

Programmation Orientée Objet – Concepts Avancés

Cahier des charges – SIM CITY

CARPENTIER Sylvain
MION Giovanni
PANCARTE Kenny
MBIADA Jacques Charles

1.Introduction

2.Contexte

3. Description du projet

3.1 Objectifs

3.2 Produit du projet

3.3 Fonctions du produit

3.4 Détails des fonctionnalités

3.4.1 Environnement du jeu

3.4.2 Paramètres de jeu

3.4.3 Zones

3.4.4 Budget

3.4.5 Population

3.4.6 Désastres

3.4.7 Infrastructures

3.4.8 Energie/Alimentation

3.4.9 Transports

3.4.10 Pollution/Criminalité

3.4.11 Sauvegarde et chargement de partie

4. Contraintes

4.1 Contraintes de coûts

4.2 Contraintes de délais

4.3 Contraintes d'implémentation

4.4 Déroulement du projet

5. Propriétés fonctionnelles

1. Introduction

Le but de ce projet est de développer une copie du jeu Sim City (sorti en 1989) implémentée en Scala. Ce jeu est une simulation de ville interactive où le joueur prend le rôle de maire. Il peut, selon son budget, construire de nouveaux bâtiments ou ajuster les taxes relatives au budget. De plus pour que sa ville soit en croissance démographique, il se doit de résoudre les problèmes que lui révèlent ces citoyens.

Notre groupe se compose de :

CARPENTIER Sylvain

MION Giovanni

PANCARTE Kenny

MBIADA Jacques Charles

2. Contexte

Dans le cadre de notre formation en informatique, nous sommes amenés à développer une version du jeu Sim City en utilisant le langage Scala et les notions vues tout au long de notre parcours.

Ce logiciel doit être extensible, modulable et doit ressembler le plus possible à la version Open Source de Sim City nommée Micropolis, qui peut être téléchargée à l'adresse suivante : <http://sourceforge.net/projects/micropolis.mirror/>

3. Description du projet

3.1 Objectifs

Obtenir un logiciel reprenant les concepts majeurs du jeu Sim City. Le fonctionnement de ce jeu sera décrit dans la partie 3.5(détails des fonctionnalités).

Les concepts à retenir sont une simulation de ville régit par des règles de croissance simple de manière à ce que le jeu soit ludique, une modélisation de la ville au travers de bâtiments peu nombreux et des interactions permettant au joueur au cours de la partie de modifier les caractéristiques de sa ville pour influencer son évolution.

3.2 Produit du projet

Le produit est un programme développé dans le langage Scala, capable de supporter des modifications ultérieures. Il se doit d'être pensé de manière objet en intégrant le savoir-faire des élèves membres du groupe. Les connaissances de ceux-ci en génie logiciel, architecture de logiciel et en programmation objet seront donc requises lors du développement.

3.3 Fonctions du produit

Sim City est un jeu de gestion de ville. Le jeu a pour but de simuler une ville de manière simple et amusante.

Le joueur, simulant le maire d'une ville, construit des infrastructures, lève des impôts, gère le budget, la satisfaction des citoyens, la criminalité, la maintenance des voies de circulation, le risque d'incendie. Le jeu n'a pas de fin prédéfinie, le joueur est laissé libre quand au développement de sa ville.

Plus de détails à l'adresse suivante : [http://fr.wikipedia.org/wiki/SimCity_\(jeu_vidéo,_1989\)](http://fr.wikipedia.org/wiki/SimCity_(jeu_vidéo,_1989))

Les fonctionnalités clés pour se programme sont donc la modélisation d'une ville de manière simple (peu de paramètre pris en compte), de son évolution (de ces différents paramètres et du modèle de la ville), et une interaction avec le joueur lui permettant d'influencer cette évolution.

3.4 Détail des fonctionnalités

3.4.1 Environnement du jeu

La question qui se pose est la modélisation d'une ville.

Si l'on se base sur le modèle utilisé dans le jeu Simcity on peut alors se basé sur le modèle suivant :

Une ville est un terrain délimité sur lequel sont construits des infrastructures et où vit une population.

Le jeu se déroule sur une carte interactive générée aléatoirement, composée de zones terrestres, forestières et maritimes ayant des propriétés propres.

Le temps au cours du jeu est modélisé sous forme de tour représentant un mois, impliquant l'évolution du plateau de jeu sur cette échelle temporelle en renvoyant les plaintes de la ville.

Le joueur peut construire sur ces zones des infrastructures selon des règles spécifiques (voir paragraphe 3.5.3).

Ces infrastructures vont influencer les paramètres du jeu (voir paragraphe 3.5.2) et donc influencer la partie.

En se basant sur se modèle on peut alors voir l'évolution de la ville comme une série de règles prenant en compte les paramètre du jeu tel que la population et qui induisent un évolution de ce paramètre qui est le point centrale du jeu.

3.4.2 Paramètres de jeu

Les paramètre clés du jeu sont :

- population : nombre d'habitants, lesquels sont caractérisés par leur classe sociale
- pollution : générée par certains bâtiments, elle affecte la croissance de la population
- criminalité : affecte la croissance de la population
- impôts : affectent la population, source de revenus du joueur
- investissements : argent dépensé pour l'entretien des infrastructures

De plus on peut voir que Simcity propose les paramètre suivants :

- difficulté : affecte les rentrées et sorties d'argent, et les chances de désastres
- vitesse de jeu: régler la durée d'un tour (super rapide, rapide, normal, lent , pause)

Les chapitres suivant vont détaillés une modélisation de la ville qui se rapproche de celle de Simcity.

3.4.3 Zones

Une carte vierge est constituée de zones de trois types :

- Terre : zone constructible sans restriction
- Forêt : zone constructible sans restriction absorbant la pollution
- Eau : seules les routes et rails peuvent y être construits

Au cours de la partie on peut avoir un terrain contaminé suite à une explosion nucléaire qui est inconstructible et qui augmente la pollution.

3.4.4 Budget

Le joueur dispose d'une certaine somme au départ dont il se sert pour construire des infrastructures. Cette somme augmente grâce aux taxes et sert également à l'entretien des infrastructures.

Une certaine somme est recommandée pour assurer chaque entretien. Elle est déterminée par le nombre de constructions.

3.4.5 Population

Nombre d'habitants de la ville dont l'évolution dépend des critères du jeu et limité par le nombre de zones résidentielles.

3.4.6 Désastres

Ce sont des événements déclenchés automatiquement ou par le joueur qui détruisent des bâtiments et affectent la population.

Un désastre est représenté par une certaine zone sur la carte, peut se déplacer et dure un certain temps.

Ce sont par exemple un Monstre, une Inondation, un Tremblement de Terre, un Incendie, une Tornade, une Catastrophe Nucléaire.

3.4.7 Infrastructures

Les infrastructures disponibles et leurs caractéristiques sont les suivantes :

→ zones : elles sont créées par le joueur mais se remplissent et fonctionnent d'elles mêmes. Elles augmentent la criminalité. Ces zones ont une taille de 3x3 zones de terrain et un coût de 100\$.

→ résidentielles : elles permettent d'augmenter la population de la ville et correspondent à un certain niveau de richesse (ex : classes moyennes, aisées...).

→ commerciales : elles fournissent du travail à la population (ce qui est un critère de croissance).

→ industrielles : elles fournissent du travail à la population (ce qui est un critère de croissance) mais augmentent la pollution.

→ postes : ont une zone d'effet en fonction du budget alloué. Ils coutent 500\$ et ont un taille de 3x3

→ de police : permettent de réduire la criminalité.

→ de pompiers : luttent contre les désastres

→ centrales : alimenter les bâtiments en électricité pour qu'ils soient fonctionnels, elles ont une taille de 4x4

→ au charbon : polluent beaucoup, coutent 3000\$

→ nucléaires : sont sujettes à l'explosion et coûtent 5000\$, polluent moins

→ aéroports/ports : augmentent la population et le commerce via le tourisme. Ils ont une taille de respectivement 6x6 et 4x4. Ils permettent également de dépasser un cap de population.

→ stade : permet d'augmenter la population et de dépasser un cap.

→ parcs : augmente la population, réduisent la pollution, de manière similaire aux forêts

→ routes et chemins de fer : permettent de relier les différentes infrastructures entre elles, génèrent de la pollution, se détériorent et doivent être entretenues grâce aux investissements décidés dans le budget.

→ lignes électriques : permettent de relier les infrastructures aux centrales afin de les alimenter en énergie et de les rendre fonctionnelles.

3.4.8 Energie / Alimentation

L'électricité est produite dans les centrales. Elle est nécessaire pour que les infrastructures (autres que routes et rails) fonctionnent. Il faut donc que ces infrastructures soient reliées aux centrales via les lignes électriques. Notons que deux infrastructures contigües sont reliées électriquement. Une centrale ne peut alimenter qu'un certain nombre d'infrastructures.

3.4.9 Transports

Les transports (routes et rails) forment un réseau permettant de distiller les services dispensés par les infrastructures appartenant à ce réseau les unes aux autres. Par exemple : les habitants d'une zone résidentielle vont travailler dans une zone commerciale reliée par une route. Le trafic sur ces axes est quantifié, un trafic trop élevé augmente la pollution et l'usure.

3.4.10 Pollution / Criminalité

La criminalité et la pollution sont deux facteurs apparaissant avec l'augmentation de la population et/ou des infrastructures.

Ces facteurs réduisent la croissance de la population de la ville. Ils sont modulables grâce aux stations de police et aux parcs.

3.4.11 Sauvegarde et chargement de partie

On a la possibilité de sauvegarder les paramètres d'une partie en cours puis de fermer le jeu et ensuite de pouvoir la charger ultérieurement.

4. Contraintes

4.1 Contraintes de coûts

Ce logiciel n'a pas vocation à être lucratif. Il sera développé de manière Open Source, sans licences.

4.2 Contraintes de délais

Ce logiciel devra être opérationnel pour la soutenance prévue en Janvier 2014, ce qui correspond à la fin du premier semestre universitaire.

Le déroulement spécifique est vu plus en détail dans la partie suivante.

4.3 Contraintes d'implémentation

Ce logiciel doit être développé dans le langage de programmation Scala. Il peut faire appel à des bibliothèques Java compatibles, par exemple JCurses.

4.4 Déroulement du projet

Ce projet se décompose en plusieurs phases :

- Spécification, qui donne lieu à ce cahier des charges.

- Définition de l'architecture sous formes de diagrammes de classe UML ou d'autres diagrammes dont les choix sont justifiés, contenant :

- une interprétation du sujet de manière informelle
- une définition des concepts de modélisation utilisés
- une description de l'architecture (contenant diagrammes et justifications)
- une liste des extensions envisagées permises par cette architecture.

Un rendu est attendu pour le 28 octobre 2013.

- Une première ébauche de l'implémentation (*skeleton*), contenant les abstractions de l'architecture et une ébauche de l'interface utilisateur.

Un rendu est attendu pour le 4 novembre 2013.

- Implémentation et test, débouchant sur un prototype fonctionnel et testé du logiciel.

Un rendu est attendu pour le 18 novembre 2013.

Enfin, une soutenance en janvier permettra de présenter la version finale du projet.

5. Propriétés non fonctionnelles

- Le jeu doit être programmé en Scala

- Le jeu doit être compatible avec les systèmes d'exploitation Linux et/ou Windows

- Le joueur interagit avec la carte par le biais d'une interface ergonomique et adaptée
- Possibilité de changer la difficulté du jeu au cours de la partie
- Disponibilité d'un fichier d'aide ainsi qu'un manuel d'utilisation.