

Léonie Leroux

Brice Labbé

Kerrien Deneuille

Lila Cassan

Rapport d'analyse

Version 2. 03/05/2024

PDI 10 - Prototype et synergies Meshroom X MicMac

Table des matières

Glossaire	1
Contexte du projet	2
Description de l'outil	3
Objectifs de l'étude	4
Objectifs	4
Contraintes	5
Recueil du besoin	6
Analyse fonctionnelle - solutions proposées	7
Étude technique : choix des logiciels et des langages - architecture	8
Réalisation et suivi du projet	9
Risques	9
Planning prévisionnel	10
Conclusion	10
Annexe 1 : nœuds MicMac implémentés dans Meshroom au début du projet	11

Glossaire

Architecture nodale	Désigne ici une structure composée de nœuds représentant des fonctions, et nommés grâce à un identifiant unique, permettant d'exécuter à nouveau certaines parties de la suite de commandes sans avoir à la relancer entièrement.
Consortium 3D SHS	Regroupement de laboratoires de recherches à l'échelle nationale pour favoriser l'utilisation de données 3D en sciences humaines et sociales
Noeuds	Pour Meshroom, un nœud correspond à une tâche à exécuter : ici, cela correspond souvent à une commande MicMac.
Template	C'est un modèle qui peut être réutilisé pour exécuter une tâche similaire, par exemple ici le traitement d'un certain type de jeux d'images, dans un même contexte.
Pipeline	Séquence de commandes

Contexte du projet

Notre projet MicMacRoom a été commandité par Monsieur Anthony Pamart, ingénieur d'études à l'UPR modèles et simulations pour l'architecture et le patrimoine du CNRS, à Marseille.

Ce projet fait suite à une initiative du Consortium 3D SHS, qui cherche à rendre accessible les commandes du logiciel MicMac, logiciel libre de photogrammétrie permettant de réaliser des nuages de points 3D à partir de photos et développé par l'Institut National de l'Information Géographiques et Forestières (IGN). L'objectif est de rendre ces commandes accessibles grâce à l'interface graphique d'un autre logiciel open-source de photogrammétrie développé par Technicolor / MIKROS image, Meshroom, pour permettre une utilisation plus large de Micmac par des utilisateurs familiers de la photogrammétrie, mais préférant une solution facile à prendre en main. L'architecture de Meshroom permet de plus une certaine modularité, du fait de l'ajout possible et simple de nœuds. De plus, le logiciel MicMac reste un outil assez puissant mais utilisable majoritairement en ligne de commande, et reste jusque-là limité aux utilisateurs ayant une certaine dextérité informatique.

Dans les domaines d'application de la photogrammétrie tels que l'architecture ou la conservation du patrimoine, les logiciels utilisés sont souvent des solutions propriétaires et/ou sous licence commerciales, comme Metashape, entraînant souvent des contraintes de gestion et d'accès pour les utilisateurs. Par ailleurs, du fait du contexte international, Metashape étant développé par une entreprise Russe, il n'est plus possible pour les institutions publiques d'avoir accès à de nouvelles licences, créant ainsi un besoin pour les nouveaux utilisateurs. Le projet MicMacRoom souhaite donc répondre à un double enjeu: développer une nouvelle solution open-source pour les nouveaux utilisateurs de logiciels photogrammétriques et rendre plus accessible l'utilisation de commandes MicMac grâce à une interface graphique stable, intuitive et évolutive.

Pour mettre en œuvre cette initiative, un hackathon du Consortium 3D SHS a eu lieu en juillet 2023. Une première implémentation de l'intégration de MicMac a pu alors voir le jour, sans être pour autant stabilisée et utilisable à grande échelle. Il a aussi été proposé un script Python permettant de réduire le temps d'ajout d'un nœud en automatisant en partie le processus d'écriture du code à partir de l'aide de MicMac.

En simplifiant l'accès à MicMac et en l'intégrant dans un environnement visuel comme Meshroom, ce projet vise à offrir une solution puissante mais facile à prendre en main, sans les restrictions associées aux logiciels sous licence (coût, disponibilité...). Un apport supplémentaire pourra être tiré de l'écriture de tutoriels pour favoriser le passage à cette solution et l'exploitation de ses possibilités.

Il est difficile d'estimer un coût pour notre projet, qui s'évaluerait plutôt en heures et qui correspondra plus vraisemblablement au temps que l'on peut accorder au projet (tant sur les journées dédiées au travail sur le projet : 12 en total, c'est-à-dire 72 heures, que sur notre temps

libre). Le coût financier sera nul ou très faible (déplacement sur Paris) car l'objectif est de travailler pour une solution open source.

Suivant le même parcours d'étudiants en géomatique depuis presque deux ans, nous avons acquis des compétences similaires en programmation en Python, et dans l'utilisation du logiciel MicMac. Nous n'avons par contre pas eu l'occasion d'utiliser le logiciel Meshroom avant le début de ce projet. Le choix de la cheffe de projet, Léonie Leroux a été fait sur une base volontaire. Kerrien Deneuville a de plus eu l'occasion, par son parcours d'étudiant géomètre, d'utiliser MicMac de façon poussée et pourra donc apporter des connaissances techniques précieuses au début du projet.

Description de l'outil

Notre projet consiste donc à stabiliser les fonctions déjà présentes en les testant grâce à différents jeux de données, ajouter de nouveaux nœuds et de nouvelles pipelines, et développer de nouveaux tutoriels pour permettre une prise en main facile des nouveaux utilisateurs. Cette partie du rapport a pour objectif de décrire l'architecture qui a été mis en place par le Consortium et sur laquelle nous allons nous baser pour stabiliser les pipelines.

Une vingtaine de commandes MicMac ont été implémentées dans l'interface Meshroom sous la forme de nœuds grâce à des fichiers python utilisant l'orienté objet. Chaque fonction est implémentée en tant que classe héritant que la classe MicMacNode et possède plusieurs attributs:

- `commandLine`: chaîne de caractère qui retranscrit la ligne de commande nécessaire pour lancer la commande MicMac dans un terminal.
- `description`: chaîne de caractère décrivant brièvement la commande
- `inputs`: liste des différents paramètres de la commande qui peuvent être eux-mêmes de différents types. Chaque paramètre est décrit par son type (nom de dossier, booléen, float...), son nom, son label sur l'interface graphique, une courte description et une valeur. La valeur du paramètre est soit laissée vide, soit pré-rempli pour intégrer une valeur par défaut. Un booléen "`advanced=true`" peut être ajouté si on souhaite que le paramètre ne soit pas tout de suite visible dans l'interface
- `outputs`: liste des différents fichiers de sortie de la commande

Enfin, ces commandes sont utilisées sous la forme de pipelines implémentées grâce à des fichiers `mg`. Ces fichiers possèdent à chaque fois une entête permettant de donner la version de pipelines, puis liste toutes les commandes utilisées dans la pipeline. Pour chaque commande utilisée, plusieurs informations doivent être renseignées: le type de nœuds (classe qui définit la commande), sa position dans la pipeline au niveau de l'interface graphique ainsi que les inputs (inputs qui peuvent reprendre les outputs de précédentes commandes). Trois pipelines ont déjà été implémentées: `MicmacBasic`, `MicmacOrtho` et `MicmacTexturedMesh`.

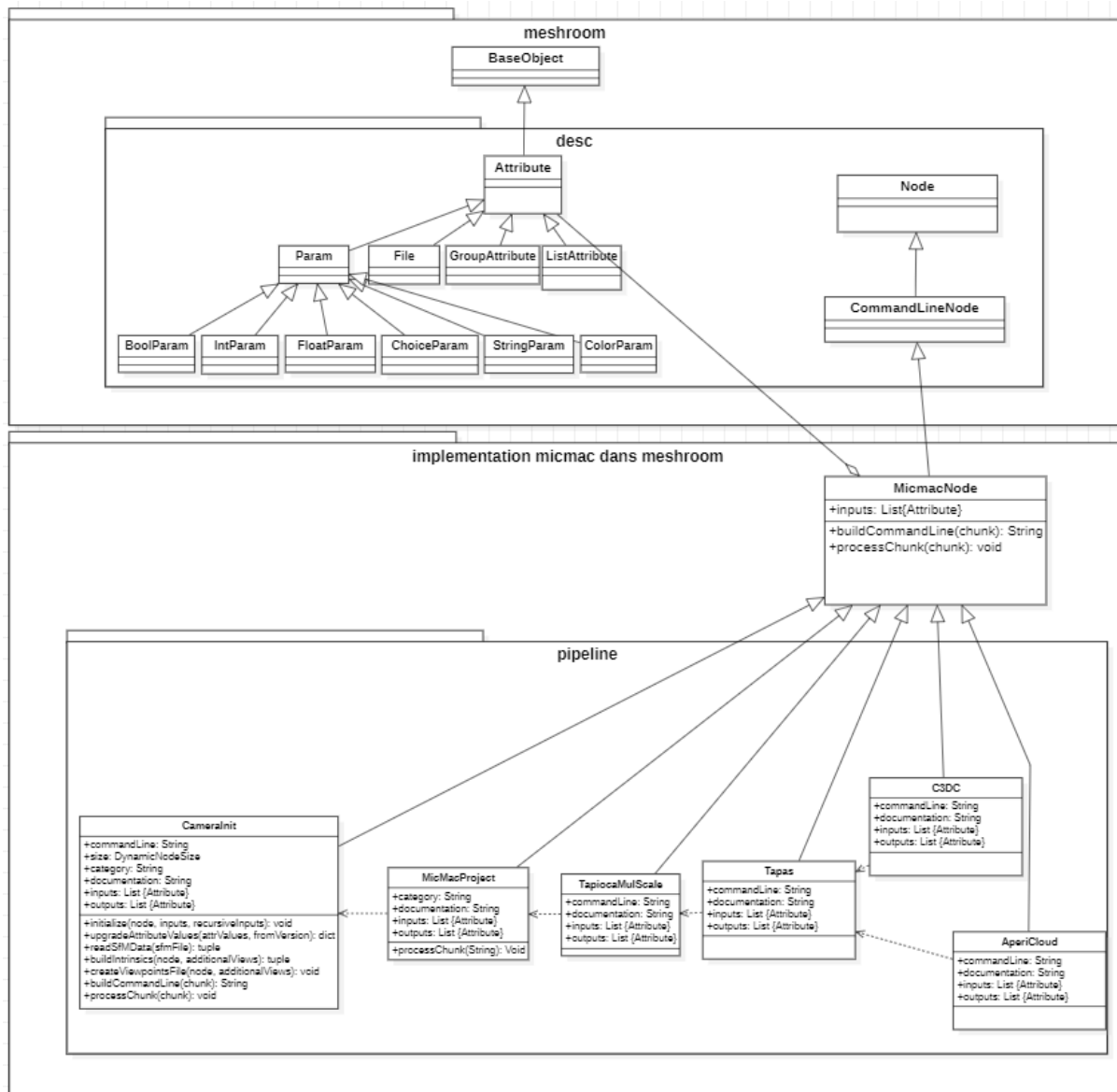


Figure 1 : diagramme de classes de l'implémentation actuelle de Meshroom X MicMac

Objectifs de l'étude

Objectifs

- **Tester et déboguer la version actuelle ses commandes MicMac disponibles actuellement dans Meshroom** à partir du test dans ce logiciel des tutoriels disponibles sur micmac.ensg.eu. Tester aussi différents jeux de données fournis par le commanditaire (menhir, dolmen).
- **Créer des templates fonctionnels.**
- **Définir et ajouter les fonctionnalités MicMac à intégrer dans l'interface Meshroom.** Mener une analyse approfondie des besoins des utilisateurs, dans les domaines de l'architecture et de l'archéologie en particulier. Sélectionner les paramètres utiles pour permettre une plus grande facilité d'utilisation.
- **Chercher à élaborer une architecture technique permettant l'intégration de Meshroom et de MicMac.** Améliorer la compatibilité et l'interopérabilité entre Meshroom et MicMac en observant le type et l'organisation des fichiers.
- **Améliorer l'expérience utilisateur en proposant un Wiki et un ensemble de tutoriels pour apprendre ou approfondir son utilisation de l'interface Meshroom.**

Contraintes

Le projet se tient du 7 février 2024 au 15 mai 2024, avec une soutenance prévue le 13 ou le 14 mai, et une présentation le 15 mai.

Délais	7 février - 13 mai
Budget	Aucun
Équipement logiciel	MicMac, Meshroom, solutions open source
Langage de programmation	Python

Une contrainte majeure de ce projet est l'absence actuelle de synergie réelle entre Meshroom et MicMac, qui a été mise en évidence durant le hackathon. Cette contrainte tient à l'architecture de MicMac, différente de celle de Meshroom, et au format des fichiers, différent dans les deux cas. Par exemple, lorsque l'on exécute une commande MicMac, les fichiers en sortie sont écrits à l'endroit où étaient stockées les images, tandis que Meshroom stocke tout au dans des dossiers différents pour chaque nœud (et aussi pour chaque changement de paramètre) . Les types de fichiers sont différents aussi : le nuage de point (.ply) en sortie de MicMac contient l'emplacement et l'orientation des caméras, et ce n'est pas le cas pour le .abc en sortie de Meshroom. Il n'est donc pour l'instant pas possible d'afficher directement le résultat d'une suite de commandes MicMac dans le visionneur de Meshroom.

Recueil du besoin

L'utilisation de la photogrammétrie dans les domaines des sciences humaines, et en particulier pour l'architecture et l'archéologie répond à des besoins de diminution des coûts, de facilité d'utilisation, et d'efficacité par rapport à l'utilisation d'un LiDAR par exemple. L'intégration de MicMac dans Meshroom doit alors être relativement intuitive pour des utilisateurs basiques tout en garantissant la possibilité d'ajuster des paramètres fins pour un utilisateur avancé. Un enchaînement de commandes basique devrait ainsi pouvoir être lancé facilement, et posséder des options avancées. Pour des institutions de recherche, il est nécessaire d'avoir un outil gratuit, qui permette au plus grand nombre de s'en servir, sans avoir à se soucier du nombre de licences attribuées. Comme la photogrammétrie est ici un outil, il est nécessaire d'avoir des informations claires permettant d'aboutir rapidement à des résultats lors des premières utilisations. Les modèles devront aussi être de qualité suffisante pour permettre des détections de motifs, des mesures. Il serait aussi souhaitable d'arriver à un fichier contenant le nuage de point qui soit intégrable dans d'autres logiciels et facilement visualisable afin de garantir des collaborations possibles.

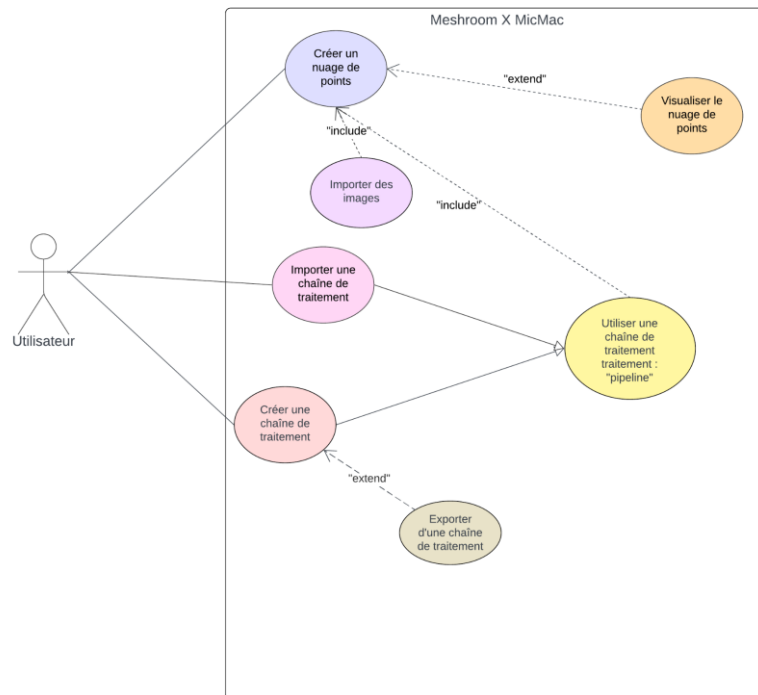


Figure 2 : Diagramme de cas d'utilisation

Dans le cadre de l'intégration de MicMac dans Meshroom, l'utilisateur cherche principalement à obtenir un fichier contenant un nuage de points créé à partir des images qu'il importe dans Meshroom. Pour ce faire, il dispose d'un ensemble de fonctionnalités : il peut importer facilement ses images dans l'interface de Meshroom, en suivant la procédure habituelle, qui seront les données d'entrée. Ensuite, il a la possibilité d'utiliser des chaînes de traitement (pipelines) qui ont été créées soit par d'autres utilisateurs, soit par lui-même lors d'utilisations antérieures. Il peut par ailleurs modifier et sauvegarder ses chaînes de traitement pour les adapter à ses besoins d'automatisation du processus.

L'utilisateur de Meshroom X MicMac cherche à aboutir à un fichier contenant un nuage de points / maillage texturés / orthophotos (ou tout autre produit photogrammétrique) à partir d'images qu'il importe dans l'interface graphique de Meshroom. Pour cela, il peut utiliser une chaîne de traitement déjà présente (créée par d'autres utilisateurs ou par l'utilisateur lors d'utilisations présentes) (Figure 3.), qu'il peut modifier pour répondre à ses besoins, et dont il peut ajuster les paramètres. Il doit ensuite pouvoir visualiser le résultat du traitement, si possible dans l'interface graphique. Une fois le traitement terminé, il est nécessaire pour l'utilisateur de visualiser le résultat, au sein même de l'interface de Meshroom.

Analyse fonctionnelle - solutions proposées

Les fonctionnalités nécessaires comprennent alors la création d'un projet MicMac, à partir du nœud proposé par Meshroom dans lequel il y a les données du projet (dont les images). L'utilisateur doit ensuite utiliser des fonctions correspondant aux grandes étapes d'un traitement

photogrammétrique. Le logiciel doit d'abord rechercher des points homologues (c'est le rôle de la commande Tapioca de MicMac), puis calculer la position et l'orientation des appareils photos (Tapas), et densifier le nuage de point (C3DC). D'autres fonctions peuvent être ajoutées : c'est le cas de la saisie d'un masque avec SaisieMasqQT par exemple. Ensuite, il doit être possible de visualiser le nuage de points, ce que nous permet de faire le visionneur de Meshroom pour les fichiers .abc.

Ces grandes fonctionnalités sont cependant déjà présentes dans Meshroom X MicMac, le projet consiste donc principalement en leur stabilisation et l'amélioration de leur accessibilité pour les utilisateurs novices, ainsi que l'implémentation de nouvelles fonctions.



Figure 3 : diagramme d'activité

Étude technique : choix des logiciels et des langages - architecture

Le choix des logiciels et des langages est antérieur au lancement du projet. Le langage Python est utilisé pour ajouter des nœuds dans Meshroom. Les scripts Python décrivent la construction d'une ligne de commande, en y incluant des paramètres, et en prenant en entrée le résultat des nœuds précédents, et les paramètres qui ont été ajustés.

Réalisation et suivi du projet

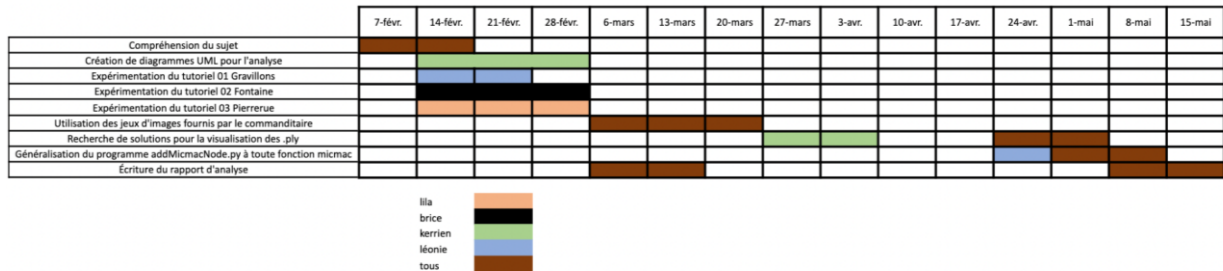
Risques

Les risques associés au déroulement de ce projet de développement informatique sont présentés dans le tableau suivant : ils ont souvent des conséquences peu importantes, sauf pour l'organisation du temps, ou une possible perte du travail effectué, mais ils peuvent être efficacement prévenus, en s'organisant et en veillant à sauvegarder correctement nos données.

Nature du risque	Gravité	Probabilité	Conséquence	Solution préventive	Solution corrective
Problème de serveur ENSG	Faible	Moyenne	Difficulté à accéder à la documentation Micmac	Télécharger les jeux de données dont on a besoin	Si possible, utiliser une autre connection
Panne d'une machine sur laquelle on travaille	Faible	Faible	Perte de données non sauvegardées, besoin d'une autre machine	Sauvegarder régulièrement notre travail	Utiliser une autre machine de l'école
Passer trop de temps à tester les commandes Micmac dans Meshroom	Moyenne	Fort	Moins de temps pour d'autres parties du projet	Vérifier régulièrement l'avancement, se fixer une limite	
Impossibilité de convertir certains types de fichier (ex: .ply et .abc)	Moyenne	Moyenne	Mauvais affichage des nuages de point / non compatibilité des noeuds MicMac et Meshroom		Utiliser un autre logiciel pour représenter la donnée (ex: CloudCompare) / Implémentation des solution possible prévues lors du développement initial

Perte de données lors d'un push sur github	Moyenne	Forte	Perte de données	Travailler sur un fork plutôt que sur le repository d'origine	
--	---------	-------	------------------	---	--

Planning prévisionnel



Conclusion

Le projet vise à stabiliser les fonctionnalités existantes, tester des nouveaux jeux de données, développer de nouveaux nœuds et proposer de nouvelles pipelines, notamment à travers l'écriture de tutoriels.

L'architecture mise en place par le Consortium 3D SHS fournit une base solide sur laquelle s'appuyer pour stabiliser les pipelines. En effet, une vingtaine de commandes MicMac ont été implémentées sous forme de nœuds dans Meshroom, ce qui offre déjà de nombreuses possibilités.

Il est nécessaire de prendre en compte les contraintes (notamment temporelle) et les risques potentiels pour mener à bien ce projet.

Ce projet représente alors une opportunité de permettre l'accès à une solution puissante et gratuite pour des applications de photogrammétrie majoritairement dans les sciences humaines et sociales.

Annexe 1 : nœuds MicMac implémentés dans Meshroom au début du projet

Commandes MicMac : AperiCloud, C3DC, Campari, GCPBascule, Martini, Pims2Mnt, RepLocBascule, SaisieMasqQT, Schnaps, SetExif, Tapas, TapiocaAll, TapiocaFile, TapiocaGraph, TapiocaLine, TapiocaMulScale, Tawny, Tequila, TiPunch

MicMacProject permet de débiter un projet MicMac en copiant les images dans un dossier.