

## Projet de Programmation :

Développement d'une interface web pour l'intégration  
d'un algorithme de ré-assemblage de fragments  
d'Ostraca par réseaux neuronaux siamois

## Cahier des charges

---

Jlajla Hamza, Pétrus Louis, Jelin Rémy, Luxey Victor, Pratx Julie, Lesourd-Aubert Valentine

2 octobre 2020

# Table des matières

<b>1</b>	<b>Présentation du projet</b>	<b>2</b>
1.1	État de l'art . . . . .	2
1.1.1	Archéologie et informatique . . . . .	2
1.1.2	Puzzles et reconstruction d'artefacts . . . . .	3
1.1.3	Deep-Learning et réseaux siamois de neurones . . . . .	3
1.1.4	Visualisation des graphes . . . . .	3
<b>2</b>	<b>Analyse des besoins</b>	<b>4</b>
2.1	Besoins fonctionnels . . . . .	4
2.2	Besoins non-fonctionnels . . . . .	4
2.3	Langages, formats et bibliothèques . . . . .	5
2.3.1	Langages . . . . .	5
2.3.2	Format . . . . .	6
2.3.3	Bibliothèques . . . . .	6
<b>3</b>	<b>Engagements de réalisation</b>	<b>7</b>
3.1	Livrables attendus . . . . .	7
3.2	flux opérationnel . . . . .	7
3.3	Planning de réalisation . . . . .	8
3.4	Visuels de l'interface . . . . .	8
3.5	Ajouts optionnels . . . . .	10
3.6	Conclusion . . . . .	10
<b>4</b>	<b>Références</b>	<b>11</b>

# Chapitre 1

## Présentation du projet

Les réseaux de neurones artificiels sont de plus en plus utilisés dans de nombreux domaines scientifiques, permettant de remplacer certains travaux encore laborieux par des modèles algorithmiques. L'archéologie est un de ces domaines qui peut bénéficier de cette nouvelle approche.

Le but de ce projet de programmation sera de réaliser une interface web à partir de laquelle les archéologues pourront travailler sur leurs données d'ostraca. Cette interface garantira une utilisation optimale de l'algorithme ainsi qu'une lecture claire des résultats obtenus.

Les deux fonctionnalités essentielles seront l'accès aux métadonnées pour donner la possibilité à l'utilisateur de sélectionner les fragments en fonction de divers paramètres (langues, site de fouille, date, etc). Il pourra ainsi créer un jeu de données spécifiques correspondant à ses besoins. Puis de visualiser et interagir avec les propositions de reconstruction des ostracas.

### 1.1 État de l'art

#### 1.1.1 Archéologie et informatique

Les débuts de l'utilisation des outils informatiques dans la recherche archéologique, dans les années 80, étaient focalisés sur des problématiques d'inventaire et les bases de données jouaient alors un rôle prépondérant. Encore aujourd'hui, les bases de données constituent un moyen extrêmement fiable de stocker durablement les données d'excavation et permettent, en outre, de contrôler l'accès aux ressources pour les travaux de recherches.

En parallèle du stockage, il fallait développer des moyens facilitant le transfert des données, d'une base de données à l'autre. Cela a conduit les archéologues à se regrouper au sein de congrès internationaux afin de s'accorder sur les standards expérimentaux. On peut citer l'initiative TEI [1] qui, en 2002, a fait adopter le format de fichier XML comme outil de référence dans le partage standardisé des données.

L'archéologie reste cependant une science empirique. Il en découle une diversité des méthodes, qui varie selon les spécialités et les époques étudiées. Les chercheurs doivent exploiter des ressources très diverses (codex, objets de cultes, traces d'activité humaine etc...).

Le progrès de l'informatique et la multiplication de projets, visant à répondre à des problématiques variées dans ce domaine, a vu l'émergence de projets comme le **OCHRE** [2], une plateforme d'outils computationnels capable de travailler avec des données récemment acquises jusqu'à l'archivage de celles-ci. C'est un programme divisé en plusieurs parties dont le but est de faciliter le travail des chercheurs.

L'une d'elles, visant à l'acquisition des données aussi bien issues des bases de données de l'OCHRE ou de bases de données externes, est une application client JAVA qui permet l'acquisition et l'analyse des données dans des workflow analytiques. Une autre plate-forme permet la mise en ligne des résultats ou des travaux publiés. La dernière plate-forme est la mise en relation de serveurs qui utilisent les outils analytiques et les bases de données avec le client JAVA.

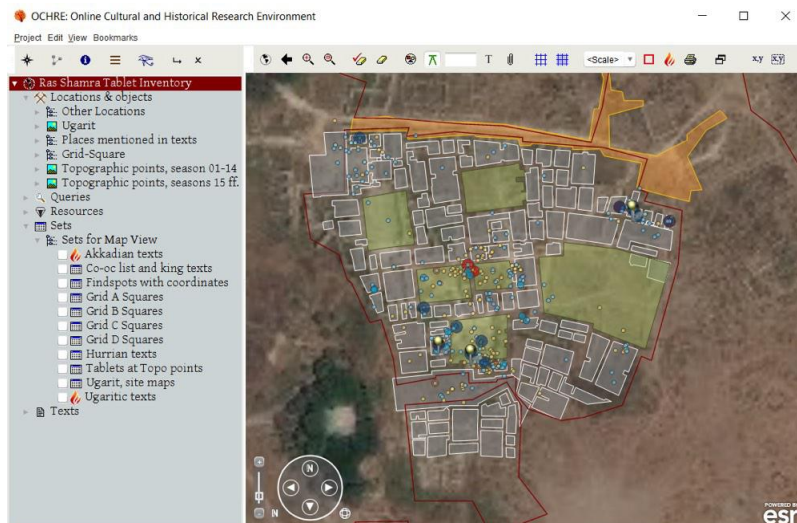


FIGURE 1.1 – Interface OCHRE

### 1.1.2 Puzzles et reconstruction d'artefacts

Les premiers puzzles apparaissent il y a environ 250 ans. Les mathématiques et la logique derrière la résolution de ces puzzles peuvent être appliqués dans de nombreux domaines tels que : la reconstruction d'artefact, l'édition d'image, l'assemblage de fragments d'ADN, etc. La classification et la reconstruction de poteries antiques est un processus long et coûteux pour les archéologues. Ils sont amenés à travailler avec de très nombreux fragments, qui doivent être caractérisés afin de pouvoir reconstruire la forme originale et de déchiffrer le contenu inscrit dessus.

Les programmes informatiques de reconstruction 3D, utilisés par les archéologues, ont comme point commun d'avoir besoin des caractéristiques physiques des fragments (dimensions, état des bords, etc) et leurs propriétés (matière, couleur, etc) (Sablatnig et al., 1997, Melero et al., 2003) [3][4]. Ainsi les modèles mathématiques de ces programmes de reconstruction empruntent une logique commune avec celle de la résolution de puzzles.

De nos jours, avec le développement de l'intelligence artificiel, il est possible de reconstruire des images, à partir de fragments, grâce à des algorithmes de reconstruction à apprentissage automatisé qui utilisent la logique et les mathématiques cachés derrière les puzzles, le machine learning.

### 1.1.3 Deep-Learning et réseaux siamois de neurones

Le Machine Learning est une méthode d'apprentissage automatique. Elle se base sur des approches statistiques, pour donner à un programme la capacité d'"apprendre" à l'aide de données d'entraînement, afin d'améliorer sa précision dans une tâche donnée.

Les réseaux de neurones siamois (SNN) sont un type de réseaux de neurones artificiels utilisés pour résoudre des problèmes de reconnaissances d'images (Bromley et al., 1993)[5]. Ils sont entraînés à apprendre de nouvelles classes d'objets avec un minimum de données.

Il existe de nombreuses bibliothèques open source permettant de développer et d'exécuter des applications de Machine Learning comme des réseaux neuronaux : *OpenNeuralNetwork* pour le langage C++, en langage python *TensorFlow* et *Scikit-Neural Network* ou encore *Brain.js* en langage Javascript.

### 1.1.4 Visualisation des graphes

L'algorithme python, proposé par C.Ostertag, permet la reconstruction d'ostraca, des fragments de poteries datant de l'antiquité qui sont retrouvés en large quantité lors de fouilles archéologiques. Il est capable, à partir des images liées aux données d'ostraca fournies, de proposer une reconstruction en 2D des fragments (C.Ostertag et al., 2020) [6] grâce à un SNN. Les résultats sont sous forme de graphes au format NetworkX et peuvent être exportés pour être utilisés dans un environnement logiciel autre (format image ou vectoriel).

# Chapitre 2

## Analyse des besoins

### 2.1 Besoins fonctionnels

Les fonctionnalités essentielles attendues sont :

- Ajout, suppression et édition des métadonnées,
- Choix manuel des filtres à appliquer sur le tableau de métadonnées parmi les paramètres disponibles dans le-dit tableau,
- Passage possible entre la vue du tableau de métadonnées et la vue des images de fragments correspondants pour faciliter sa sélection,
- Lancement de l'algorithme de création d'un jeu de données artificiel à partir de n'importe quelle image ou ligne du tableau,
- Lancement de l'algorithme de ré-assemblage si le panier de fragments contient plus d'un élément,
- Visualisation des graphiques/résultats et interaction avec eux.

### 2.2 Besoins non-fonctionnels

Une attention particulière sera portée à la mise en forme des métadonnées qui seront intégrées dans le fichier **XML**[7]. Celles-ci ne pourront être intégrées que si elles respectent des contraintes pré-établies dans un fichier XML Schema Definition (XSD). XSD est, à l'instar du CSS pour HTML, un fichier externe utilisé pour définir des standards XML, contrôlant le typage des données.

Les images d'ostraca en lien avec les métadonnées devront respecter le format **NPZ**.

En ce qui concerne la présentation de l'interface web, nous utiliserons les langages **HTML** et **CSS**. Pour un rendu interactif et dynamique, nous ajouterons des fonctionnalités (boutons, menus déroulants, etc.) grâce au langage **Javascript**.

Une fois intégrées dans un fichier XML, les méta-données pourront être modifiées et manipulées grâce à des fonctions implémentées elles aussi en langage JavaScript.

La page web devra, dans la mesure du possible, être compatible avec différents navigateurs, les plus connus étant Google Chrome, Mozilla et Safari.

## 2.3 Langages, formats et bibliothèques

### 2.3.1 Langages

#### HTML :

- HTML signifie Hyper Text Markup Language.
- Permet la création de pages Web en respect des standards développés par le W3C.
- HTML se compose d'éléments ou "tags" qui stockent un contenu divers.
- A un tag, peuvent être spécifiés des attributs qui permettent de contrôler le rendu de son contenu.
- Le navigateur reconnaît les tags et affiche uniquement leur contenu, avec le rendu adéquate.
- Au moment du chargement de la page Web, le navigateur génère le DOM (Document Object Model) HTML.
- Le DOM est un standard du W3C et une interface de programmation permettant d'interagir avec les tags.

#### CSS :

- CSS pour Cascading Style Sheet ou feuilles de style en cascade, est un langage de feuille de style.
- Il est utilisé pour décrire la présentation d'un document écrit en HTML ou en XML.
- Il décrit la façon dont les éléments doivent être affichés à l'écran (polices, couleurs, espacements, etc.).
- C'est l'un des langages principaux du Web ouvert.

#### JavaScript :

- Javascript, nommé alternativement ECMAScript, est le langage de programmation du Web.
- Langage de script, léger et orienté-objet, utilisant le concept d'héritage de classes ou "prototypage".
- Sa version actuelle (ES6) est supportée par la plupart des navigateurs modernes.
- Tous les navigateurs intègrent nativement un interpréteur javascript.
- Permet la création de pages Web dynamiques, en suivant le paradigme de programmation événementielle.
- Possède des outils lui permettant d'interagir avec des balises HTML/XML par l'intermédiaire du DOM.

#### XML :

- XML signifie eXtensible Markup Language.
- Conçu pour le stockage et le transport de données non-sécurisées.
- Reconnu officiellement par le W3C et intégré au projet HTML5.
- Comme HTML il appartient à la famille des langages à balises et possède son propre DOM.
- A l'instar de CSS, il permet une séparation de l'information des données et de leur affichage.
- Léger, portable et utilisable aussi bien côté client que côté serveur.
- Syntaxe aisément compréhensible tant par l'homme que par la machine.
- Bonne alternative aux systèmes de gestion de base de données tant que les données ne sont pas sensibles.
- Peut être associé à des schémas XSD pour définir les contraintes imposées aux données.

#### XQuery :

- A l'instar de SQL sur les modèles relationnels, permet d'effectuer des requêtes sur les fichiers XML et d'autres standards W3C [8].
- Syntaxe simple nommée FLOWER en référence à ses 5 éléments syntaxiques (for, let, where, order by et return).
- Permet d'extraire des données au sein d'un fichier XML mais aussi des données relatives à une collection de fichiers XML.
- Permet également, à partir des résultats d'une requête, de produire un nouveau document XML tout en conservant l'original (programmation fonctionnelle)

### 2.3.2 Format

#### **NPZ :**

NPZ est un format de fichier traité avec la bibliothèque NumPy [9] qui permet le stockage des données d'une liste à l'aide de la compression gzip. Lors de l'ouverture du fichier .npz, un objet NpzFile est renvoyé, de type semblable aux dictionnaires.

### 2.3.3 Bibliothèques

#### **SheetJS js-xlsx :**

- Implémente la lecture, l'écriture et l'export de données tabulaires.
- Supporte de nombreux formats de fichier, dont .xls, .ods et .csv.
- Compatible avec les navigateurs supportant ES3+ (ie : tous les navigateurs actuels).
- Existe en version professionnelle ou communautaire (libre)

#### **JSNetworkX :**

- Implémentation Javascript de la librairie Networkx écrite en Python.
- Utilisé dans la création, la manipulation et l'analyse de graphes.
- Permet le chargement et le stockage de graphes sous de nombreux formats.
- Offre une visualisation facilitée des graphes directement sur une page Web.

# Chapitre 3

## Engagements de réalisation

### 3.1 Livrables attendus

Une interface Web ergonomique et fonctionnelle permettant à l'utilisateur de :

- Visualiser les métadonnées des Ostraca ainsi que les images des fragments,
- Editer, trier ces données,
- Construire une sélection de fragments dans le but de lancer l'algorithme de ré-assemblage,
- Lancer l'algorithme de création d'un jeu de données artificiel à partir de n'importe quelle image de fragment,
- Lancer l'algorithme de ré-assemblage,
- Visualiser et interagir avec les graphes obtenus.

### 3.2 flux opérationnel

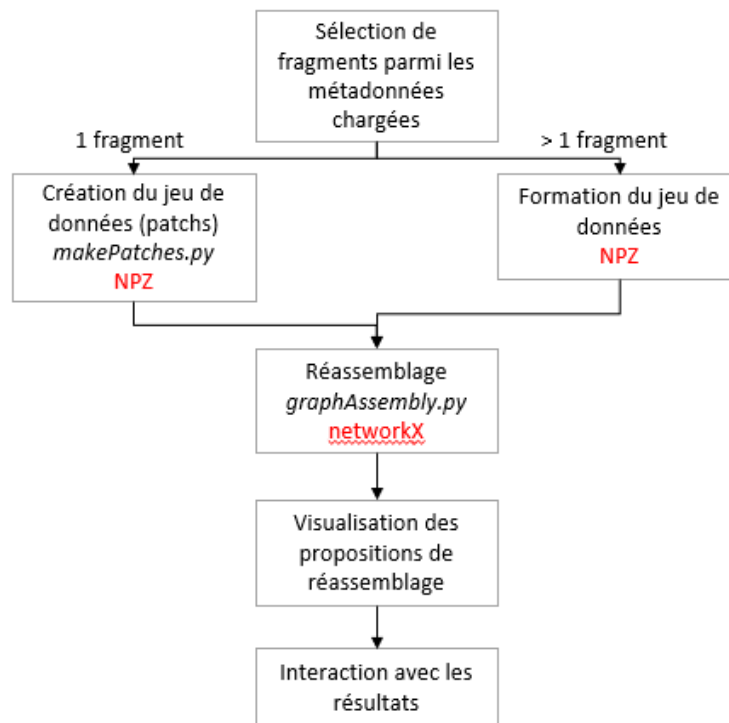


FIGURE 3.1 – Flux opérationnel imaginé pour l'interface du projet Ostraca



### 3.3 Planning de réalisation

Ci-dessous l'organisation du planning pour les différentes tâches du projets. L'effectif de travail ainsi que la durée indiquée pour chaque tâche sont informatifs et pourront être sujets à des adaptations en cours de projet pour assurer le bon déroulement de celui-ci.

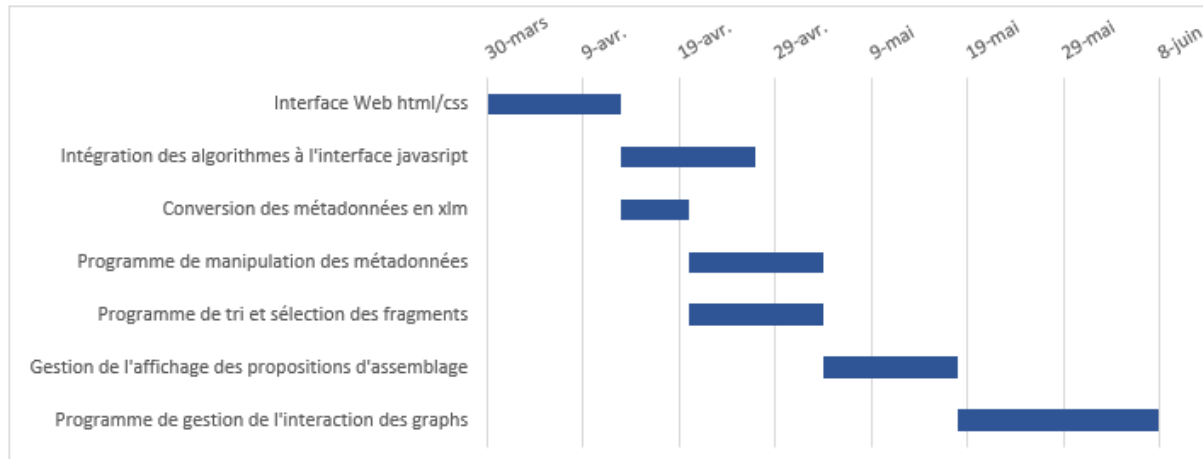


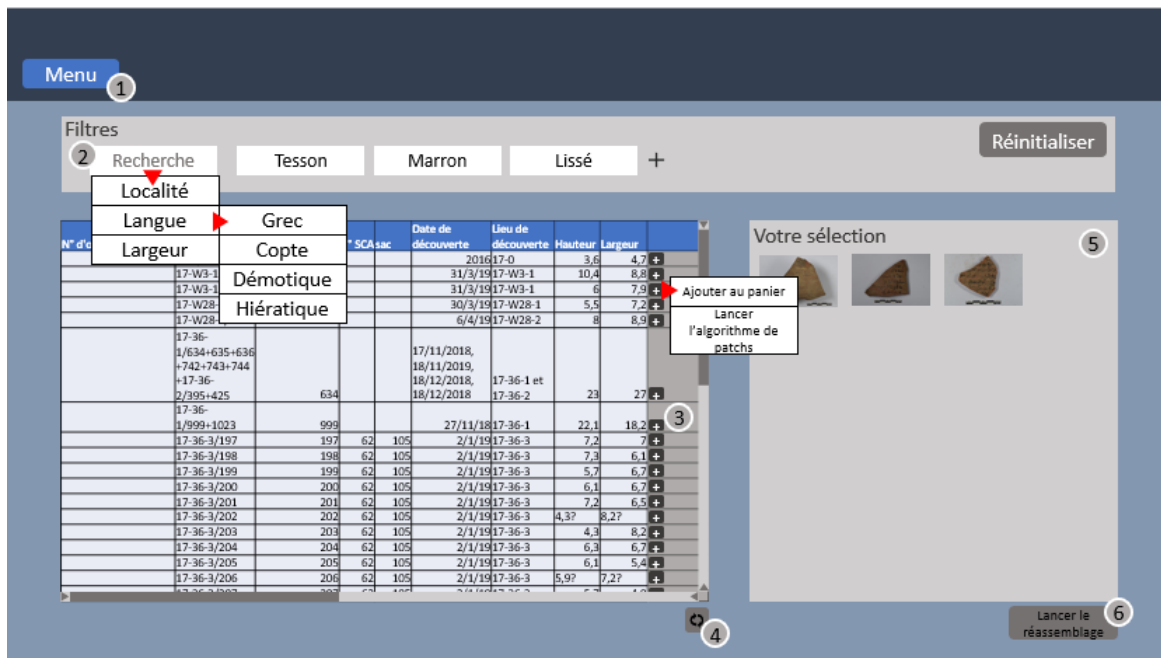
FIGURE 3.2 – Diagramme de Gantt du projet

Tâches	Date début	Date fin	duree (j)	Personnes impliquées
Interface Web html/css	30/03/2020	13/04/2020	14	3
Intégration des algorithmes à l'interface javascript	13/04/2020	27/04/2020	14	2
Conversion des métadonnées en xlm	13/04/2020	20/04/2020	7	3
Programme de manipulation des métadonnées	20/04/2020	04/05/2020	14	2
Programme de tri et sélection des fragments	20/04/2020	04/05/2020	14	2
Gestion de l'affichage des propositions d'assemblage	04/05/2020	18/05/2020	14	3
Programme de gestion de l'interaction des graphs	18/05/2020	08/06/2020	21	4

FIGURE 3.3 – Echancier du projet

### 3.4 Visuels de l'interface

Les maquettes présentées ci-dessous permettent de visualiser l'organisation globale de certaines pages web que comprendra l'interface. Il s'agit seulement d'idées préalables qui pourront évoluer au cours du projet.



1. Bouton de retour à la première page
2. Champs de recherche des paramètres disponibles, Les paramètres sélectionnés sont affichés à droite
3. Passer la souris sur une ligne affiche dans un pop-up le fragment correspondant
4. Bouton permettant de visualiser les fragments filtrés sous forme d'images ou de tableau de données (voir Figure 4.5)
5. Champ permettant de visualiser une miniature des fragments sélectionnés
6. Lance l'algorithme de ré-assemblage avec les fragments sélectionnés

FIGURE 3.4 – Visuel : sélection des paramètres, métadonnées et fragments sélectionnés

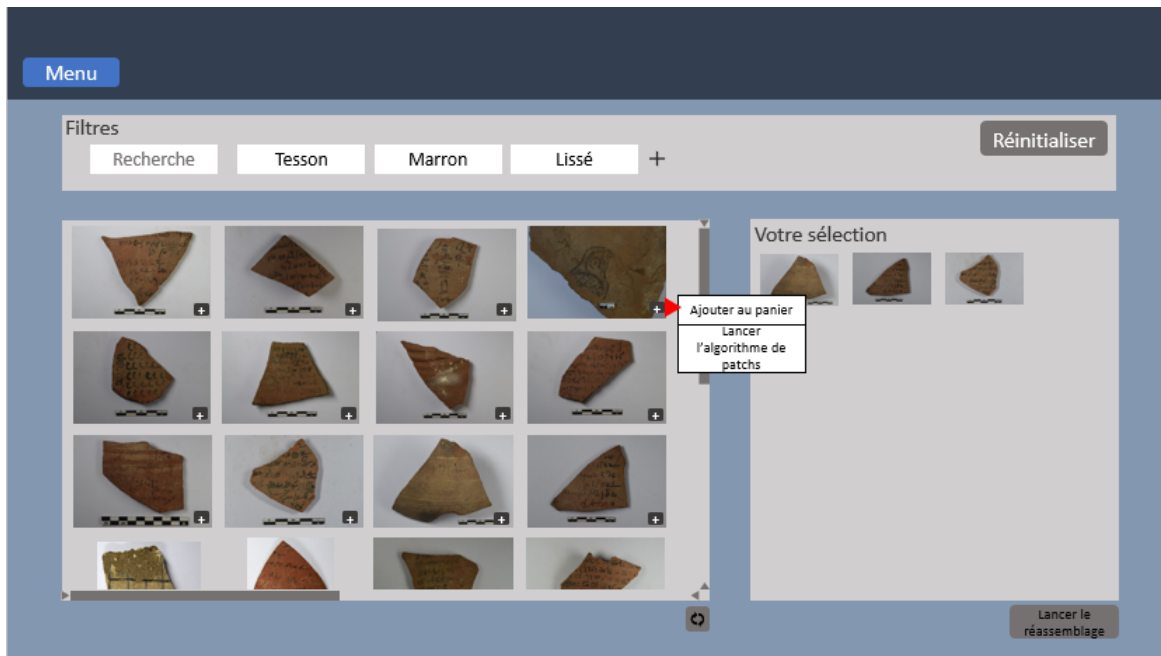
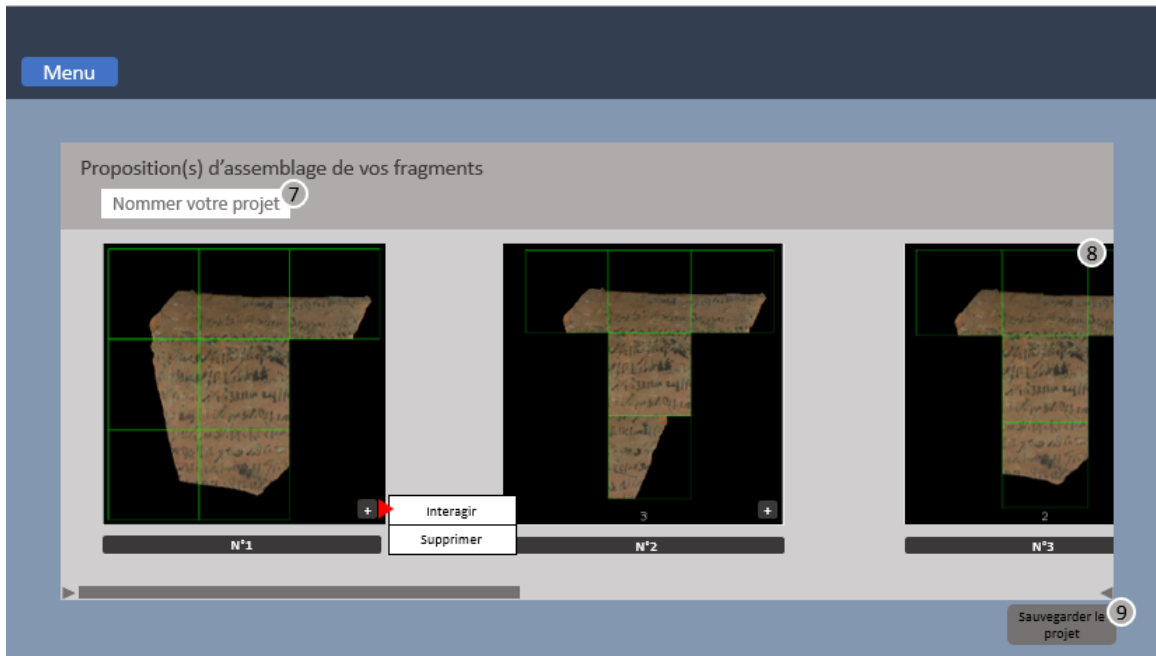


FIGURE 3.5 – Page de sélection des paramètres, images d'ostraca et fragments sélectionnés



- 7. Champ permettant de nommer le projet en cours (optionnel)
- 8. Espace permettant la visualisations des images d'assemblages proposés par l'algorithme
- 9. Bouton de sauvegarde du projet en cours (optionnel)

FIGURE 3.6 – Page des résultats, propositions de reconstruction

### 3.5 Ajouts optionnels

Implémentation d'un serveur :

En ce qui concerne l'introduction de l'algorithme de ré-assemblage à l'interface javascript, deux possibilités peuvent être envisagées : la création d'un serveur qui permettra d'exécuter le script en ligne via une interface CGI (pour Common Gateway Interface) [10] ou simplement par la création d'un serveur à l'aide d'un script en Python.

Sauvegarde des résultats :

Permettre à l'utilisateur de sauvegarder ses données sélectionnées ainsi que les graphes obtenus, en ajoutant par ailleurs un titre au projet.

### 3.6 Conclusion

L'analyse des besoins nous permet d'identifier les points qui seront les plus importants pour permettre à l'utilisateur une utilisation efficace de l'algorithme de ré-assemblage. L'ergonomie et le visuel de l'interface doivent permettre à des non-initiés au développement informatique d'utiliser l'algorithme sans difficultés. Un autre point important est la fonction de tri des métadonnées car elle est un besoin essentiel des utilisateurs potentiels de cette interface.

Au delà de l'aspect de chargement des métadonnées, la visualisation et l'interactivité des résultats de ré-assemblage sont au autre point clef de ce projet.

Finalement le développement de cette interface devra être suffisamment organisée pour que chaque étudiant puisse prendre part aux différentes tâches. Cela implique notamment de produire des codes propres et suffisamment commentés, selon les bonnes pratiques de programmation.

# Chapitre 4

## Références

- [1] Text Encoding Initiative, <https://tei-c.org/>
- [2] <https://dhant.sciencesconf.org/resource/page?id=11forward-action=pageforward-controller=resourcelang=en>
- [3] Sablatnig, Robert Menard, Christian. (1997). 3D Reconstruction of Archaeological Pottery using Profile Primitives. Proc. of International Workshop on Synthetic-Natural Hybrid Coding and Three-Dimensional Imaging. 93-96.
- [4] Melero, Francisco Torres, Juan León Salas, Alejandro. (2003). On the Interactive 3D Reconstruction of Iberian Vessels.. 71-78. 10.2312/VAST/VAST03/071-078.
- [5] Bromley, J., Bentz, J. W., Bottou, L., Guyon, I., Lecun, Y., Moore, C., Sackinger, E., Shah, R. (1993). Signature verification using a Siamese time delay neural network. International Journal of Pattern Recognition and Artificial Intelligence, 7(4).
- [6] Cecilia Ostertag, Marie-Beurton Aimar, 2020. Matching Ostraca fragments using a siamese neural network, Pattern Recognition Letters, doi : <https://doi.org/10.1016/j.patrec.2020.01.012>
- [7] XML Schema (W3c), February 2017. Page Version ID : 763143988.
- [8] World Wide Web Consortium Standards, <https://www.w3.org/standards/>
- [9] Oliphant, T. E. (2006). A guide to NumPy (Vol. 1, p. 85). USA : Trelgol Publishing.
- [10] Blackwell Jr, W. C., Alston, R., Farrington, K. M. (1999). U.S. Patent No. 5,857,191. Washington, DC : U.S. Patent and Trademark Office.