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.

ASD2

Lazhar Farjallah / Aurélien Da Campo / Pierre-Dominique Putallaz

Heig-vd

ASD

TD

Table des matières

[1 Analyse préliminaire 3](#_Toc248740874)

[1.1 Introduction [LAZHAR] 3](#_Toc248740875)

[1.2 Organisation [LAZHAR] 3](#_Toc248740876)

[1.3 Objectifs [LAZHAR] 4](#_Toc248740877)

[1.4 Planification initiale [AURELIEN] 5](#_Toc248740878)

[2 Analyse 6](#_Toc248740879)

[2.1 Concepts algorithmique [LAZHAR] 6](#_Toc248740880)

[2.2 Etude de concurrentielle [AURELIEN] 6](#_Toc248740881)

[2.3 Etude de faisabilité [AURELIEN] 6](#_Toc248740882)

[2.4 Planification [AURELIEN] 6](#_Toc248740883)

[3 Conception 7](#_Toc248740884)

[3.1 Dossier de conception 7](#_Toc248740885)

[3.1.1 Systèmes d’exploitation [LAZHAR] 7](#_Toc248740886)

[3.1.2 Outils logiciels [LAZHAR] 7](#_Toc248740887)

[3.1.3 Librairies externes [LAZHAR] 7](#_Toc248740888)

[3.1.4 Architecture de l’application [AURELIEN] 7](#_Toc248740889)

[3.1.5 Interface graphique [AURELIEN] 7](#_Toc248740890)

[3.1.6 Schémas UML [PIERRE-DO] et [AURELIEN] 7](#_Toc248740891)

[3.1.7 Gestion de la concurrence [PIERRE-DO] 8](#_Toc248740892)

[4 Réalisation 9](#_Toc248740893)

[4.1 Dossier de réalisation 9](#_Toc248740894)

[4.1.1 Résultat de l’interface graphique [AURELIEN] 9](#_Toc248740895)

[4.1.2 Résultat de l’implémentation du maillage [PIERRE-DO] 9](#_Toc248740896)

[4.2 Description des tests effectués [LAZHAR] 9](#_Toc248740897)

[4.3 Erreurs restantes [LAZHAR] 9](#_Toc248740898)

[5 Conclusions *[LAZHAR]* 9](#_Toc248740899)

[5.1 Objectifs atteints / non-atteints 9](#_Toc248740900)

[5.2 Points positifs / négatifs 9](#_Toc248740901)

[5.3 Difficultés particulières 9](#_Toc248740902)

[5.4 Avenir du projet 10](#_Toc248740903)

[6 Annexes 11](#_Toc248740904)

[6.1 Sources – Bibliographie [LAZHAR] 11](#_Toc248740905)

[6.2 Journal de bord de chaque participant [LAZHAR] 11](#_Toc248740906)

[6.3 Manuel de TDA maillage [PIERRE-DO] 11](#_Toc248740907)

[6.4 Manuel d'Utilisation [LAZHAR] 11](#_Toc248740908)

[6.5 Archives du projet [LAZHAR] 11](#_Toc248740909)

# Analyse préliminaire

## Introduction [LAZHAR]

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.

## Organisation [LAZHAR]

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

## Objectifs [LAZHAR]

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.

## Planification initiale [AURELIEN]

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

## Concepts algorithmique [LAZHAR]

Le principal concept algorithmique utilisé dans ce projet est celui de graphes. En effet, ce concept, bien connu dans le domaine algorithmique, constitue un point clé dans la réalisation de notre cahier des charges. Il intervient principalement dans le problème de la recherche d’un chemin (le plus court) entre deux points donnés A et B.

Si on regarde à un niveau plus abstrait, le but premier de notre application est de lancer des objets (créatures) à partir d’un point A pour qu’ils se dirigent vers un point B selon un chemin optimal, calculé à la volée en fonction d’un graphe qui ne cesse de se modifier (dynamique). Cette simple vision est tout à fait suffisante pour comprendre les outils algorithmiques dont nous devons s’armer.

Nous allons maintenant développer plus en détails quelles sont les structures et algorithmes de graphes dont nous avons besoin.

### Graphe pondéré non orienté

Partant du constat du paragraphe 2.1, on comprend assez facilement que nous aurons besoin des structures suivantes :

* un graphe pondéré non orienté
* des nœuds dont le graphe est composé
* des arcs pour relier les différents nœuds du graphe
* un algorithme de recherche d’un chemin optimal entre deux nœuds (connu sous l’appellation « algorithme de Dijkstra »)

## Etude de concurrentielle [AURELIEN]

## Etude de faisabilité [AURELIEN]

*Détailler les 3 a****s****pects 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, …).*

## Planification [AURELIEN]

*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.*

# Conception

## Dossier de conception

### Systèmes d’exploitation [LAZHAR]

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 [LAZHAR]

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 [LAZHAR]

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.

### Architecture de l’application [AURELIEN]

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

### Interface graphique [AURELIEN]

### Schémas UML [PIERRE-DO] et [AURELIEN]

#### Maillage **[PIERRE-DO]**

#### Interface graphique (vue et controleur) **[AURELIEN]**

#### Gestionnaire du jeu (model) **[AURELIEN]**

### Gestion de la concurrence [PIERRE-DO]

*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 !***

# Réalisation

## Dossier de réalisation

### Résultat de l’interface graphique [AURELIEN]

### Résultat de l’implémentation du maillage [PIERRE-DO]

*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 [LAZHAR]

*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 [LAZHAR]

*S'il reste encore des erreurs:*

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

# Conclusions *[LAZHAR]*

## Objectifs atteints / non-atteints

## Points positifs / négatifs

## Difficultés particulières

## Avenir du projet

# Annexes

## Sources – Bibliographie [LAZHAR]

*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 [LAZHAR]

## Manuel de TDA maillage [PIERRE-DO]

## Manuel d'Utilisation [LAZHAR]

## Archives du projet [LAZHAR]

*CD, … dans une fourre en plastique*