



Université Libre de Bruxelles
Faculté des sciences appliquées
BA2 orientation ingénieur civil

Projet 2 d'algorithmique et programmation

Générateur de ville & Interface graphique

Delhayé Quentin, Giaux Camille

Bruxelles, 11 mai 2012

1 Introduction

Le projet numéro deux consiste à réaliser une interface graphique permettant le tracé de la ville du premier projet sans passer ni par le fichier .obj ni par le logiciel blender. La bibliothèque utilisée est Qt et l’affichage de la ville se fait via la spécification OpenGL. L’algorithme qui se charge de générer la ville doit aussi être retravaillé afin d’adopter une approche orientée objet et d’utiliser des listes liées.

2 Approche du problème

Le projet pose ainsi deux principaux problèmes : l’assimilation de la bibliothèque Qt et le remaniement du code du premier projet. Pour approcher Qt nous sommes passé par le site du zéro (<http://www.siteduzero.com/tutoriel-3-11406-programmez-avec-le-langage-c.html>) et par la documentation en ligne de nokia (<http://doc.qt.nokia.com/>). Il n’a pas été nécessaire de trouver de nouveaux documents pour remanier le code.

Voici les différentes classes qui ont été implémentées pour mettre au point l’interface graphique :

- la classe *MainWindow* : gère la saisie des paramètres et contient glWidget ;
- la classe *GlWidget* : gère l’affichage de la ville.

Les différentes classes utilisées pour générer la ville sont :

- la classe *Ville* : récupère les paramètres et gère la découpe en blocs. Elle contient aussi un objet *ListeBlocs* (blocs de la ville) et *ListeElements* (routes).
- la classe *Bloc* : crée un bloc de dimensions données par une méthode de la classe *Ville* et gère la découpe en éléments ;
- la classe *Element* : crée un élément dont les caractéristiques sont données par une méthode de la classe *Bloc* (hauteur, type, etc.) ;
- la classe *ListeBlocs* : contient la liste des blocs, chaque bloc contenant une liste d’éléments ;
- la classe *ListeElements* : contient la liste des éléments ;
- la classe *Aléatoire* : classe qui permet de regrouper tous les calculs de nombres aléatoires ;
- les classes *IterateurElement* et *IterateurBloc* : ces deux classes sont utilisées pour parcourir les listes de blocs/éléments avec OpenGL.

3 Répartition du travail

Lors de ce projet nous avons travaillé en deux temps. Une partie du travail a d'abord été partagée : Quentin a ainsi réalisé le plus gros de l'algorithme pour générer la ville et Camille a fait la majeure partie de l'interface graphique. Nous avons ensuite travaillé simultanément sur le rendu OpenGL.

4 Améliorations

Ce projet ouvrait grand la porte aux améliorations. Nous en avons réalisé plusieurs. Des aperçus se trouvent dans la section 5.

- La largeur des routes est variable.
- Certains bâtiments possèdent des toits.
- Les blocs de bâtiments sont de couleurs différentes.
- Utilisation de textures pour le rendu (des jardins, routes et toits).
- La ville peut évoluer au cours des saisons !
- Tapez « ulb ».

5 Aperçu du rendu final

Ci-suivent quelques rendus de notre travail. Notez les différentes textures pour les jardins et les toits en fonction des saisons.

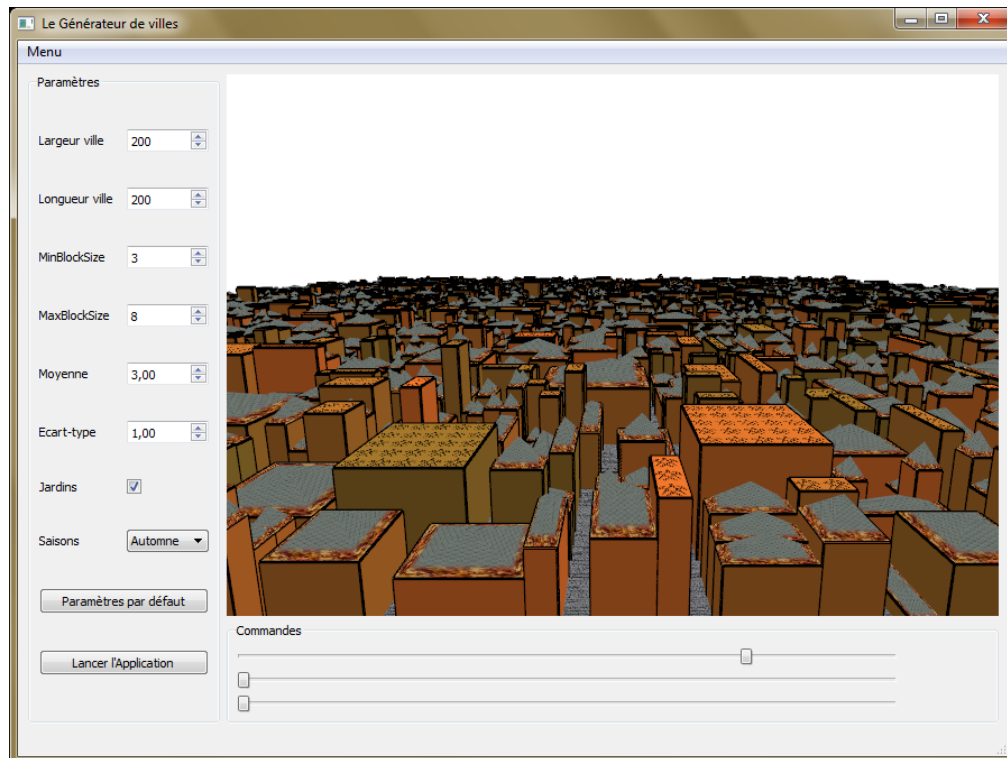


FIGURE 1 – Dimensions 200x200, Automne

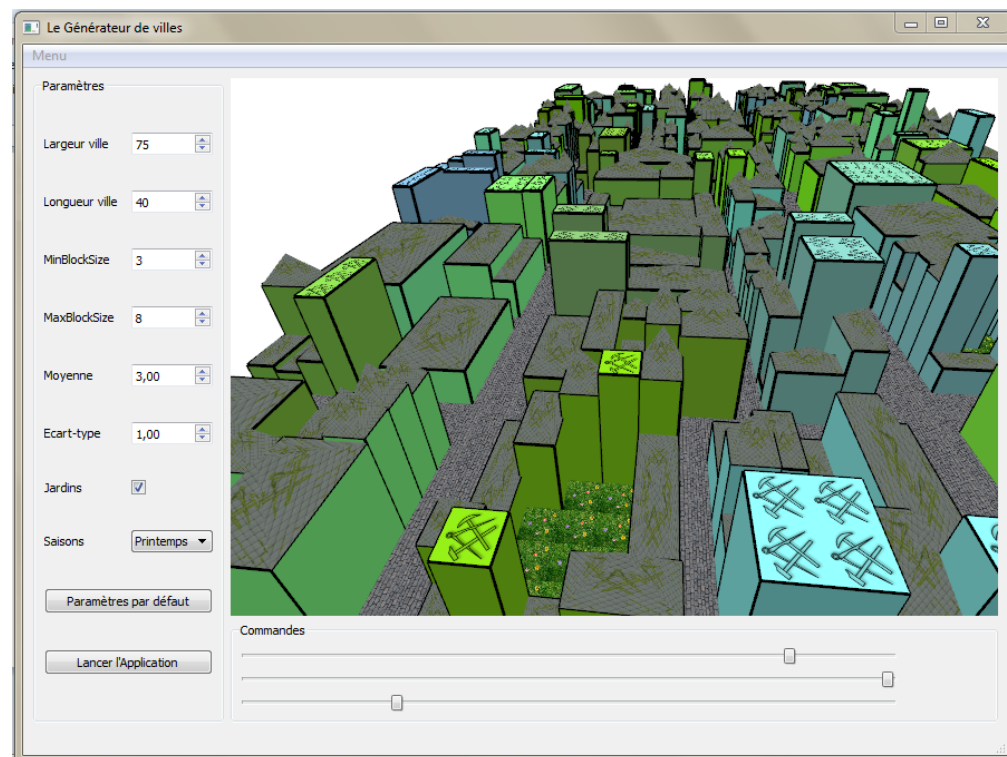


FIGURE 2 – Paramètres par défaut, Printemps

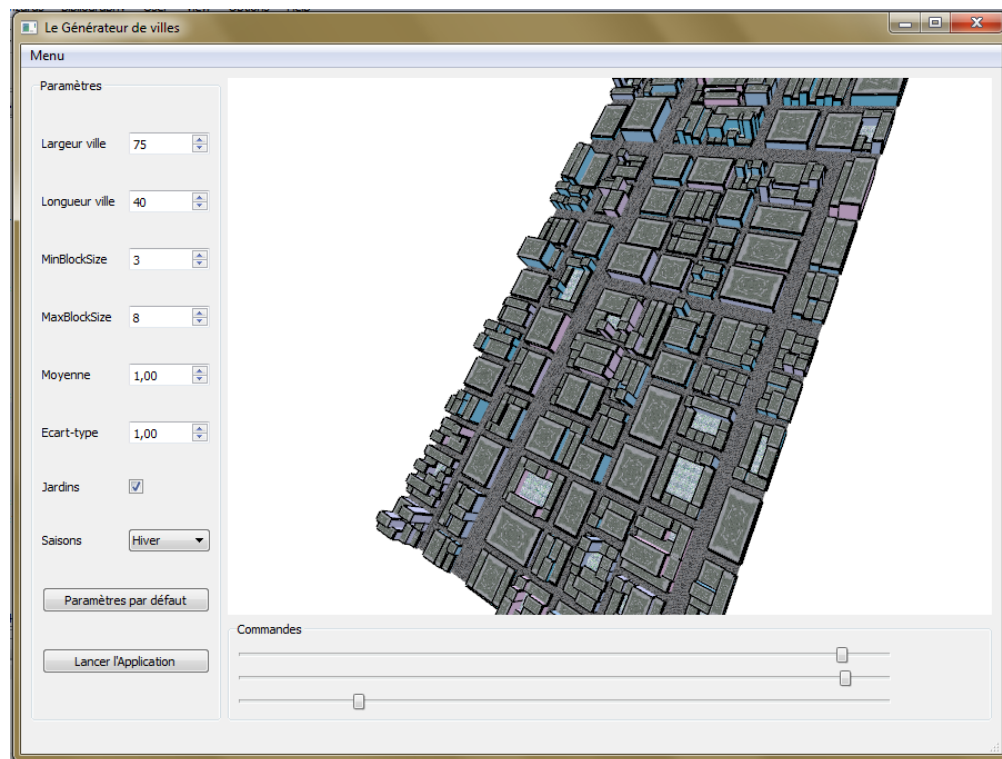


FIGURE 3 – Mu très petit, Hiver