Tower Defense

Mise en œuvre d’un graphe interactif

Création d’une variante du célèbre jeu Tower Defense dans le cadre d’un cours d’algorithmique.

ASD

Lazhar Farjallah / Aurélien Da Campo / Pierre-Dominique Putallaz

Heig-vd

ASD

TD

Table des matières

[1 Analyse préliminaire 3](#_Toc248138382)

[1.1 Introduction 3](#_Toc248138383)

[1.2 Organisation 3](#_Toc248138384)

[1.3 Objectifs 4](#_Toc248138385)

[1.4 Planification initiale 5](#_Toc248138386)

[2 Analyse 5](#_Toc248138387)

[2.1 Cahier des charges détaillé 5](#_Toc248138388)

[2.2 Stratégie de test 5](#_Toc248138389)

[2.3 Budget initial 5](#_Toc248138390)

[2.4 Etude de faisabilité 6](#_Toc248138391)

[2.5 Compléments d'analyse éventuels 6](#_Toc248138392)

[2.6 Planification 6](#_Toc248138393)

[2.7 Historique 6](#_Toc248138394)

[3 Conception 7](#_Toc248138395)

[3.1 Dossier de conception 7](#_Toc248138396)

[3.2 Historique 7](#_Toc248138397)

[4 Réalisation 8](#_Toc248138398)

[4.1 Dossier de réalisation 8](#_Toc248138399)

[4.2 Description des tests effectués 8](#_Toc248138400)

[4.3 Erreurs restantes 8](#_Toc248138401)

[4.4 Dossier d'archivage 8](#_Toc248138402)

[5 Mise en service 9](#_Toc248138403)

[5.1 Rapport de mise en service 9](#_Toc248138404)

[5.2 Liste des documents fournis 9](#_Toc248138405)

[6 Conclusions 9](#_Toc248138406)

[7 Annexes 10](#_Toc248138407)

[7.1 Sources – Bibliographie 10](#_Toc248138408)

[7.2 Journal de bord de chaque participant 10](#_Toc248138409)

[7.3 Manuel d'Installation 10](#_Toc248138410)

[7.4 Manuel d'Utilisation 10](#_Toc248138411)

[7.5 Présentation de fin de projet (6 diapos par page) 10](#_Toc248138412)

[7.6 Archives du projet 10](#_Toc248138413)

# Analyse préliminaire

## Introduction

Ce projet de fin de semestre consiste à créer une application ludique mettant en œuvre des algorithmes et structures de données étudiées en cours. Nous avons choisi pour cela de créer une variante du célèbre jeu « Tower Defense », dans lequel des personnages se déplacent d’un point A à un point B selon un chemin optimal. En effet, on aura pour cela besoin d’une structure de graphe ainsi que les algorithmes associés, ce qui colle parfaitement avec la contrainte de départ car nous les avons étudiés en cours.

Le but est également ici de créer une application « didacticiel » qui sera présentée dans les futurs cours de cette unité d’enseignement. Ce projet permettra en effet de montrer une application réelle de l’utilisation d’algorithmes associés à des graphes (en particulier celui de recherche du chemin le plus court entre deux nœuds).

Nous commencerons par effectuer une analyse du projet, notamment en ce qui concerne l’organisation, les objectifs ainsi que la planification initiale. Puis nous détaillerons les étapes de conception et de réalisation. Enfin, avant de conclure, nous présenterons les fonctionnalités de l’application finale sous forme de mode d’emploi et nous présenterons des captures d’écran.

*Ce chapitre décrit brièvement le projet, les raisons de ce choix et ce qu'il peut apporter à l'élève ou à l'école. Il n'est pas nécessaire de rentrer dans les détails (ceux-ci seront abordés plus loin) mais cela doit être aussi clair et complet que possible (idées de solutions). Ce chapitre contient également l'inventaire et la description des travaux qui auraient déjà été effectués pour ce projet.*

*Ces éléments peuvent être repris des spécifications de départ.*

## Organisation

Les membres participant à ce projet sont :

**Etudiant 1** :

Aurélien Da campo, *aurelien.dacampo@heig-vd.ch*

**Etudiant** **2** :

Pierre-Dominique Putallaz, *pierre-dominique.putallaz@heig-vd.ch*

**Etudiant 3 (responsable de projet)** :

Lazhar Farjallah, *lazhar.farjallah@heig-vd.ch*

Nous prévoyons de nous répartir les tâches de manière suivante entre chaque entité du groupe :

1. *Responsable de projet*
   1. Rédactions, administrations
   2. Suivi des rendus (*deadlines*)
   3. Surveillance et coordination
   4. Développement
2. *Etudiant 1*
   1. Création de l’interface graphique
   2. Gestion de l’affichage
   3. Rendu graphique
   4. Interaction avec l’utilisateur
3. *Etudiant 2* 
   1. Algorithmique
   2. Implémentation des algorithmes de graphe
   3. Fournir les briques logicielles pour permettre la construction de la partie fonctionnelle de l’application

Ce chapitre décrit l'organisation du projet :

Eleve 1 : Nom, prénom, e-mail et téléphone

Eleve 2 : Nom, prénom, e-mail et téléphone

Responsable de projet : Nom, prénom, e-mail et téléphone

Expert 1 : Nom, prénom, e-mail et téléphone

Expert 2 : Nom, prénom, e-mail et téléphone

Ce chapitre peut également montrer la répartition des tâches dans le projet.   
Exemple :

|  |  |  |
| --- | --- | --- |
|  | Eleve 1 | Eleve 2 |
| Tâche 1 | X |  |
| Tâche 2 |  | X |
| … |  |  |
| Tâche n |  | X |

*Ces éléments peuvent être repris des spécifications de départ.*

## Objectifs

Les objectifs de ce projet sont les suivants :

* Illustrer le concept de graphe de manière ludique et interactive.
* Acquérir de l’expérience dans la planification et l’accomplissement d’un projet conséquent.
* Utiliser et découvrir des librairies existantes implémentant le concept de graphe
* Apprendre à mettre en œuvre une interface graphique en Java.
* Séparer le travail en plusieurs niveaux d’abstraction pour faciliter l’élaboration et l’évolutivité de ce projet.
* Comprendre la nécessité d’utiliser des algorithmes complexes dans les applications informatiques.
* Mettre en œuvre un algorithme de recherche de chemin le plus court (ACPC).
* Respecter le design pattern MVC (*Model* – *View* – *Controller*) qui structure un programme en trois couches principales.

*Ce chapitre énumère les objectifs du projet. L'atteinte ou non de ceux-ci devra pouvoir être contrôlée à la fin du projet. Les objectifs pourront éventuellement être revus après l'analyse.*

*Ces éléments peuvent être repris des spécifications de départ.*

## Planification initiale

Le projet se déroulera du 18 novembre 2009 au 15 janvier 2010, ce qui représente un total de 25 périodes en classe (par personne). Nous prévoyons également de passer un total d’environ au moins 25 périodes par personne en dehors des heures encadrées.

Au total, c’est environ 150 périodes de travail que nous allons planifier comme suit :

*Ce chapitre montre la planification du projet. Celui-ci peut être découpé en tâches qui seront planifiées. Il s'agit de la première planification du projet, celle-ci devra être revue après l'analyse. Cette planification sera présentée sous la forme d'un diagramme de Gantt et/ou de PERT (l'utilisation de MS project est conseillée).*

*Ces éléments peuvent être repris des spécifications de départ.*

# Analyse

## Cahier des charges détaillé

*Le cahier des charges complet avec toutes ses annexes:*

* *multimédia: carte de site, maquettes papier, story board préliminaire, …*
* *bases de données: interfaces graphiques, modèle conceptuel.*
* *programmation: interfaces graphiques, maquettes, analyse fonctionnelle…*

## Stratégie de test

*Décrire la stratégie globale de test:*

* *types de des tests et ordre dans lequel ils seront effectués.*
* *les moyens à mettre en œuvre.*
* *couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).*
* *données de test à prévoir (données réelles ?).*
* *les testeurs extérieurs éventuels.*

## Budget initial

*Le budget détaillé incluant tous les coûts du projet:*

* *achats de matériel, de livres...*
* *en cas de projet rémunéré : coût en main d’œuvre.*

*Si le projet n'est pas rémunéré : "budget horaire" (nombre total d'heures de travail planifiées).*

## Etude de faisabilité

*Détailler les 3 aspects de l'étude de faisabilité:*

* *risques techniques (complexité, manque de compétences, …).*
* *risques concernant le planning & les ressources humaines.*
* *risques concernant le budget.*

*Décrire aussi quelles solutions ont été appliquées pour réduire les risques (priorités, formation, actions, …).*

## Compléments d'analyse éventuels

*Inclure les compléments d'analyse s'ils ne sont pas définis dans le cahier des charges:*

* *pour un site: analyse audience & concurrence, navigation, charte graphique complète, …*
* *pour une base de données: analyse des informations à stocker.*

## Planification

*Révision de la planification initiale (Gantt et/ou PERT) du projet :*

* *planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.*
* *partage des tâches en cas de travail à plusieurs.*

*Il s’agit en principe de la planification* ***définitive du projet****. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l’historique.*

## Historique

* *Pour le cahier des charges: lister toutes les modifications demandées par le client, et aussi celles décidées pour d'autres raisons*
* *Pour le budget : comparaison entre le budget initial et le budget.*

# Conception

## Dossier de conception

**Systèmes d’exploitation**

Le système d’exploitation sur lequel nous travaillons est Windows (XP et Seven). Cependant, grâce au choix qui a été fait d’utiliser un encodage de type UTF-8 ainsi que celui du langage portable Java, nous pouvons sans soucis travailler sur un environnement Linux ou Mac par exemple.

**Outils logiciels**

Nous développons avec l’IDE Eclipse (version 3.4 et supérieure) intégrant tous les outils nécessaires au développement d’applications Java. Ce logiciel est très largement répandu dans le monde des développeurs et est très utilisé. Il possède de nombreuses fonctions spécialement conçues pour augmenter la productivité des développeurs et leur simplifier la vie (comme le *refactoring* par exemple). De plus, cette plateforme nous permet d’ajouter toute une série de plugins qui peuvent ajouter des fonctionnalités, telles que SVN (logiciel de gestion des versions du code). En ce qui concerne la génération des diagrammes de classe UML, nous utilisons le plugin ***eUML 2.0*** de chez ***Soyatec***.

**Librairies externes**

Nous utilisons la librairie externe ***JGraphT*** codée en Java. En effet, cette librairie possède toutes les briques logicielles nécessaires à la création de graphes. Nous l’utilisons comme une boîte noire sans se soucier de son implémentation.

**Programmation**

Nous avons choisi comme architecture du programme le fameux design pattern MVC (Modèle – Vue – Contrôleur).

*Fournir tous les document de conception:*

* *le choix du matériel HW*
* *le choix des systèmes d'exploitation pour la réalisation et l'utilisation*
* *le choix des outils logiciels pour la réalisation et l'utilisation*
* *site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, …*
* *bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.*
* *programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme…*

***Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !***

## Historique

*Si la conception du projet a du être modifiée plusieurs fois, ou de manière significative, expliquez ces changements et leurs causes.*

*Attention: Pour faciliter la maintenance, à la fin du projet, le dossier de conception doit correspondre à ce qui a été effectivement réalisé !*

# Réalisation

## Dossier de réalisation

*Décrire la réalisation "physique" de votre projet*

* *les répertoires où le logiciel est installé*
* *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
* *les versions des systèmes d'exploitation et des outils logiciels*
* *la description exacte du matériel*
* *le numéro de version de votre produit !*
* *programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.*

*NOTE : Evitez d’inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n’incluez que cette partie…*

## Description des tests effectués

*Pour chaque partie testée de votre projet, il faut décrire:*

* *les conditions exactes de chaque test*
* *les preuves de test (papier ou fichier)*
* *tests sans preuve: fournir au moins une description*

## Erreurs restantes

*S'il reste encore des erreurs:*

* *Description détaillée*
* *Conséquences sur l'utilisation du produit*
* *Actions envisagées ou possibles*

## Dossier d'archivage

*Décrire de manière détaillée les 2 archives du projet (CD-ROM, disque zip ou jazz, bandes magnétiques, …)*

***Attention: les documents de réalisation doivent permettre à une autre personne de maintenir et modifier votre projet sans votre aide !***

# Mise en service

## Rapport de mise en service

*Fournir une description:*

* *de l'installation du projet chez le client (pour un site web: publication chez un provider)*
* *des test officiels effectués chez le client et/ou par le client.*
* *des erreurs répertoriées   
  - description détaillée   
  - conséquences pour le client  
  - actions envisagées.*

## Liste des documents fournis

*Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions*

* *le rapport de projet*
* *le manuel d'Installation (en annexe)*
* *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
* *autres…*

# Conclusions

*Développez en tous cas les points suivants:*

* *Objectifs atteints / non-atteints*
* *Points positifs / négatifs*
* *Difficultés particulières*
* *Suites possibles pour le projet (évolutions & améliorations)*

# Annexes

## Sources – Bibliographie

*Liste des livres utilisé (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)… Et de toutes les aides externes (noms)*

## Journal de bord de chaque participant

## Manuel d'Installation

## Manuel d'Utilisation

## Présentation de fin de projet (6 diapos par page)

## Archives du projet

*CD, disquettes… dans une fourre en plastique*