

Tache 1 :

Sites	Carole Granger	Matt Farley	Pierre Burel	Cafe Rati
Support	Site WEB	Site WEB	Site WEB	Site WEB
Champ sémantique	Vocabulaire technique pour graphisme et jeu vidéo.	Sémantique de développeur Web	Sémantique du code	Sémantique développeur WEB
Orga de l'info	Bloc infos avec une image et un moyen de cliquer pour en apprendre plus sur l'info si besoin.	Liste d'infos sans images ou avec un lien.	Orga de haut en bas dans un code	Listes d'Images avec des boutons pour choisir certains types de projets.
Constru des pages	Plusieurs Catégories/Rubriques par pages agencé en colonne qui sont renseigné dans un sommaire en haut de page	1 seule pages construite en Plusieurs Catégories / Rubriques en plusieurs blocs en ligne.	Une seule page avec sa description et son logo	Titre -> catégories de tri -> projet. Projet -> changement de pages avec flèches sur cotés et lecture verticale de chaque points du projet.
Barre de nav	Fixe en haut de page, basique (noms des catégories cliquables)	Aucune	Pas de barre de navigation	Bouton burger en sticky
Header / Footer	Barre nav + Nom personne en grand (header prend 100% écran de base) / Contacts + Copyright	Contacts dans le footer + insitations a travailler avec lui.	Pas de header / footer	Header n'a qu'un titre.
Choix couleurs	Blanc et pétant	Couleurs rappelant visual studio	Couleurs rappelant visual studio + logique du code	Fond noir.
Eléments de design	Blocs carré	Beaucoup de logos	Représentation en code	Banderole GitHub en sticky en haut a droite.
Accessibilité	Compréhensible, navigation via tabulation pour liens important, description de chaque image lorsque la souris est dessus. Contraste couleurs.	Contraste couleurs. Pas de ALT	Navigation entre les différents sites avec tabulation, beaucoup de texte.	Contrast entre images et fond. Trop peu de texte. Pas de ALT Navigation entre les différents sites avec tabulation mais aucun élément

	Pas de ALT		Contraste couleurs. Pas de ALT	visuel pour savoir lequel est sélectionner.
Incitations clic/Scroll	Petit boutons de couleurs différentes des blocs / Répartition en colonne des infos.	Boutons en violet sur blanc sobres.	Pas de scroll et element cliquable en bleu	Réactions au passage de la souris sur chaque éléments cliquables.
Parcours utilisateur	Contact en fin de pages, liens projet -> vidéo youtube -> fin du chemin (-),	Ouvre un nouvel onglet lors d'un clic sur le lien d'un projet.	Aucun parcours, pas de description de projet, juste contacts et historique de travaux.	Bouton burger sur toutes les pages + flèches pour passer de pages en pages.
Version PC/Mobile	Versions mobile légèrement réarranger	Versions mobile légèrement réarranger	Versions mobile légèrement réarranger	Versions mobile légèrement réarranger

Conclusion :

Mon portfolio sera Sobre avec 2 à 3 couleurs différentes. Il n'y aura pas trop de textes, mais suffisamment pour tout comprendre (mots clés dans certains cas si un texte est inutile comme pour des images). Utilisation de la sémantique du sujet des projets. Implémentation d'un modèle portable et de l'accessibilité avec les ALT et les notes pour décrire chaque contenu. Utilisation de facilitations graphiques pour inciter à cliquer / Scroll comme des petites flèches ou des couleurs attirantes pour les boutons

Tache 2 :

Refonte du site internet pour un conservatoire – Recueil et analyse des besoins du client

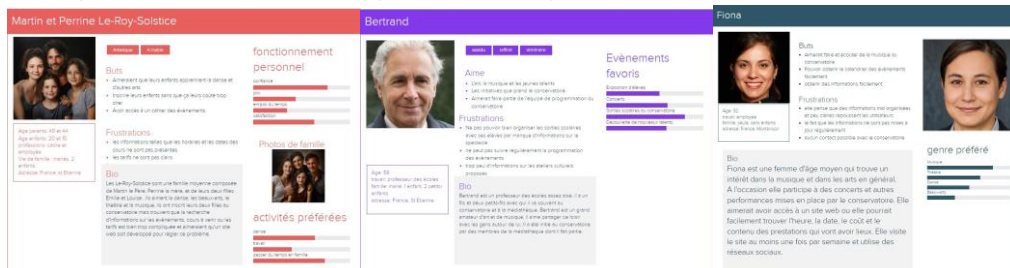
Missions :

- Présenter de manière claire, concise et vulgariser le contexte, les attentes et les exigences du client.
- Extraire les attentes des utilisateurs finaux par le biais de persona.
- Faire une veille concurrentielle de certains sites de conservatoires pour orienter nos choix
- Faire un tri de carte des contenus et fonctionnalités retenus suite à la veille.

Etapes :

- 1^{er} étape, compréhension et résumé de la commande
 - Phase 1: Étude préliminaire
Objectifs : Définir les orientations générales du projet, notamment la charte graphique, l'architecture technique, et les aspects de sécurité.
Livrables: Identité visuelle du site, cahier des spécifications techniques et fonctionnelles, arborescence détaillée du site, planning des phases à venir.
 - Phase 2: Développement, tests et validation
Objectifs: Développer le site en fonction des spécifications définies, tester son fonctionnement et assurer son adaptation à différents supports.
Livrables: Gabarits des différentes pages du site.
 - Phase 3: Elaboration et intégration des contenus
Objectifs: Préparer et intégrer le contenu éditorial, former les administrateurs du site.
Livrables: Manuel de procédures, page de la version 1 du site.
 - Phase 4: Test des produits livrables
Objectifs: Tester les liens, modules, pages et fonctionnalités du site avant sa mise en ligne.
Livrables: Produits testés et validés.
 - Phase 5: Formation et transfert de compétences
Objectifs: Former les administrateurs du site pour une gestion autonome.
Livrables: Formation pour 3 agents chargés de l'administration du site.
 - Phase 6: Hébergement du site
Objectifs: Mettre en place l'hébergement du site en interne, en assurant la sécurité et la confidentialité des données.
Livrables: Description des solutions d'hébergement et de sécurité.
 - Phase 7: Promotion et référencement
Objectifs: Optimiser le référencement du site et assurer sa visibilité sur les réseaux sociaux.
Livrables: Stratégie de promotion et référencement.
 - Phase 8: Maintenance et garantie
Objectifs: Assurer la maintenance du site et garantir sa conformité pendant une période définie.
Livrables: Forfait de maintenance pour 12 mois.
 - Phase 9: Planning prévisionnel
Objectifs: Assurer que le site soit opérationnel avant une date précise.
Livrables: Planning détaillé pour l'ensemble des phases.

- 2^e étape, création de l'architecture du site. Penser à la possibilité d'ajouts de la part du client, mais aussi des exigences ergonomiques nécessaires pour les utilisateurs et les admins.
- 3^e étape , Etude des clients type à l'aide de personas.



Puis analyse des envies et besoins des clients pour les fonctionnalités du site.

4^e étape, Veille concurrentiel. Nous avons étudié 2 autres sites de conservatoire pour pouvoir analyser les choses à faire et à ne pas faire. Ensuite, nous avons mis au clair les fonctionnalités retenues pour notre site.

5^e étape, conclusion du projet.

Nous avons créé un tri de carte et l'arborescence du site final à l'aide de toutes les recherches précédemment faites.

Compétences techniques :

- Appréhender les besoins du client et de l'utilisateur
- Mettre en place les outils de gestion de projet
- Identifier les acteurs et les différentes phases d'un cycle de développement
- Formaliser les besoins du client et de l'utilisateur
- Identifier les critères de faisabilité d'un projet informatique

Compétences transversales :

- Respecter une durée donnée pour un projet
- Savoir travailler en équipe
- Savoir être critique

Conclusion :

Ce projet nous a permis de découvrir le travail en équipe étant donné que celui-ci était le premier dans notre formation. Il nous a permis d'apprendre à discuter, être critique avec les idées et travail de tout le monde, à travailler beaucoup dans un court laps de temps. Il nous a appris aussi que bien mettre au clair le sujet était la chose la plus importante d'un projet, ici cela n'a pas été entièrement fait. On a aussi appris qu'il fallait travailler plus en groupe sur chaque petite partie, car si une personne se retrouve dans l'incapacité d'être là le jour de la présentation, il faut quand même être capable de présenter son travail.

Développement d'un convertisseur d'image en ASCII-ART - Développement d'applications en réponse à un besoin client

Missions :

- Développer un convertisseur d'image en ASCII-ART
- Interagir avec des fichiers externes au programme
- Créer des interactions utilisateurs
- Développer un Réducteur d'images.
- Compréhension du fonctionnement d'un logiciel tier a l'aide d'une documentation

Etapes :

- Nous avons commencé par appliquer et comprendre comment fonctionne l'outil img2pgm avec différentes images et comparer ceux qui en ressortaient pour analyser comment étaient conçus les fichiers PGM.
- Puis nous avons décodé l'en-tête du fichier pour déterminer le type de fichier PGM (va de P1 à P6, P2 et P5 sont en binaire, et celui que nous manipulons est P5) ainsi que la taille (longueur , largeur et la taille maximale d'une valeur).
- Ensuite, nous avons utilisé la palette de caractères donnée, composée de 8 caractères différents, pour représenter l'image dans la console. Nous avons également intégré la possibilité de choisir le fichier à retranscrire, tant qu'il était dans le même dossier que le programme. Le résultat était bon, mais la limite de caractère dans la console aurait pu devenir très vite un inconvénient. Pour compenser cela, nous sommes passés à la version 2.
- Maintenant, le programme enregistre l'image en ascii art dans un fichier texte nommé par l'utilisateur au moment de l'exécution pour pouvoir l'ouvrir dans d'autres lecteurs de textes, comme le bloc-notes ou Notepad++. Le fichier sera dans le dossier où sont déjà stockés les fichiers .cpp, .h , l'image et les palettes.

```
/* Version 2
std::string nom_fout;
std::cout << "entrez le nom du fichier de sortie :"<<std::endl;
std::cin >> nom_fout;*/
img2fich(fichier, std::string(output));
//Ecriture de l'image convertie dans un fichier image F
```

- Ici, le client peut utiliser la palette de son choix à condition de l'avoir déposée dans le dossier source auparavant. Ainsi, il peut modifier l'apparence à sa guise, par exemple en variant et ajoutant des tons de gris ou l'écart entre les caractères. Le choix se fait lors de l'exécution du programme, mais les changements dans cette version ne permettent rien de plus.

```
/* Version 3
std::string nom_fichier;
std::string nom_palette;
std::cout << "entrez le nom du fichier :"<<std::endl;
std::cin >> nom_fichier;
std::cout << "entrez le nom de la palette :"<<std::endl;
std::cin >> nom_palette;*/
//récupération du nom de la palette et de l'image voulues
```

- Maintenant, les noms des fichiers sont rentrés directement au lancement du programme. Il faut cependant que les fichiers utilisés soient dans le même fichier que

l'exécutable, que l'on appelle « dossier source ». Le --help permet d'avoir l'information sur les commandes disponibles.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\IUT\IA\SI\SAE 1.1\img2txt\x64\Debug> ./img2txt.exe --help
Options:
--input fichier           Spécifie le fichier image à convertir.
                          Si ce paramètre n'est pas spécifié, le fichier est demandé via la
                          console.
--output fichier          Spécifie le nom du fichier texte qui contiendra l'Ascii Art.
                          Si ce paramètre n'est pas spécifié, l'Ascii Art est sorti dans la
                          console.
--palette fichier         Spécifie un fichier texte contenant la palette de couleur Ascii.
                          Chaque ligne du fichier contient un caractère en UTF-8, du plus
                          sombre au plus clair.
                          Si ce paramètre n'est pas spécifié, la palette par défaut
                          est "W", "w", " ", "1", "0", ".".
--help                   Affiche cette aide.
--width                  Spécifie la largeur max de l'Ascii Art.
                          Si ce paramètre n'est pas spécifié, aucune limite n'est fixée.
                          Voir Remarques.
--height                 Spécifie la hauteur max de l'Ascii Art.
                          Si ce paramètre n'est pas spécifié, aucune limite n'est fixée.
                          Voir Remarques.

Remarques :
Quelles que soient les valeurs des options --width et --height, la taille de l'Ascii Art
est bornée par la taille de l'image en entrée. La taille de l'Ascii Art conserve
toujours le même ratio que l'image d'entrée, les valeurs des options --width et --height
ne sont que des maximums.
PS C:\IUT\IA\SI\SAE 1.1\img2txt\x64\Debug>
```

- Le redimensionnement se fait via l'invite de commande directement, avec les valeurs souhaitées par l'utilisateur. En sortie, on aura l'image ASCII au format .txt redimensionné selon la demande de l'utilisateur. L'image ne sera pas déformée, donc le ratio largeur/hauteur sera le même (ex 16/9)

Cependant, il y a des limitations à ce redimensionnement, par exemple lorsque le redimensionnement est trop petit. A ce moment-là, les carrés de division sont plus grands que l'image obtenue et le programme plantera car il essayera de lire à des endroits non accessibles.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\IUT\IA\SI\SAE 1.1\SAE1_1\img2txt\x64\Debug> ./img2txt.exe --input img --output re --palette palette --width 50 --height 100
PS C:\IUT\IA\SI\SAE 1.1\SAE1_1\img2txt\x64\Debug>
```

Compétences techniques :

- Savoir Interagir avec des fichiers externes au programme
- Savoir créer des interactions utilisateurs
- Savoir s'approprier une documentation technique pour comprendre le fonctionnement d'un logiciel ou d'une extension de fichier.

Compétences transversales :

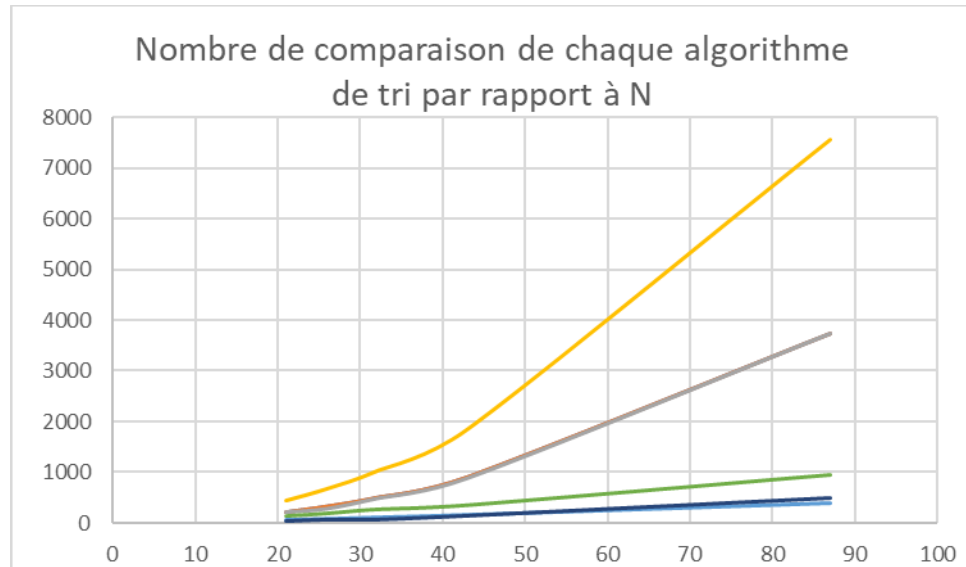
- Respecter une durée donnée pour un projet
- Savoir développer des applications
- Savoir déboguer un programme

Conclusion :

Ce projet a été le premier projet de C++ de notre formation et nous a permis de mettre à profit toutes les connaissances de base que nous avons vu à ce moment-là. Cela nous a permis de vraiment nous rendre compte de l'ampleur du travail lors d'un projet plus gros que de simple TP

Développement, analyse et comparaison d'algorithmes de tri:

- Missions :
 - o Développer une application contenant divers algorithmes de tri
 - o Analyser la complexité des tris pour ensuite les comparer
- Etapes :
 - o Développer tous les tris demandés puis essayer de trouver des tris intéressants sur internet pour les implémenter dans notre application.
 - o Analyser les algorithmes de tri pour deviner la complexité des tris puis à l'aide d'un tableur, visualiser la complexité des tris pour les assimiler à une fonction type ($f(x) = n$; $f(x) = n^2$; $f(x) = n \cdot \log(n)$, etc...)



- o Ecriture du rapport décrivant la démarche de notre projet et la conclusion de celui-ci.

Compétences techniques :

- Savoir Interagir avec des fichiers externes au programme
- Savoir créer des interactions utilisateurs
- Savoir s'approprier une documentation technique pour comprendre le fonctionnement d'un logiciel ou d'une extension de fichier.

Compétences transversales :

- Respecter une durée donnée pour un projet
- Savoir utiliser un tableur pour visualiser une liste de données.
- Savoir développer des applications
- Savoir déboguer un programme

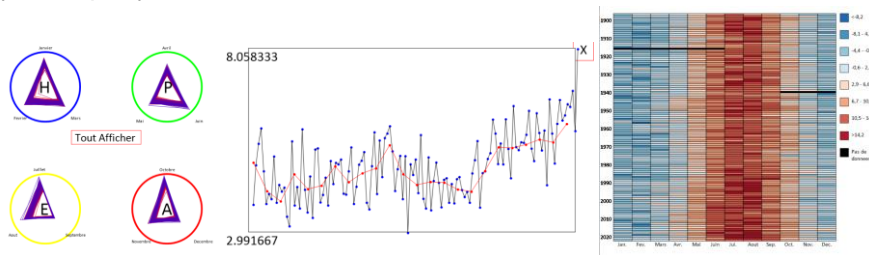
Conclusion :

Ce projet nous a appris l'intérêt de divers programmes de tri qui selon leur utilisation peuvent être plus ou moins intéressants. Certains pouvaient trier très rapidement dans certains cas optimaux mais seraient catastrophiques dans d'autres. Chaque algorithme présente un intérêt différent, que ce soit en terme de vitesse de tri, d'optimisation de la puissance de calcul, ou encore de complexité de tri pour le comprendre.

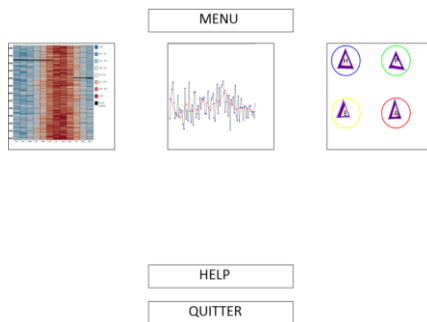
Création et visualisation de graphes - Développement d'applications en réponse à un besoin client

Missions :

- Extraire des données depuis un fichier externe
- Imaginer et réaliser un graphique grâce aux données
- Faire des interactions utilisateurs
- Etapes :
 - Création de graphes à l'aide des données
 - Appropriation d'une bibliothèque graphique pour dessiner nos graphes (LibGraph2)



- Création d'interactions utilisateur avec le menu ou encore des boutons dans les graphes



Compétences techniques :

- Appropriation d'une bibliothèque graphique (LibGraph2)
- Organisation de données depuis un fichier externe
- Création d'interaction utilisateur

Compétences transversales :

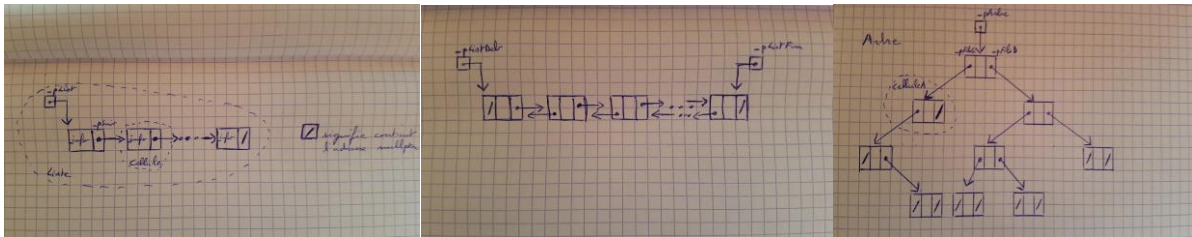
- Respecter une durée donnée pour un projet
- Savoir travailler en équipe
- Savoir développer des applications
- Savoir déboguer un programme

Conclusion :

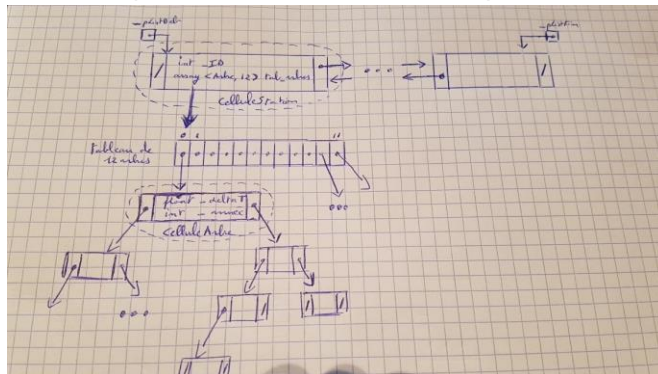
Ce projet nous a poussé à imaginer de nous même un graphique qui puisse être utile à des scientifiques tout en le rendant le plus possible lisible et clair. Notre graphique personnalisé n'a pas très bien répondu à la consigne car même si il était utile et pas proposé par le professeur, il était très peu clair et la lisibilité était impossible sans un descriptif approfondi.

Récupération de données et création d'architectures complexes :

- Missions :
 - Créer un ensemble d'architectures différentes
 - Récupération de données
 - Archivages des données dans les architectures
- Etapes :
 - Tout d'abord, nous avons créé pas à pas les différentes architectures demandées. Nous avons fait un arbre avec toutes les fonctions utiles pour le modifier. Nous avons fait 2 listes différentes, une liste chaînée simple et une liste chaînée bidirectionnelle. Nous avons terminé avec un arbre binaire.



- Nous avons ensuite récupéré les données grâce à un algorithme donné par le professeur. Celui-ci étant complexe, nous avons dans un premier temps analysé pas à pas son programme pour comprendre comment il rangeait les données. Une fois ceci fait, nous avons pu créer des fonctions permettant d'afficher les données que l'on voulait dans l'ordre que l'on voulait.



Compétences techniques :

- Organisation de données depuis un fichier externe
- Analyse d'un programme complexe
-

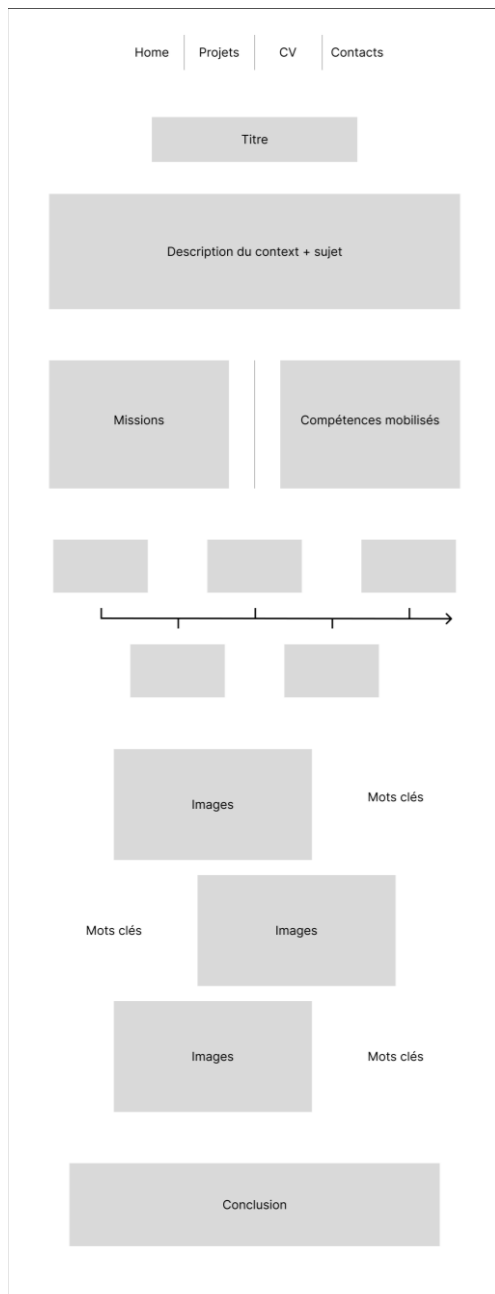
Compétences transversales :

- Respecter une durée donnée pour un projet
- Savoir travailler en équipe
- Savoir développer des applications
- Savoir déboguer un programme

Conclusion :

Ce projet nous a permis de voir à quoi pouvait ressembler un travail de grande ampleur puisqu'il nous a fallu comprendre le programme de quelqu'un d'extérieur, ce qui s'est avéré plus difficile que prévu.

Il nous a aussi familiarisé avec différentes architectures que l'on peut rencontrer et diverses fonctions types, ou encore la récursivité.



Police : Libre Baskerville

Taille : 72 Px (titres) / 48 Px (textes)

/ 36 Px (Header)

