Géométrie numérique
M1 I3D et IIRVIJ – 2018-2019
Projet

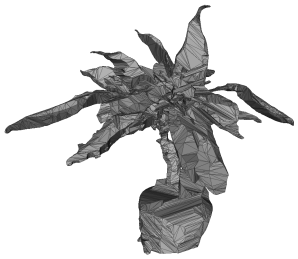
Segmentation de maillages

1 Sujet

Il vous est demandé de développer un algorithme de **segmentation** de maillages en régions. Plus spécifiquement, votre client est un botaniste qui souhaite **décomposer des plantes** de diverses espèces, tailles et complexités **en leurs branches ou tiges et leurs feuilles** (une région par tige et une région par feuille). Votre algorithme doit être **entièrement automatique** et réaliser un bon compromis entre performance en temps de calcul, en place mémoire et qualité des résultats. Le client souhaite également que le **nombre de paramètres** à fournir en entrée soit le plus faible possible, et que le résultat soit le moins sensible possible au choix des valeurs pour ces paramètres.

Le client vous fournit des exemples de maillages pour vos tests. Ces maillages sont de tailles et de qualités variées (en particulier, ils représentent tous des 2-variétés mais certaines sont à bord). Il est également disponible régulièrement pour répondre à vos questions sur ses attentes (voir section 2).





2 Organisation

Pour ce projet, vous expérimenterez une **approche de type "agile"** (inspirée plus particulièrement du schéma *Scrum*). Vous travaillerez en **équipes de 4 à 5 étudiants**, en temps libre. Les 5 séances prévues à l'emploi du temps permettront de faire des points d'avancement avec le client (voir section 2.2). Un état de l'art des méthodes de segmentation de maillages [1] vous est fourni comme point de départ.

2.1 Agilité

Cette section est reprise et adaptée d'un texte de Matthieu Moy.

L'agilité est un processus de production logicielle centré *produit* plutôt que *contrat*. La production est organisée en **sprints**, théoriquement de durée fixe. En début de sprint on choisit les fonctionnalités à développer, en fin de sprint on les livre. À la fin d'un sprint ce qui est livré est terminé et on n'y revient plus, on recommence un autre sprint avec un nouveau choix de fonctionnalités. L'organisation typique d'un sprint est la suivante :

1. Au démarrage : réunion de *planification de sprint* ;
2. Au quotidien : réunion "*stand-up*" tous les matins, mise à jour du graphique d'avancement ;
3. En fin de sprint : *démonstration* de fin de sprint puis *rétrospective*.

Il n'y a pas de longue période de spécification en amont. On passe très vite en mode production. Sur cette base, on admet forcément que les besoins ne sont pas clairement exprimés dès le départ et qu'ils peuvent même changer en cours de réalisation. En contre-partie, pour ne pas se perdre, on a besoin de bons repères, et c'est au **responsable de produit** de les instruire :

— définir la liste des fonctionnalités à mettre dans le produit. C'est ce que l'on appelle le **carnet de produit** (*backlog* en anglais). Les fonctionnalités sont appelées des **récits**, parce-qu'elles sont rédigées sous forme d'histoire pour un utilisateur. Chacune des fonctionnalités doit être formulée de façon synthétique, les détails seront discutés ensemble avec l'équipe ;

— garder la liste de fonctionnalités **ordonnée** : les récits les plus importants en début de liste, les choses moins importantes en fin. C'est primordial parce que dans un projet Agile on ne fait jamais tout ce qui est prévu, et on fait toujours des choses que l'on avait pas prévues ;

— Pour celles qui sont en tête, **préciser quelles seront les conditions de satisfaction**, en gros quels tests permettront de vérifier à la livraison que l'objectif fonctionnel est OK.

2.1.1 Planification de sprint

L'objectif de cette réunion est de planifier le contenu du sprint à venir. Le responsable de produit prépare en avance une liste de récits (le carnet de produit), ordonnées par priorité. En fin de réunion l'équipe doit disposer :

— d'une liste des récits sélectionnés pour ce sprint, avec une estimation en *points de récits* pour chacune. Le point de récit est une unité de quantité de travail arbitraire, un récit à 5 points sera a priori plus longue à réaliser qu'un récit à 3 points. Cette liste correspond à un engagement mutuel entre l'équipe et le responsable de produit ;

— si elle n'a pas été fixée avant, une définition de *fini* (on n'a pas de notion de "à moitié fini" : quelque chose est fini, ou pas fini).

L'équipe peut travailler sur le découpage des récits en tâches techniques (mais les tâches techniques sont de la responsabilité de l'équipe, et non du responsable de produit).

Voici quelques conseils :
Pour l'estimation de la quantité de travail par récit :

— faire un *planning poker* : voir `https://fr.wikipedia.org/wiki/Planning_poker` pour les détails. Il existe un outil en ligne : `https://www.planningpoker.com/` ;

— utiliser les résultats des sprints précédents pour étalonner la valeur d'un point de récit ;

— attention à ne pas sous-estimer le travail, et à prendre en compte la définition de "fini" dans l'évaluation. Une tâche d'apparence triviale peut prendre beaucoup de temps si la définition de "fini" inclut la documentation, les tests, l'acceptation par les utilisateurs, etc. Une sous-estimation mène à un surengagement pendant la planification de sprint, et donc à des problèmes en fin de sprint.

Pour la sélection des récits :
La liste des récits les plus prioritaires ne donne pas forcément une quantité de travail raisonnable pour un sprint. Il arrive souvent qu'un récit soit trop gros pour tenir dans le sprint, mais que le carnet de produit soit trop petit si on ne l'inclut pas. Parmi les variables d'ajustement :

— Modifier les priorités. Par exemple, le responsable de produit peut penser qu'une tâche est facile et la considérer comme prioritaire, mais décider de la remettre à plus tard si l'équipe la considère difficile ;

— Modifier la portée d'un récit. Par exemple, on peut demander une preuve de concept au lieu d'une version déployable en production pour réduire le nombre de points de récits ;

— Découper les récits ("on fait une partie pendant ce sprint, et on garde une autre partie pour le sprint suivant"). Même découpés, les nouveaux récits doivent garder un sens en valeur client. Il ne s'agit pas de découper en sous-tâches techniques. Même si un "gros" récit peut tenir dans le sprint, c'est intéressant de le couper en plus petits récits pour mieux pouvoir répartir le travail en cours de sprint, et limiter les risques en cas d'échec.

2.1.2 Réunion "stand-up" et graphique d'avancement

Comme son nom l'indique, lors d'une réunion "stand-up" on est debout ! Il s'agit d'une réunion **courte** (5 à 15 minutes) où les membres de l'équipe interagissent généralement autour d'un tableau ou autre support de même type. Chaque participant résume rapidement (quelques minutes) où il en est dans le développement du projet (travail réalisé depuis la dernière réunion, travail en cours et prévu d'ici à la prochaine réunion "stand-up", points de blocage). L'équipe réfléchit

ensuite **collectivement** aux moyens de résoudre les problèmes en cours. L'intérêt de ce type de réunion est d'informer tout membre de l'équipe des progrès et blocages des autres, ce qui permet de favoriser l'intégration et aux membres d'avoir une vue globale sur le projet. Ceci conduit à la fois à l'identification précoce de problèmes potentiels et à la recherche rapide de solutions. Si un point demande plus de discussions, il est conseillé non pas de prolonger la réunion "stand-up" mais de planifier une autre réunion dédiée.

Un outil utile pour visualiser les progrès de l'équipe dans le développement du projet est le graphique d'avancement (*burndown chart* en anglais). Les axes de ce graphique sont le temps en abscisse et le travail restant à réaliser en ordonnée. La quantité de travail est estimée grâce aux points de récits (voir Section 2.1.1). L'objectif étant d'arriver à une quantité nulle à la fin du sprint f, la droite passant par le point d'abscisse 0 et d'ordonnée la quantité de travail totale pour le sprint et par le point d'abscisse f et d'ordonnée 0 indique le travail restant "idéal" à chaque instant. Le graphique d'avancement est mis à jour à chaque réunion afin de voir immédiatement si la quantité de travail réalisée depuis la réunion précédente est suffisante.

2.1.3 Démonstration de fin de sprint

À la fin d'un sprint, l'équipe fait une démonstration devant son responsable de produit. L'équipe doit avoir préparé une démonstration. La préparation ne doit pas être trop longue, mais doit être suffisante pour éviter l'"effet démo" : ce qui est terminé doit marcher, ce qui ne marche pas n'est pas terminé et ne sera donc pas validé pour ce sprint. À la fin de la démonstration, on compte les points des récits validés par le responsable de produit. Le nombre de points de récits est la somme des points de chaque récit validé. Éventuellement, on liste également les récits non validés qui peuvent alimenter le carnet de produit pour les sprints à venir.

2.1.4 Rétrospective

La rétrospective, ou bilan, est une étape importante en fin de sprint. Pour un sprint court, une rétrospective de 30 minutes est suffisante. Elle se déroule en trois temps successifs, d'environ 10 minutes chacun :

1. La **collecte d'information** : chaque membre de l'équipe écrit les choses qui l'ont marqué sur des post-its (un post-it par idée). Pour susciter des idées, on peut les classer :
   — bon/mauvais : on place les post-its sur une feuille de paper-board à deux colonnes, typiquement avec un Smiley " :-)" en tête d'une colonne, et un Smiley " :-(" en tête de l'autre ;
   — par couleur (éventuellement) : par exemple, si on attribue une couleur de post-its à ce qui a trait aux émotions ressenties, certaines idées peuvent ressortir alors que les membres de l'équipe n'auraient pas forcément pensé ou osé en parler.
2. La **discussion** : les membres de l'équipe discutent des idées notées par les autres et comparent leurs points de vue.
3. La **liste d'actions** : en conclusion, l'équipe écrit une liste d'actions **concrètes**, à appliquer au prochain sprint. Lors de celui-ci, on doit pouvoir répondre à la question "cette action a-t-elle été faite" simplement par "oui" ou "non". Par exemple, "mieux tester" n'est pas une action vraiment concrète, alors qu'"utiliser un outil de couverture de code" en est une. On peut aussi rester un peu plus abstrait en classant les idées en 5 catégories : "plus de ...", "moins de ...", "commencer ...", "arrêter ...", "garder ...".

2.2 Séances prévues à l'emploi du temps

L'organisation décrite ci-dessus, classique pour les projets à plein temps, va être adaptée dans le cadre du projet de Géométrie Numérique :

— le responsable de produit sera l'un des étudiants de l'équipe ;
— les sprints ne seront pas de durée fixe ;
— un sprint commencera et se terminera lors d'une des séances prévues à l'emploi du temps (voir section 2.2) ;
— les réunions "stand-up" sont laissées à votre charge, à vous de les organiser comme bon vous semble, par exemple une fois par semaine.

La première réunion vous servira à définir avec le responsable de produit la première version du carnet de produit. Vous estimerez également tous ensemble l'effort nécessaire pour réaliser ces récits, ce qui permettra de planifier et lancer le premier sprint. Les séances suivantes (sauf la dernière) enchaîneront dans l'ordre, et en présence du responsable de produit :

1. démonstration de fin de sprint ;
2. rétrospective ;
3. planification du sprint suivant.

Des post-its vous seront fournis pour que vous puissiez les utiliser pour organiser vos récits (on parle de *tableau des tâches*, voir `http://referentiel.institut-agile.fr/taskboard.html`). La dernière séance sera consacrée aux soutenances orales des projets (voir section 2.3).

Pour rappel, les séances sont prévues aux dates suivantes :
- séance 1 (définition du carnet de produit et lancement du premier sprint) : mercredi 20 mars ;
- séance 2 (fin du premier sprint et début du deuxième) : mercredi 03 avril ;
- séance 3 (fin du deuxième sprint et début du troisième) : mercredi 10 avril ;
- séance 4 (fin du troisième sprint et début du quatrième) : mercredi 24 avril ;
- séance 5 (soutenances) : mercredi 22 mai.

Les sprints seront donc de durées respectives : 2 semaines, 1 semaine, 2 semaines et 4 semaines.

2.3 Soutenance

Vous présenterez votre travail le mercredi 22 mai devant un jury composé des deux enseignants de T.P. et éventuellement d'autres enseignants ou chercheurs. Chaque équipe disposera de **15 minutes** pour expliquer le travail réalisé et faire une démonstration logicielle, explications qui seront suivies de **10 minutes** de questions et d'échanges avec le jury. Il vous est en particulier demandé d'expliquer :
- vos choix techniques et en quoi ils répondent aux besoins du client ;
- la répartition du travail entre les membres de l'équipe et le déroulement du projet ;
- les tests effectués afin de valider le travail ;
- les limites de la méthode proposée et les perspectives (qu'auriez-vous fait de plus si vous aviez eu plus de temps ?).

3 Travail à rendre

Vous devez poster sur Moodle avant le mercredi 22 mai 20h00 une archive au format `.tar.gz` contenant :
- un document synthétique au format électronique `.pdf` (voir ci-dessous) ;
- les sources **dûment commentées** de votre réalisation en langage C++. **Un code qui ne compile pas ne sera pas accepté**.
- le support de votre présentation orale.

3.1 Document synthétique

Il vous est demandé un document synthétique tenant sur **au maximum cinq feuilles recto-verso, soit dix pages**, où vous ferez figurer :
- une description technique de l'algorithme implémenté ;
- l'organisation de votre code et vos choix d'implémentations ;
- une description des tests de validation et d'évaluation de l'algorithme que vous avez effectués, ainsi que les résultats pertinents ;
- votre avis argumenté à la fois sur l'algorithme (avantages, limitations) ainsi que sur le projet (ce qu'il vous a apporté, vos regrets, etc.).

3.2 Barème indicatif

- Performances de l'algorithme : 5 points
- Code et choix d'implémentation : 5 points
- Document synthétique : 5 points
- Soutenance : 5 points

Bibliographie

1. A. Shamir. *A Survey on Mesh Segmentation Techniques*. Computer Graphics Forum 27 (6), pp. 1539–1556, 2008. `ftp://ftp.idc.ac.il/Faculty/arik/ShamirPapers/SegmentShamir.pdf`.