4
6		Conception v0.1	9j	22/02/2014	03/03/2014	5FD 1j
7		Implémentation v0.1	9j	03/03/2014	12/03/2014	6
8		Test v0.1	1j	12/03/2014	13/03/2014	7
9		Présentation au client v0.1	0j	14/03/2014	14/03/2014	8FD 1j
10		Conception v0.2	12j	15/03/2014	27/03/2014	9FD 1j
11		Implémentation v0.2	12j	27/03/2014	08/04/2014	10
12		Test v0.2	2j	08/04/2014	10/04/2014	11
13		Présentation au client v0.2	0j	11/04/2014	11/04/2014	12FD 1j
14		Conception v0.3	5j	12/04/2014	17/04/2014	13FD 1j
15		Implémentation v0.3	5j	17/04/2014	22/04/2014	14
16		Test v0.3	2j	22/04/2014	24/04/2014	15
17		Présentation au client v0.3	0j	25/04/2014	25/04/2014	16FD 1j
18		Conception v0.4	4j	26/04/2014	30/04/2014	17FD 1j
19		Implémentation v0.4	3j	30/04/2014	03/05/2014	18
20		Test v0.4	1j	03/05/2014	04/05/2014	19
21		Finalisation v0.4	0j	05/05/2014	05/05/2014	20FD 1j
22		Conception v1	2j	06/05/2014	08/05/2014	21FD 1j
23		Implémentation v1	2j	08/05/2014	10/05/2014	22
24		Test v1	1j	10/05/2014	11/05/2014	23
25		Présentation au client v1	0j	12/05/2014	12/05/2014	24FD 1j
26		Préparation soutenance	1j	13/05/2014	14/05/2014	25FD 1j
27		Soutenance du projet	0j	15/05/2014	15/05/2014	26FD 1j

FIGURE 3 – Diagramme Gantt du projet (partie 1/3)

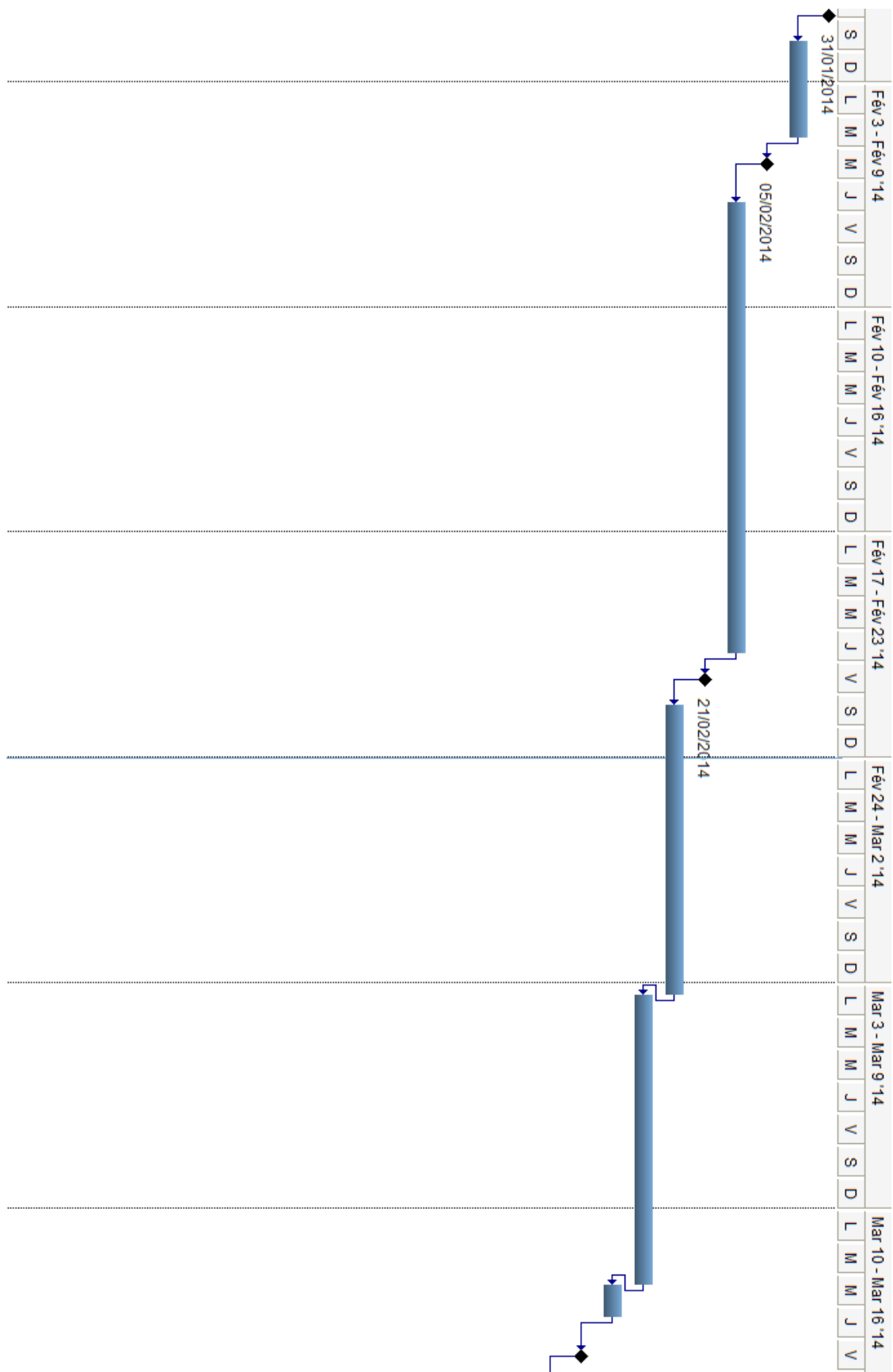


FIGURE 4 – Diagramme Gantt du projet (partie 2/3)

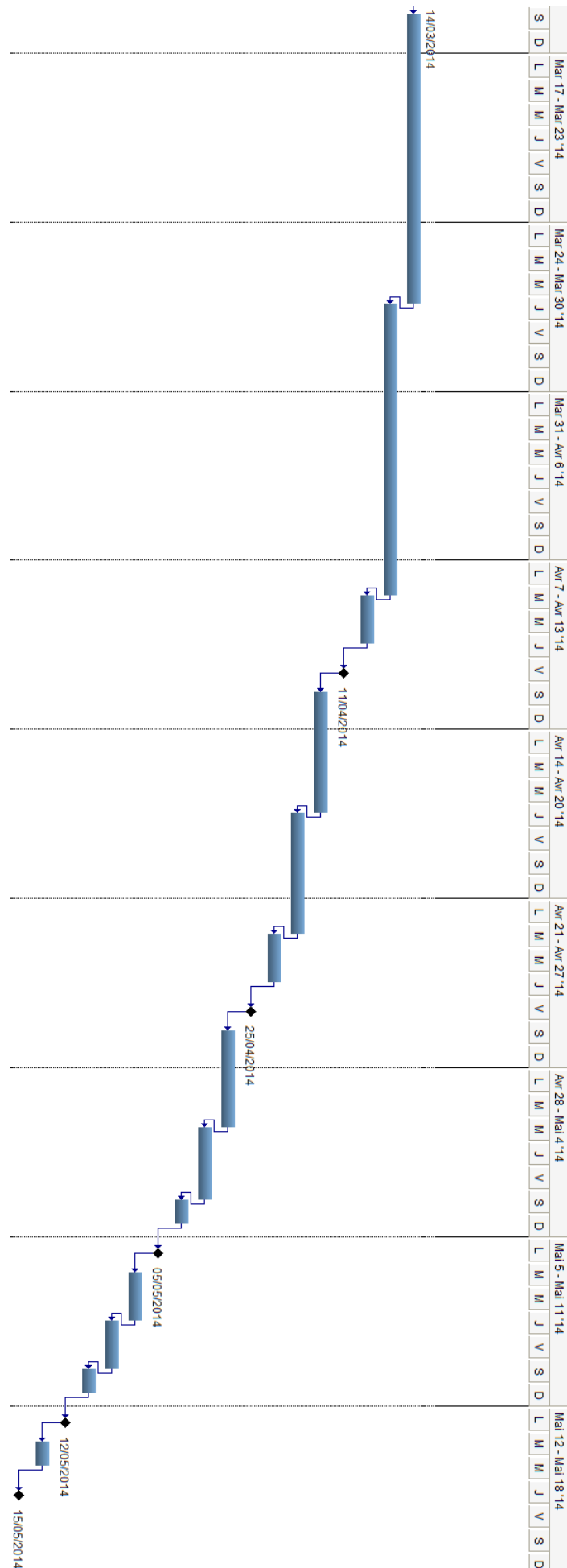


FIGURE 5 – Diagramme Gantt du projet (partie 3/3)