

# Création d'un site web et conception d'armes pour l'ASBL 7RAPL

Travail de fin d'études réalisé en vue de  
l'obtention du diplôme de Bachelier en  
Techniques graphiques



- Promoteur interne > Monsieur Souris Xavier
- Promoteur externe > Monsieur Michaux Loïc



# Remerciements

Avant de vous emmener dans l'aventure de mon TFE, je tiens à remercier Messieurs Souris Xavier et Miller Ivan de m'avoir épaulée et aidée durant mon travail.

Merci également à Monsieur Michaux Loïc, mon promoteur externe et membre-secrétaire de l'association 7rapl, pour avoir pris le temps de suivre mes états d'avancement au fil des mois malgré son emploi du temps très chargé.

Enfin, je souhaite remercier les membres de l'association « Le 7ème Régiment d'Artillerie à Pied de la Ligne » de m'avoir confié la refonte du site web de l'association ainsi que la modélisation de plusieurs armes.

## Sommaire

1	Introduction	P.6
2	Objectifs et Productions	P.8
3	Ancien site	P.10
4	Le nouveau site	P.16
	4.1. Graphisme	P.17
	4.2. 3D	P.35
	4.3. Web	P.42
5	Résultats	P.70
6	Conclusion	P.76
7	Sources	P.78
8	Annexes	P.80



# Introduction



1

Comme vous venez de le remarquer, grâce au sommaire, je vais vous développer les 3 grosses branches de ce travail de fin d'études. Ces 3 parties correspondent aux 3 axes principaux de ma formation. C'était un souhait de ma part de réaliser un TFE qui regroupe tout ce que j'ai appris durant mon bachelier et bien plus.

J'ai décidé de mettre mon TFE au service d'une association belge et même mouscronnoise (ma ville d'origine). Le 7ème Régiment d'Artillerie à Pied de la Ligne (abrégée 7 RAPL), crée en 2012. Son objectif est de promouvoir, au travers de reconstitutions et événements historiques, les mœurs et techniques de combat. Il s'agit d'une association belge de reconstitution historique spécialisée dans le domaine de l'artillerie à pied du premier empire français (de la Révolution française de 1789 à la chute du premier empire 1815).

Cette asbl m'a donné pour mission, la refonte totale de leur site internet afin, notamment, de les trouver facilement sur le web. J'ai regroupé des informations pour un plus large publique cible. L'important étant de pouvoir utiliser ce site sur n'importe quel appareil et navigateur, ce qui n'est pas le cas actuellement.

Envie de voir si je suis arrivée au bout de ce fameux défi ? Alors poursuivez la lecture...

---



# Objectifs et Productions



2



## Objectifs du projet

- > La refonte graphique du site web avec comme mots d'ordre efficacité, simple d'utilisation et responsive
- > La modélisation 3D d'armes sur Solidworks
- > L'animation des armes

## Productions Design

- > Moodboard
- > Wireframes
- > Layouts

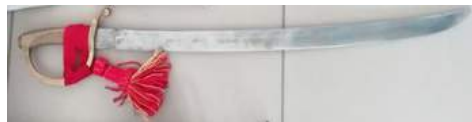
## Productions Web

Site codé en HTML, CSS, JS et PHP comprenant :

- > Icônes de contacts (Emails, Facebook, Youtube, Instagram)
- > Galerie Photos (avec agrandissement lors du clic)
- > Agenda google
- > Maps google
- > Navigation dynamique afin de voyager à travers les différentes pages
- > Présentation de l'ASBL, des membres, des armes et de leurs uniformes.
- > Code en THREE.JS pour rendre interactives les armes 3D sur le site.
- > Une page reprenant les derniers évènements marquants de l'asbl.

## Productions en 3 Dimensions

Modélisation de plusieurs armes 3D



## Logiciels

- > Atom (correspond à notepad++)
- > Laragon (correspond à WAMP)
- > FileZilla
- > Adobe Photoshop
- > Adobe Illustrator
- > Adobe Indesign
- > Solidworks

## Objectifs personnels

- > Augmenter mes connaissances en code
- > Faire du code pur sans plug-in
- > Travailler avec des personnes n'ayant aucune connaissance dans le domaine de l'infographie



Ancien site

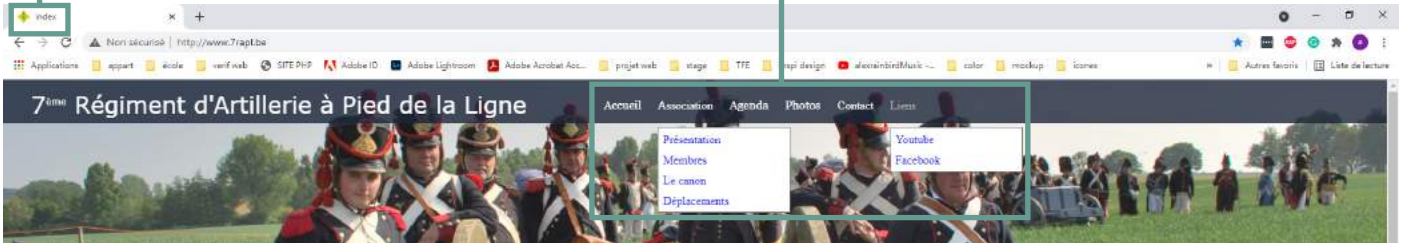


3

## Problématiques

Pas de nom de page correct

- Menu possédant des parties déroulantes laissées en bleuté

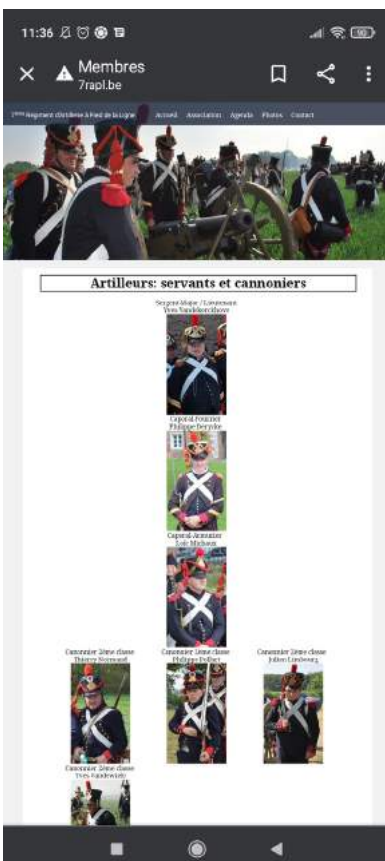


Menu non-responsive et il manque la partie « liens » présente sur ordinateur



Partie news non cliquable

Non-responsive



Hiérarchie des membres difficile à visionner en entier sur un écran

## Code de la page d'accueil

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="fr-b">
  <head>...</head>
  <body class="RWAuto" data-new-gr-c-s-check-loaded="14.1012.0" data-gr-ext-installed style="=> $0
    <form id="XForm" method="post" action="#"></form>
    <div id="XBody" class="BaseDiv RBoth OEPAGEbody OESK_XBody_Default" style="z-index:1000">
      <div class="OESZ OESZ_DivContent OESZG_XBody">
        <div class="OESZ OESZ_XBodyContent OESZG_XBody OECT OECT_Content OECTAbs">
          <div id="WE9530867c05" class="BaseDiv RBoth OEWEPanel OESK_WEPANEL_Default barreMenu" style="z-index:1">
            <div class="OESZ OESZ_DivContent OESZG_WE9530867c05">...</div>
          </div>
          <div id="WE389a12fee9" class="BaseDiv RBoth OEWEcadre OESK_WECadre_Default" style="z-index:1001">
            <div class="OESZ OESZ_DivContent OESZG_WE389a12fee9">...</div>
          </div>
          <div id="WEaba4931da8" class="BaseDiv RWidth OEWEPanel OESK_WEPANEL_Default grandpanneau" style="z-index:1002">
            <div class="OESZ OESZ_DivContent OESZG_WEaba4931da8">...</div>
          </div>
        </div>
        <div class="OESZ OESZ_XBodyFooter OESZG_XBody OECT OECT_Footer OECTAbs"></div>
      </div>
      <div id="WEf67812d591" class="BaseDiv RBoth OEWEMenu OESK_WEMenu_Default" style="z-index: 100000; display: block;">
        <div class="OESZ OESZ_DivContent OESZG_WEf67812d591">

```

Au niveau du code, un programmeur débutant peut directement constater qu'il n'est pas codé de manière optimum.

L'ancien site de l'association a été fait sur openElement.

Le code n'est pas optimal car

> il y a de nombreuses balises div dans d'autres div, ... (cette balise est un conteneur qui permet d'organiser le contenu d'une page web. Il peut être utilisé afin de grouper d'autres éléments pour leur appliquer un style.)

> la partie CSS est mélangée à l'html et ce, dans chaque balise. Il aurait fallu les séparer.

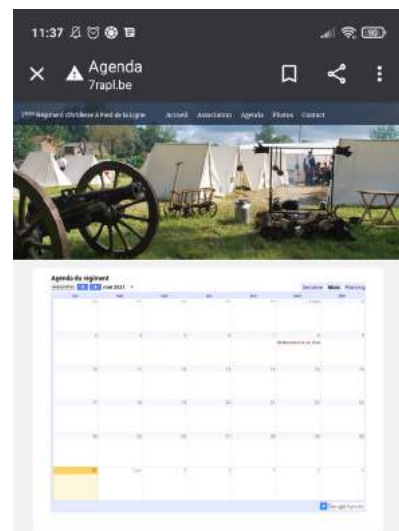
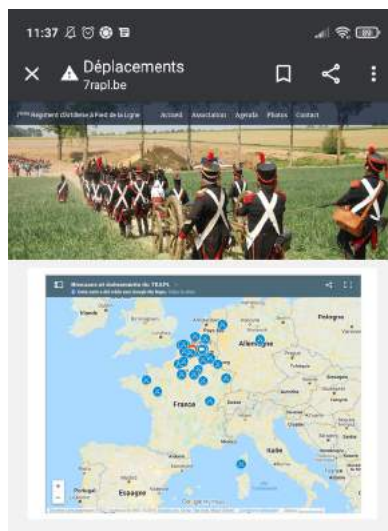
```

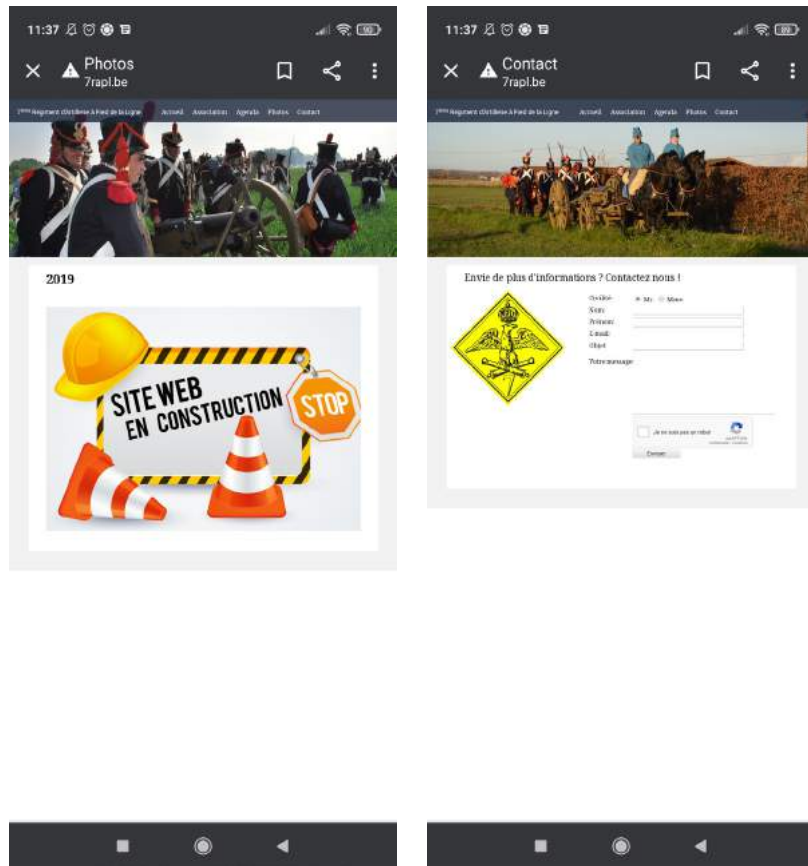
  <div id="WEaba4931da8" class="BaseDiv RWidth OEWEPanel OESK_WEPANEL_Default grandpanneau" style="z-index:1002">
    <div class="OESZ OESZ_DivContent OESZG_WEaba4931da8">
      <div class="OECT OECT_Content OECTRel OEDynTag0" style="overflow:hidden">
        <div class="OERellLine OEAlignL OEAlignB">
          <div id="WE54d0402a1d" class="BaseDiv RWidth OEWEPanel OESK_WEPANEL_Default ombre" style="z-index:1001">
            <div class="OESZ OESZ_DivContent OESZG_WE54d0402a1d">
              <div class="OECT OECT_Content OECTRel OEDynTag0" style="overflow:hidden">
                <div class="OERellLine OEAlignL OEAlignT">
                  <div id="WE6e015426d" class="BaseDiv RWidth OEWEPanel OESK_WEPANEL_Default comment" style="z-index:1001">
                    <div class="OESZ OESZ_DivContent OESZG_WE6e015426d">
                      <div class="OECT OECT_Content OECTRel OEDynTag0" style="overflow:hidden">
                        <div class="OERellLine OEAlignL OEAlignB">
                          <div id="WEa69132c5c5" class="BaseDiv RBoth OEWEcadre OESK_WECadre_Default" style="z-index:1001">
                            <div class="OESZ OESZ_DivContent OESZG_WEa69132c5c5">
                              <div class="OESZ OESZ_Top OESZG_WEa69132c5c5"></div>
                              <div class="OESZ OESZ_Content OESZG_WEa69132c5c5"></div>
                              <div class="OESZ OESZ_Bottom OESZG_WEa69132c5c5"></div>
                            </div>
                          </div>
                        </div>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```



## Les différentes pages (sur mobile)





Actuellement, le site est :

- > non-responsive
- > Navigation et hiérarchie des pages à repenser
- > Design et couleurs peu présents
- > Beaucoup de code pour pas grand-chose
- > Utilisation non simplifiée
- > Page galerie vide
- > Mise en page basique

Attention :

Cette analyse a pour unique but d'informer et de faire comprendre la nécessité d'une refonte du site web. Elle n'a pas pour objectif de rabaisser ou de critiquer la personne l'ayant réalisé. Il est courant d'être confronté à un site comme celui-ci lorsqu'il est réalisé par un individu n'étant pas du domaine de l'infographie ni de la programmation.

Désormais vous connaissez l'apparence de l'ancien site web de l'association. Celui-ci a été fait par une personne de l'asbl, autodidacte, car Il faut savoir qu'aucun membre ne fait partie d'un métier en lien avec l'informatique. Certains sont pompiers, d'autres infirmiers, chimistes, ...

---

Après cette analyse, J'ai commencé à rechercher des inspirations sur des sites comme Behance, Dribbble, Awwwards, Pinterest, ... Pour ensuite pouvoir réaliser des wireframes et layouts.

Mais avant tout, je devais savoir ce qu'allait comprendre le menu ? Quelle serait la hiérarchie des pages ? Quelles informations étaient à mettre en avant ? Et bien d'autres questions. Ce qui est sûr c'est que j'entre dans le corps de mon travail de fin d'études !



## Le nouveau site



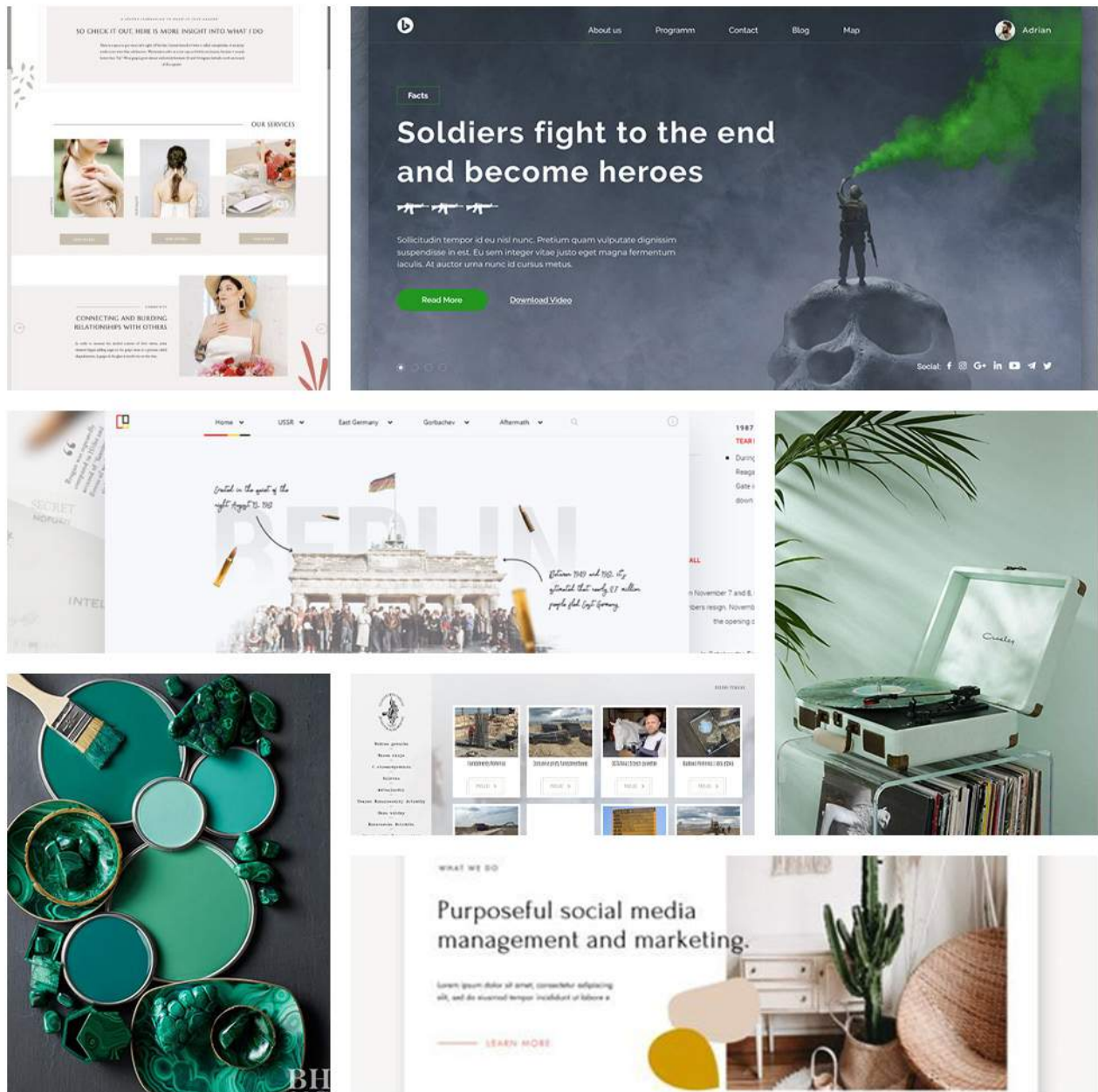
4



## 4.1. Graphisme

Dans un site web, l'étape du graphisme est non-négligeable. L'association m'a demandé un site efficace et simple. Je vais me baser sur ces deux critères et commencer mes recherches !

### Moodboard



Voici mon tableau d'inspiration. Je compte partir sur un design assez linéaire, clair et avec une touche de couleur verte (en accord avec le logo).

## Navigation

Avant de commencer mon étape suivante que sont les wireframes. Il faut déterminer les pages principales avec l'ajout de diverses informations.

Voici le menu qui a été décidé.

Accueil	Informations	Galerie	Association
Il présenterait l'asbl, et montrerait quelques images d'événements de reconstitutions.	Elles comprendraient l'agenda et la carte.	Il y aurait toutes les photos.	La plus grosse page, elle rassemblerait l'histoire de l'asbl, les membres et les armes.

## Propositions Wireframes

Dans un premier temps, j'ai réalisé 4 designs de la page d'accueil afin d'essayer de mieux cerner ce que mon promoteur externe souhaitait niveau visuel.

1

Cette première proposition montre une navigation verticale composée d'icônes. Au clic sur la flèche en haut, la navigation se décale vers la droite afin que les textes correspondent aux icônes.



Voici ce que ça donne une fois la navigation ouverte.

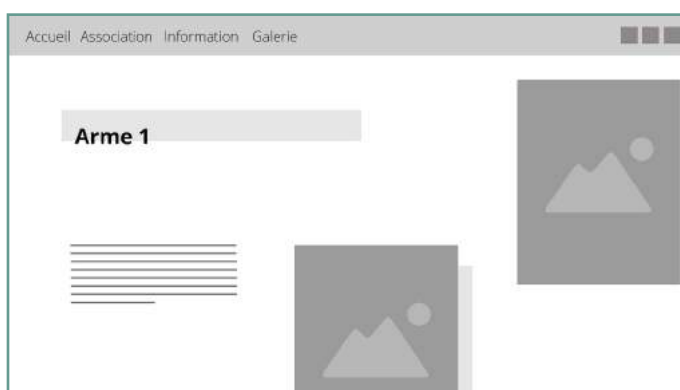
Ici, l'asbl préférerait le principe de slider mais pas la structure de la navigation.



2

Ma seconde idée fut un design composé d'aplats de couleurs.

J'ai continué de manière cohérente sur la page « armes » mais Monsieur Michaux n'était pas convaincu. La navigation ne lui plaisait pas non plus.

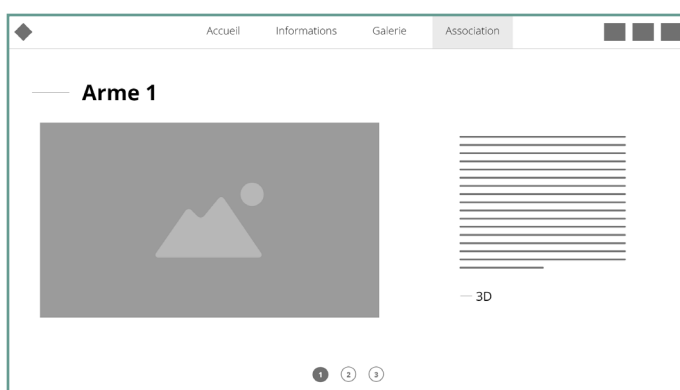


3

La structure du menu plaisait car elle faisait plus aérée et le logo y était inclu.

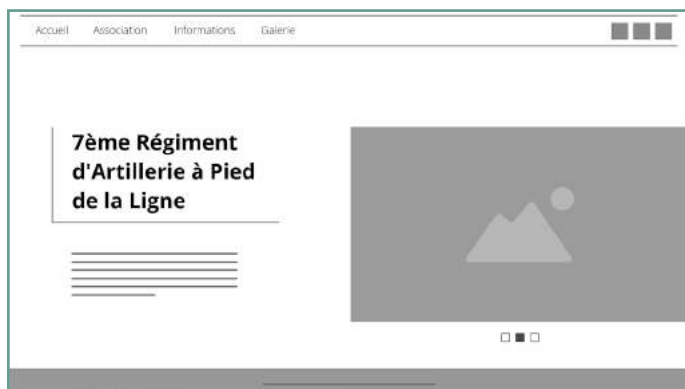
Les armes en mode slider ne convenaient pas.

Cette proposition se basait sur du slider et de la superposition image texte. Ceci a inquiété mon promoteur externe au niveau de la lisibilité des textes sur l'image. Je lui ai donc proposé d'aller voir le site que j'avais réalisé dans le cadre du cours de projet web<sup>1</sup> car les textes superposaient les images sans gêner la lisibilité. Cependant, l'asbl ne souhaitait pas prendre ce risque.



<sup>1</sup> <https://axelleswert.be/musee/index.html>

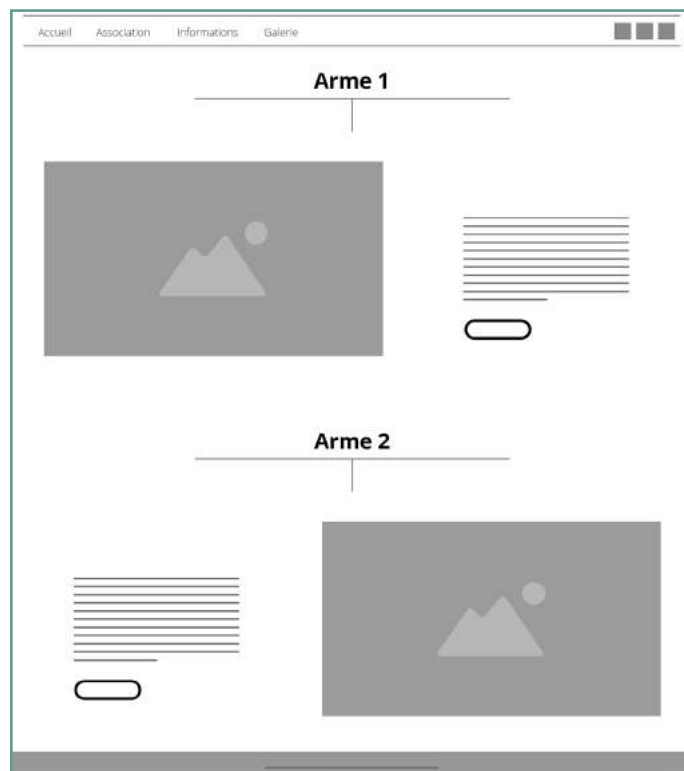
4



Enfin, la proposition qui a le plus retenu leur attention est celle-ci.

Design plus linéaire, slider uniquement sur la page d'accueil.

Seule la navigation ne leur plaisait pas.



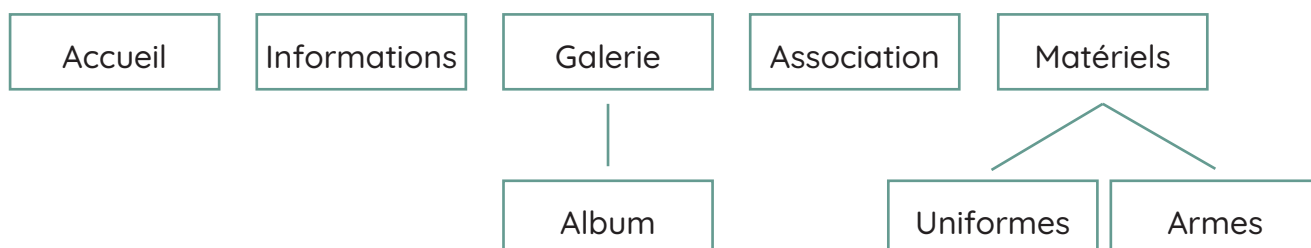
Une fois ces 4 propositions envoyées simultanément, j'ai réalisé un wireframe qui correspondait à toutes les remarques de Monsieur Loïc Michaux.

Parmi ces remarques, il y avait la navigation. Il faut ajouter une page pour les armes car dans la page asbl ça ne fonctionne pas.

La décision fut d'ajouter une page « Matériels » qui comprendrait les armes mais aussi les uniformes et un reportage sous forme de lien externe qui sera redirigé sur le site de « notélé » à l'endroit du documentaire réalisé vers le mois de mai.

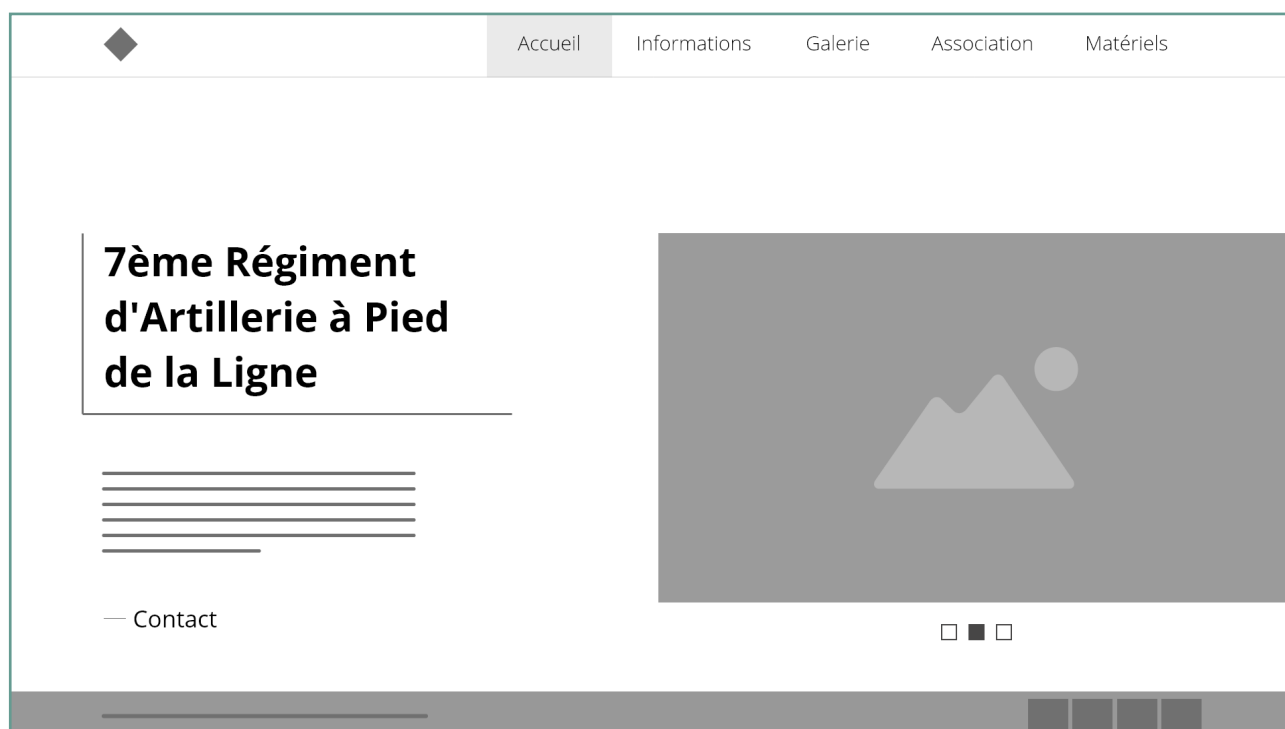
Au vu du nombre de photos que l'association souhaite mettre sur leur site. Mon promoteur externe et moi-même décidons que la galerie comprendrait des albums triés par évènement.

A ce stade, le site comprend 5 pages principales et 3 sous-pages. Voici ce que ça donne concrètement niveau hiérarchie des pages.

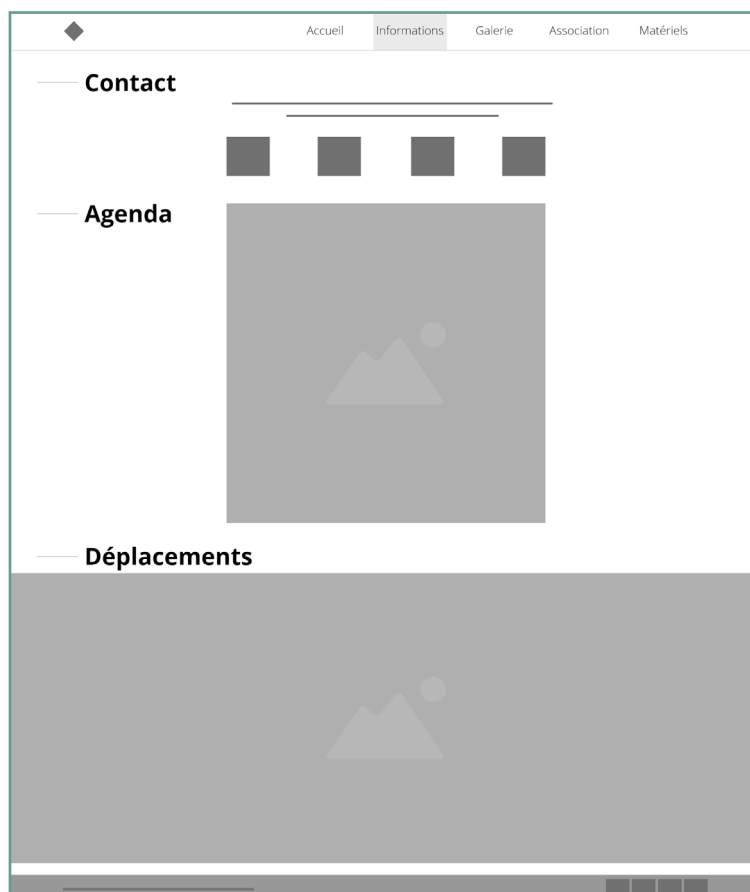


## Wireframe page accueil

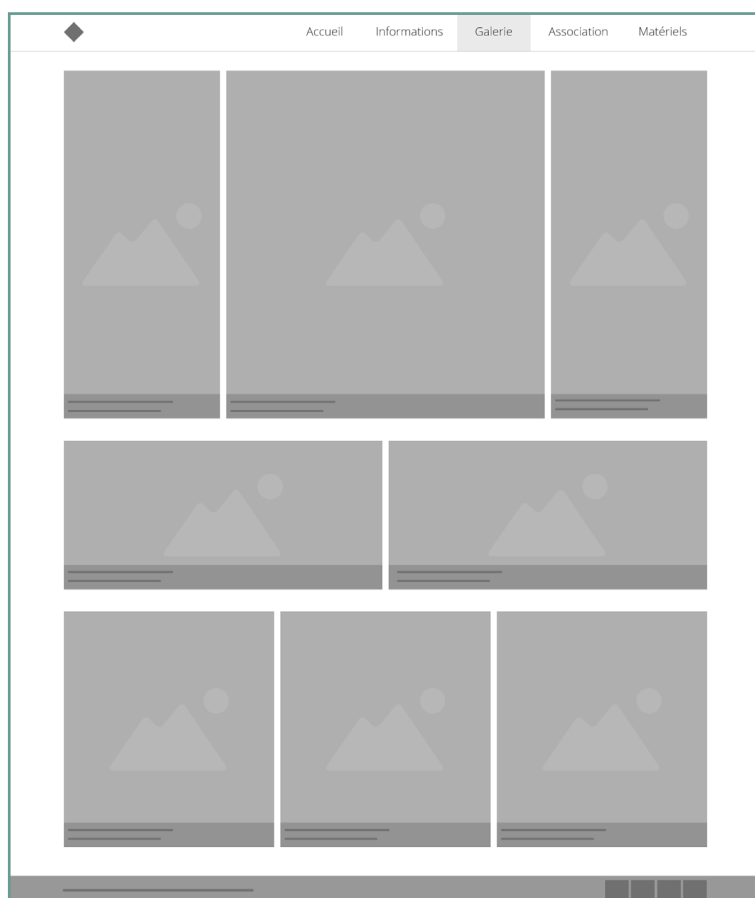
Après toutes ces discussions, le wireframe de la page d'accueil reprend la navigation de la proposition 3, le design linéaire de la proposition 4, le slider que possède toutes les propositions.



## Wireframe page informations

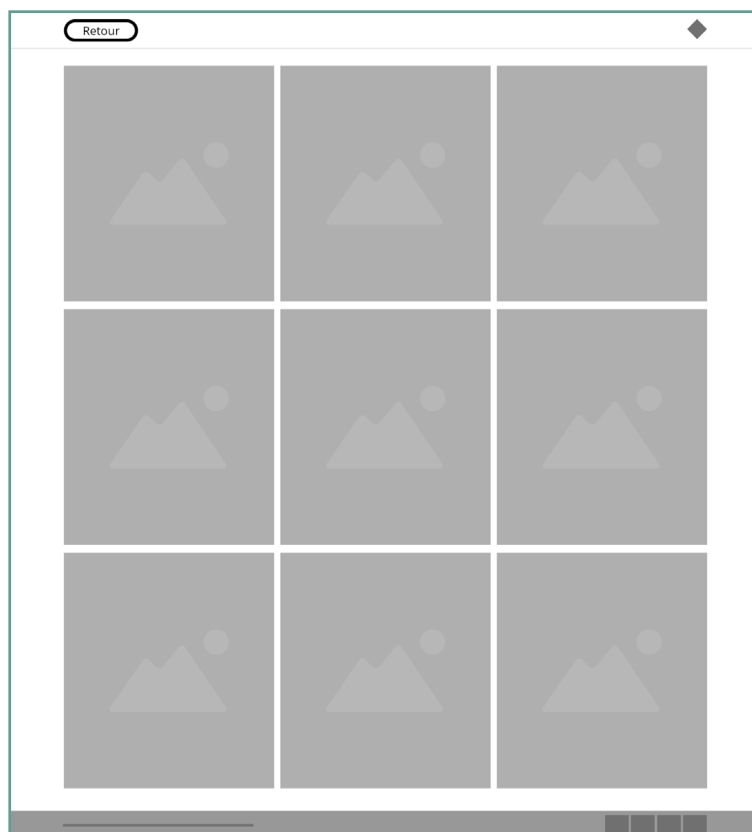


## Wireframes page galerie



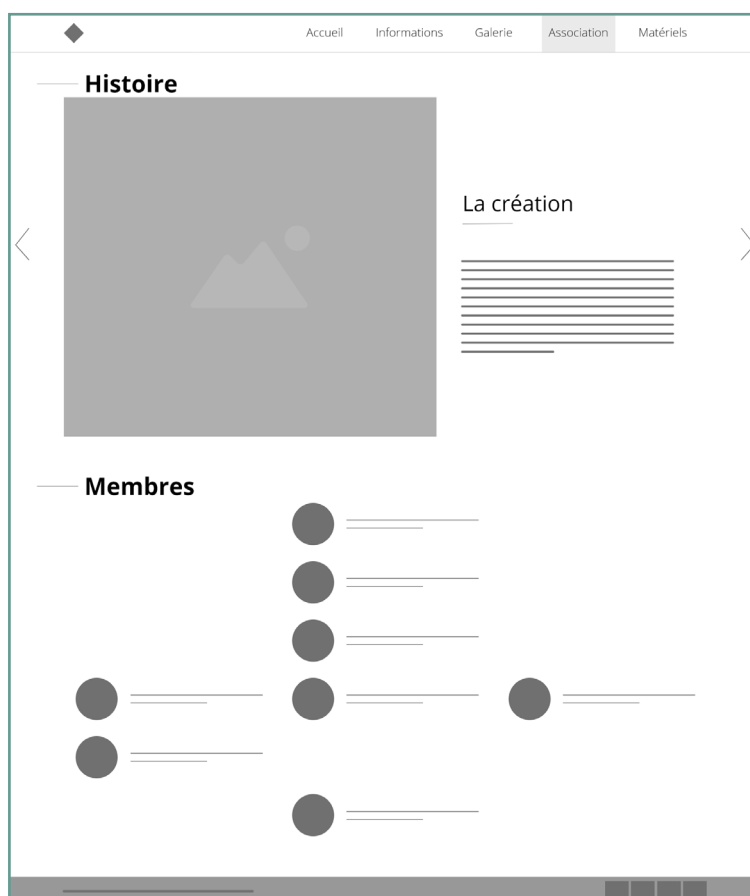
Malgré le design très linéaire, j'ai souhaité ajouter un peu d'asymétrie dans la présentation des albums de la page d'accueil.

Pour la sous-page qui comprendra le contenu d'un album, je reprends le même footer et ici je pars dans du design rectiligne. Le header comprend uniquement un bouton retour et le logo qui renvoie à la page d'accueil.

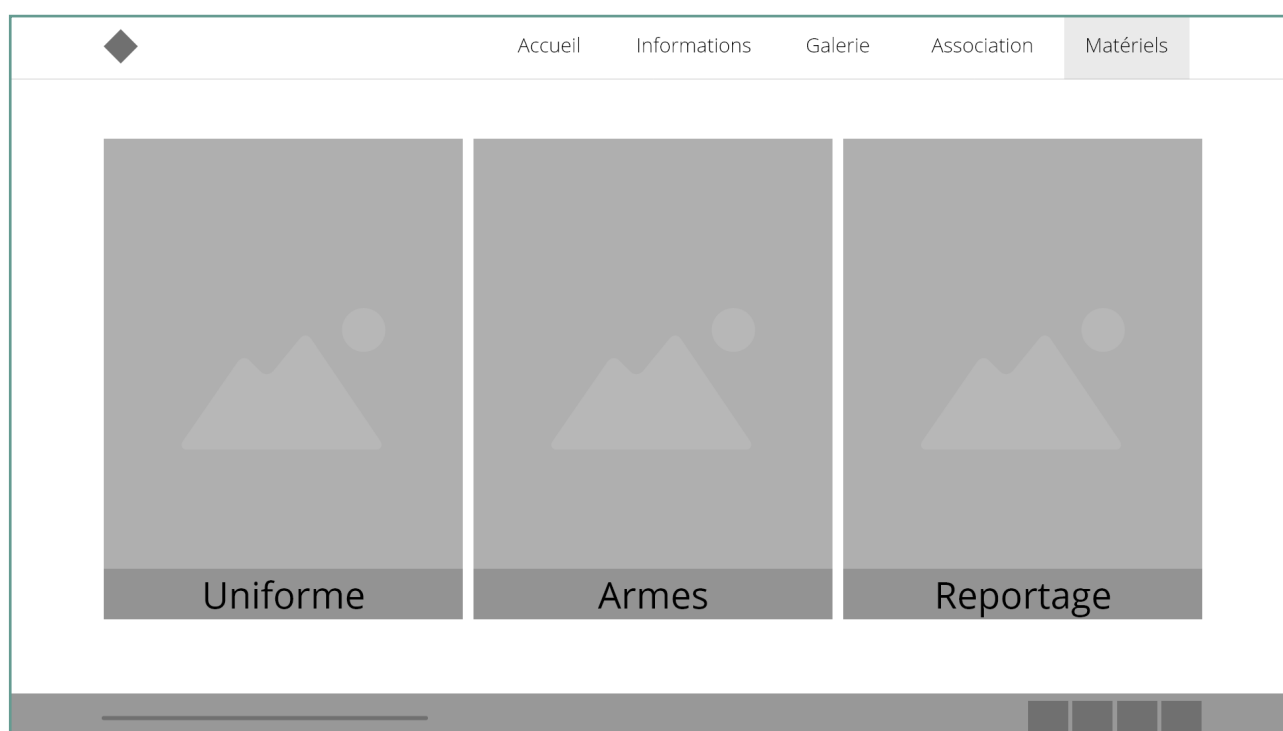


## Wireframe de la page de l'association

Un slider va permettre de traverser l'histoire de l'association et juste en dessous, les membres actuels sont repris suivant l'ordre hiérarchique. (de lieutenant à vivandière)



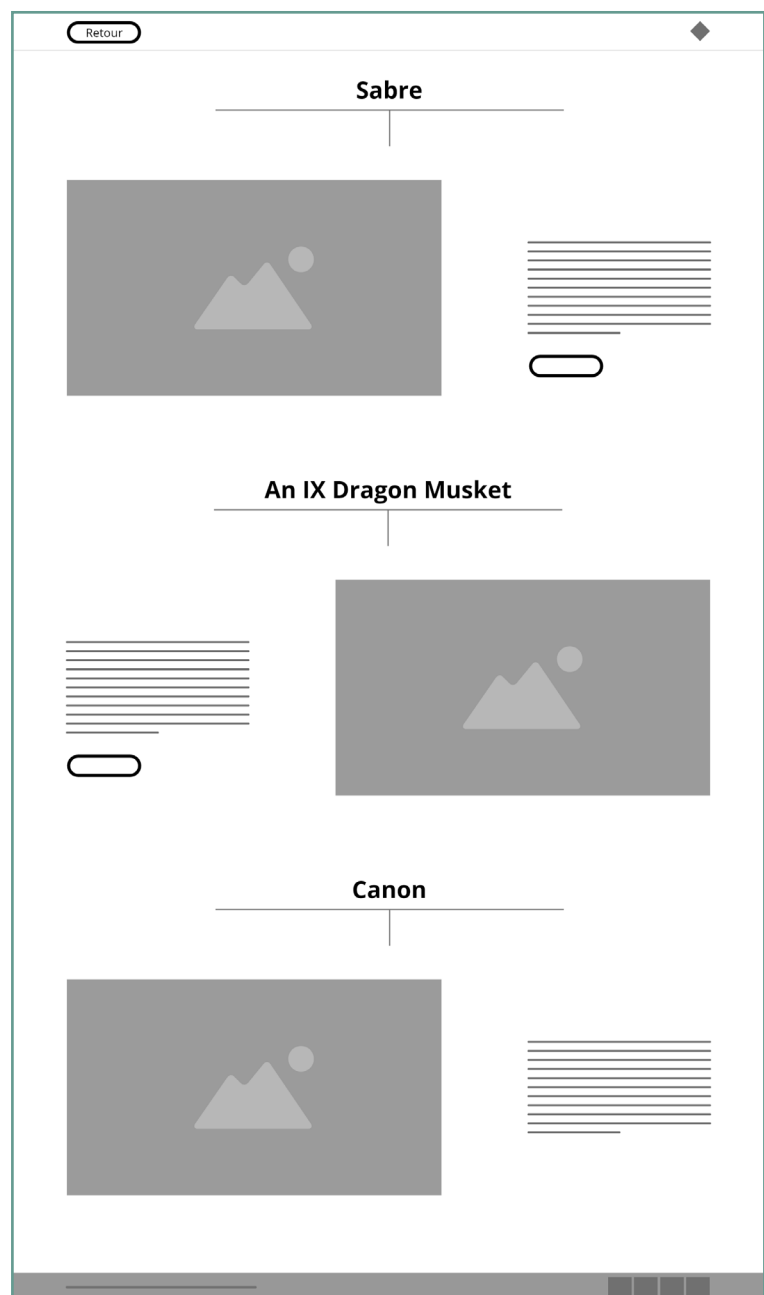
## Wireframe page matériels



La page « matériels » comprend deux sous-pages étant donné que la 3ème image fait une redirection directe (vers le site de « notélé », par exemple).



La partie uniforme se fera comme représentée avec un système de « liste » pour changer le grade.



La section armes se présentera comme telle avec un bouton 3D afin de passer à l'animation 3D.

Les wireframes validés par l'asbl 7rapl, je peux désormais m'attaquer aux layouts des pages.



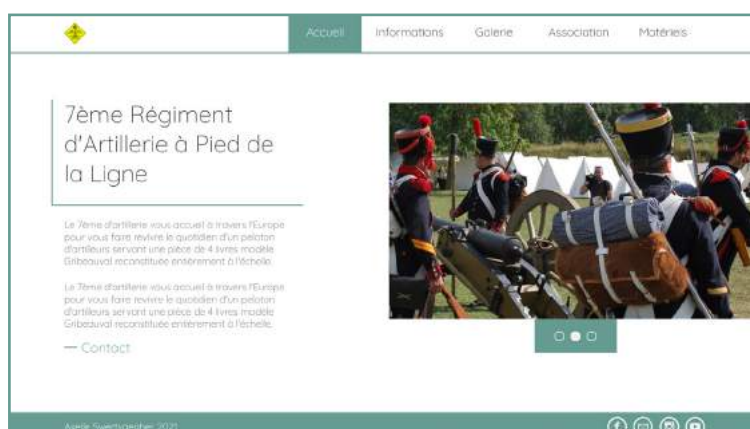
## Layouts propositions

Cette étape consiste à mettre de la colorimétrie, à faire des choix typographiques, en bref des choix visuels et artistiques. Ci-dessous, vous pouvez observer les 3 propositions de la page d'accueil. 3 colorimétries et typographies différentes. Je me suis basée sur la couleur du logo pour mes recherches.



Typographie > Open Sans  
Colorimétrie (en hexadécimal) >  
#E39F3E  
#DEDDCB

Typographie > Quicksand  
Colorimétrie (en hexadécimal) >  
#659B91  
#FFFFFF (blanc)



Typographie > Archive  
Colorimétrie (en hexadécimal) >  
#BEE0E4  
#659B91

L'asbl a eu le devoir de faire un choix. L'orange n'a pas tellement plu contrairement au vert et au bleu. Il a été décidé de prendre la 2ème proposition et d'ajouter le bleu aux interactions de survol, à l'écran de chargement,...

Niveau typographie, c'est la Quicksand qui a gagné.

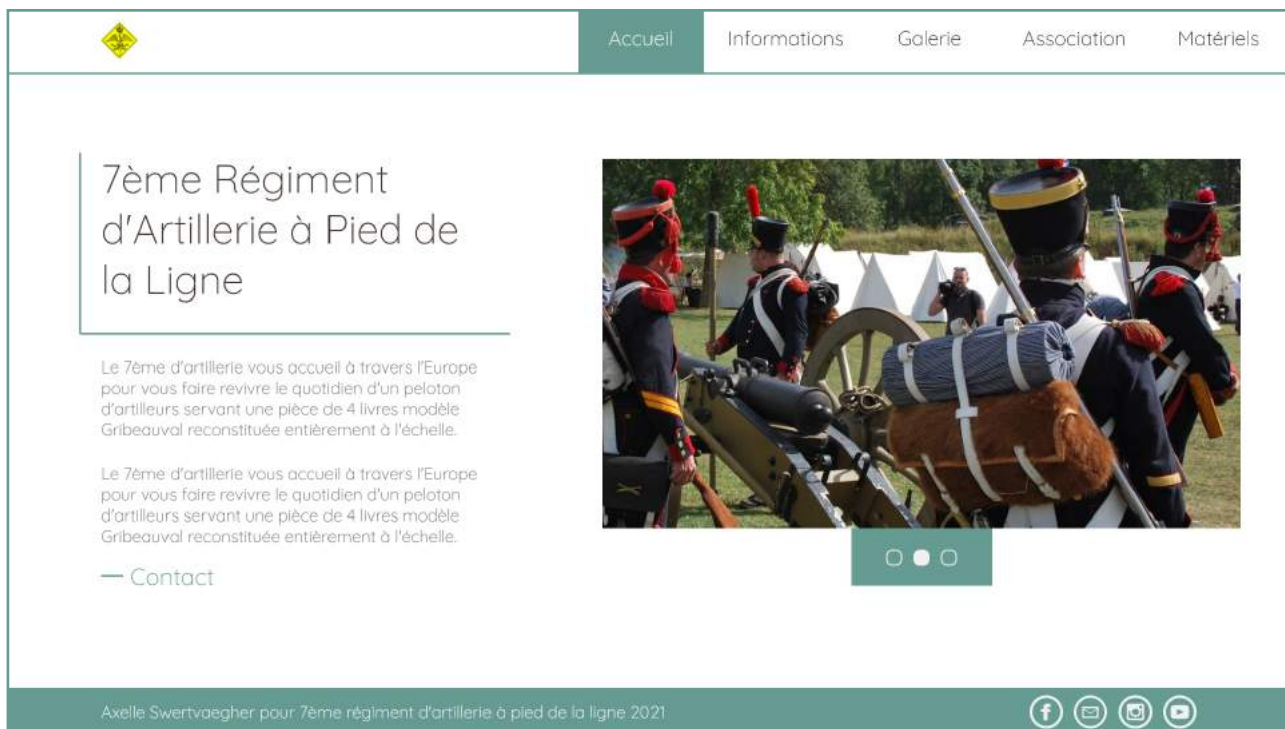
## Layouts finals

Colorimétrie > couleur principale **#659B91**  
couleurs de textes **#FFFFFF** ( blanc ) et **#444444** ( gris anthracite )  
couleur d'interaction **#BEE0E4**

Typographie > Quicksand

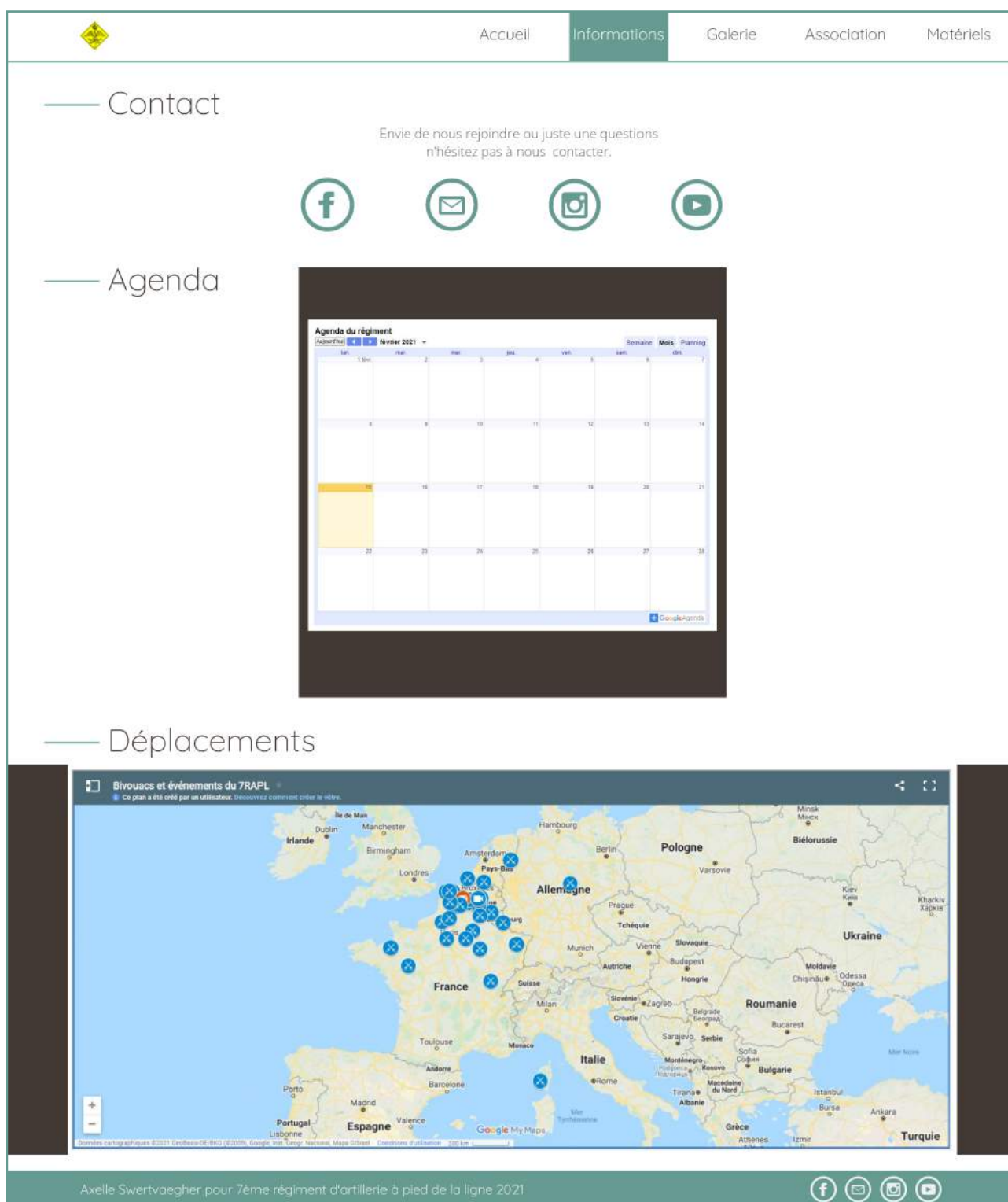
1234567890  
AZERTYUIOPQSDFGHJKLMWXCVCBN  
azertyuiopqsdfghjklmwxvcbn  
&é»'(§è!çà

Page « Accueil »



Cette page est la plus importante car c'est celle qui sera principalement référencée. Elle doit comporter les mots-clés qui définissent l'asbl « 7ème Régiment d'Artillerie à Pied de la Ligne ». Des mots comme : reconstitutions, 7RAPL, artillerie, canon, historique,...  
Le bouton contact mène à la page « informations ».


Le but principal est d'informer les visiteurs du site en leur indiquant les réseaux sociaux, l'agenda google qui indique les évènements et les déplacements.






Chaque image « résume » un album. Une indication du lieu et de l'année est notée en bas de chaque vignette. Au survol, les images se mettent en niveau de gris afin de montrer visuellement qu'elles sont cliquables.



[Accueil](#)
[Informations](#)
[Galerie](#)
[Association](#)
[Matériels](#)




25 Juin 2017 - Ville




10 Sept 2016 - Ville




27 Août 2016 - Ville




31 Juil 2016 - Ville




4 Juin 2016 - Ville



23 Avr 2016 - Ville






20 Juin 2015 - Ville



5 Juil 2014 - Ville

Axelle Swertvaegher pour 7<sup>ème</sup> régiment d'artillerie à pied de la ligne 2021
 



Retour

Axelle Swertvaegher pour 7<sup>ème</sup> régiment d'artillerie à pied de la ligne 2021

Une fois dans l'un des albums, les images apparaissent. Au survol elles passent également en niveau de gris pour indiquer qu'on peut les agrandir.

Une fois la photo agrandie on peut revenir à l'album à l'aide de la croix. L'agrandissement est possible sur pc et tablette mais pas sur téléphone.




[Accueil](#)
[Informations](#)
[Galerie](#)
[Association](#)
[Matériels](#)

## Histoire



### La création

Le 7ème Régiment d'Artillerie à Pied de la Ligne vu le jour le 7 septembre 2012 dans la région de Dottignies en Belgique sous forme d'une association sans but lucratif (ASBL). Son but premier était de faire revivre et d'entretenir par ses activités de reconstitution historique napoléonienne, l'esprit et les valeurs humaines des artilleurs sous l'Empire à travers l'Europe. Très vite le régiment perce dans le domaine de la reconstitution historique sur le territoire belge grâce à son canon modèle Gribeauval de 4 livres qu'il tracte partout où il va.

## Membres



Sergent-Major / Lieutenant  
Yves Vandekerckhove



Sergent  
Loïc Michaux



Caporal-Fourrier  
Philippe Derycke



Tambour  
Yannis Bouckaert



Canonnier 1ère classe  
Philippe Pollart



Canonnier 1ère classe  
Julien Limbourg



Canonnier 2ème classe  
Thierry Normand



Canonnier 2ème classe  
Quentin Depauw

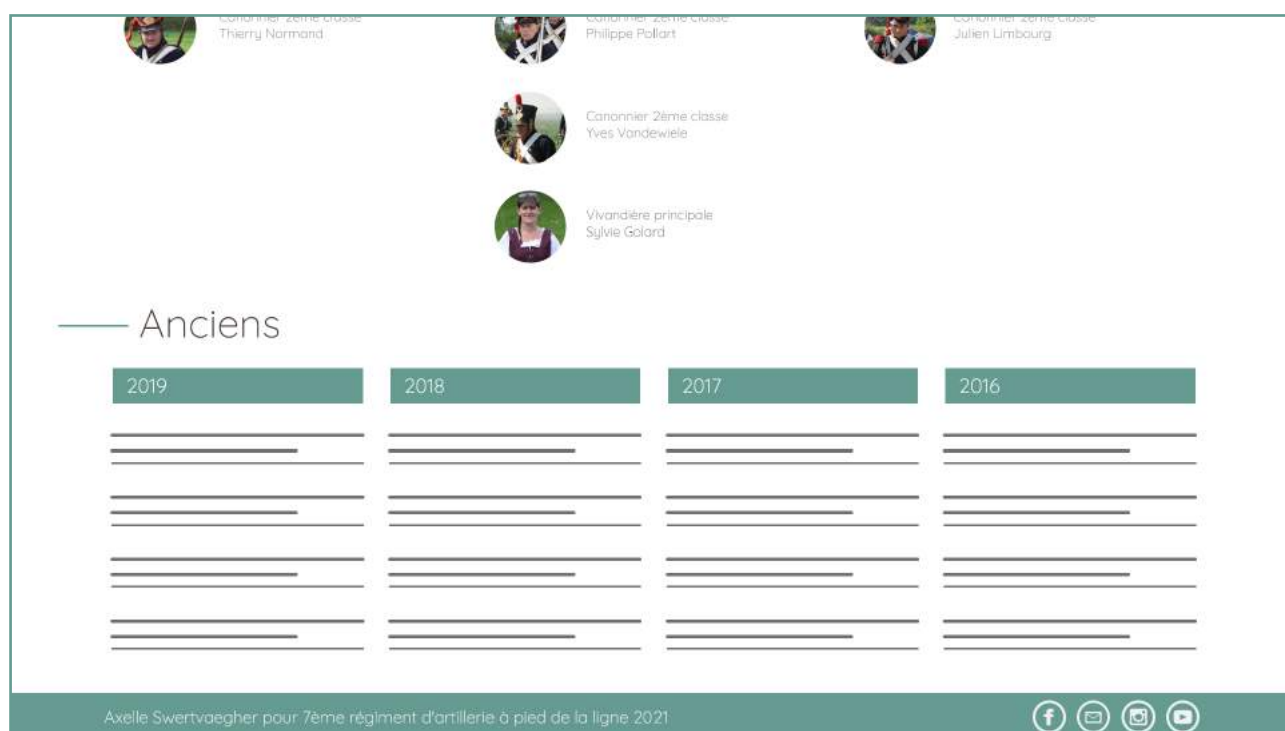


Canonnier 2ème classe  
Vandewiele Yves

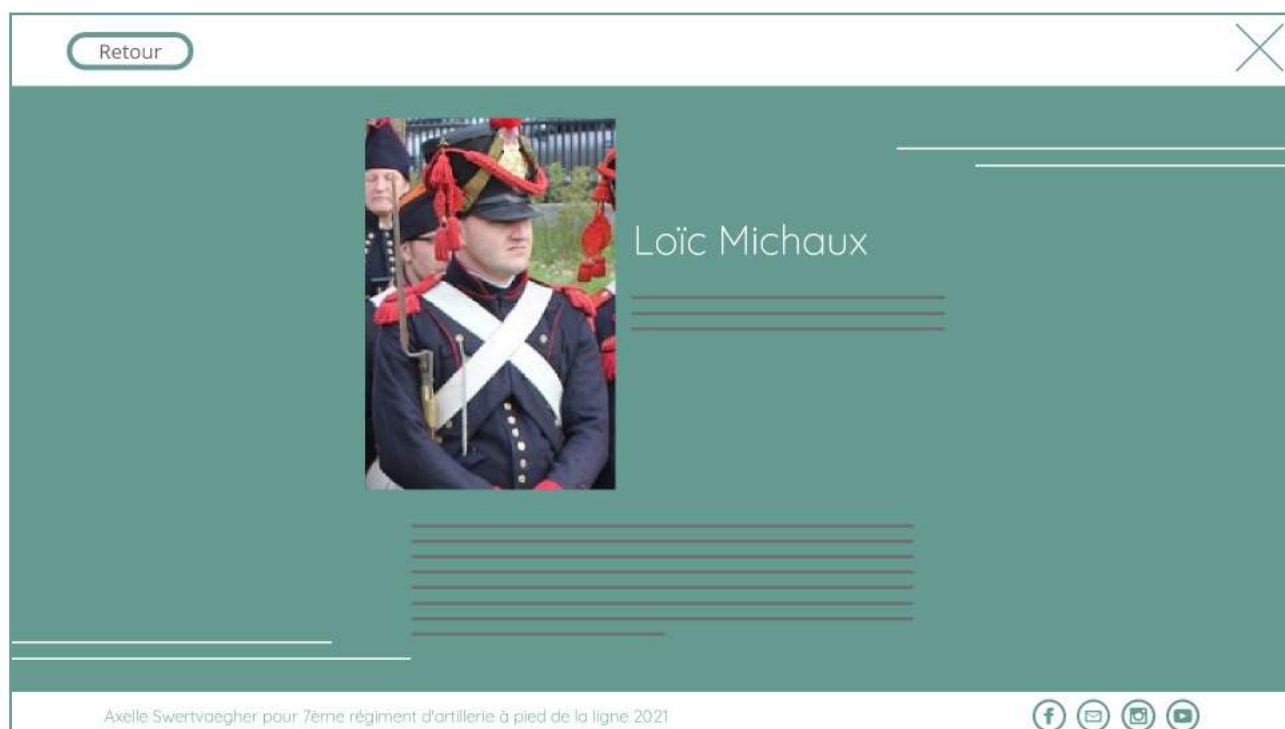


Vivandière principale  
Sylvie Golard

L'histoire, sous forme de slider, permettra de condenser l'information sur une partie de la page. Les membres sont placés hiérarchiquement, du plus haut gradé aux grades les plus bas.




Quelques semaines avant la remise de mon TFE, l'asbl a décidé qu'il fallait rendre hommage aux anciens membres, de ce fait j'ai complété mon layout comme vous pouvez le voir ci-dessus.






Afin de compléter la présentation des membres, ils deviennent cliquables sur la page « association ». Cet événement va guider vers une sous-page qui inclura le grade, le sobriquet (surnom), un petit texte écrit par la personne,...







Ici, 3 images guident vers des sous-pages ou un lien externe. Au survol, l'image passe en niveau de gris, comme le montre la 3ème vignette, afin de montrer visuellement qu'il faut cliquer.


[Accueil](#)
[Informations](#)
[Galerie](#)
[Association](#)
[Matériels](#)






[Uniforme](#)
[Armes](#)
[Reportage](#)


Axelle Swertvaegher pour 7ème régiment d'artillerie à pied de la ligne 2021

Ci-dessous, la première sous-page se nomme « uniforme ». A l'aide d'un menu déroulant, on pourra choisir un grade et son uniforme correspondant. Le but de cette page est purement informatif. Elle comprend une photo de l'uniforme, le nom du grade, des explications diverses,...


[Retour](#)




[Choisir un uniforme](#)



Lieutenant

Axelle Swertvaegher pour 7ème régiment d'artillerie à pied de la ligne 2021




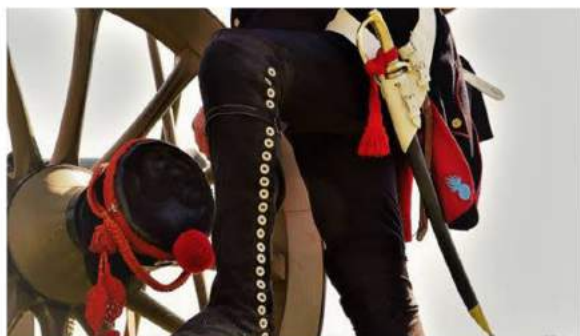



La seconde sous-page a le même but sauf qu'elle concerne les armes utilisées par l'association. Un bouton pour passer en 3D est présent. La 3D sera codée en THREE.JS.

Retour




## Sabre






3D


## An IX Dragon Musket




3D




## Canon

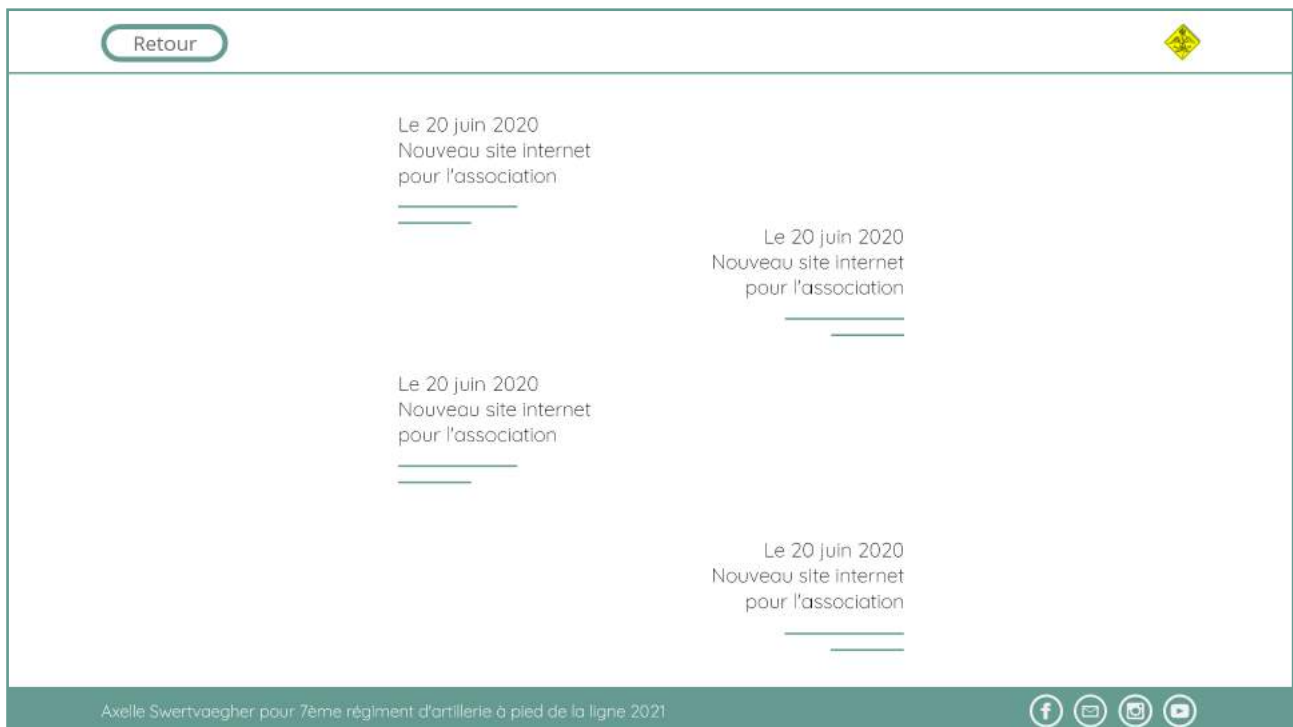




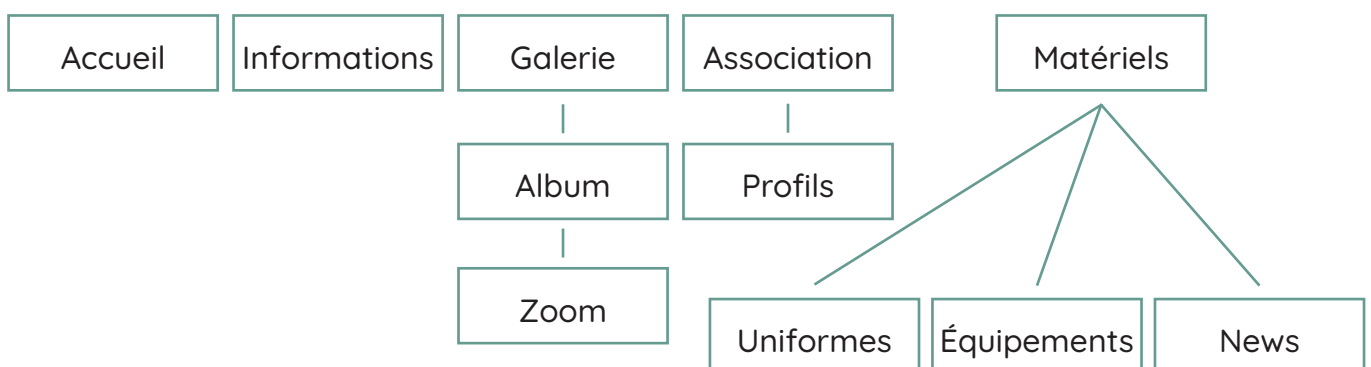
Axelle Swertvoegher pour 7ème régiment d'artillerie à pied de la ligne 2021



Au vu de la situation actuelle, le reportage n'a pas pu être filmé. Le titre de la sous-page va être renommé « news ». Cette sous-page comprendra les derniers changements au sein de l'asbl.



## Hiérarchie définitive du site

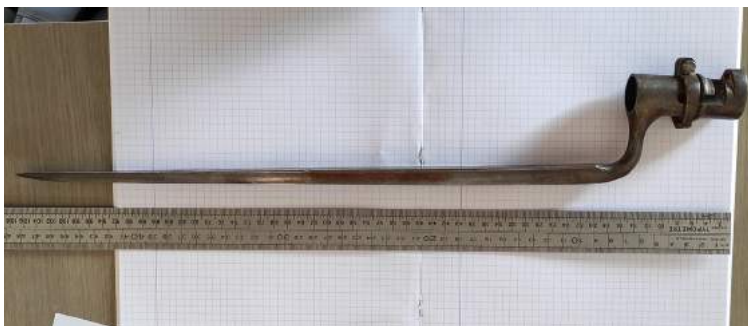
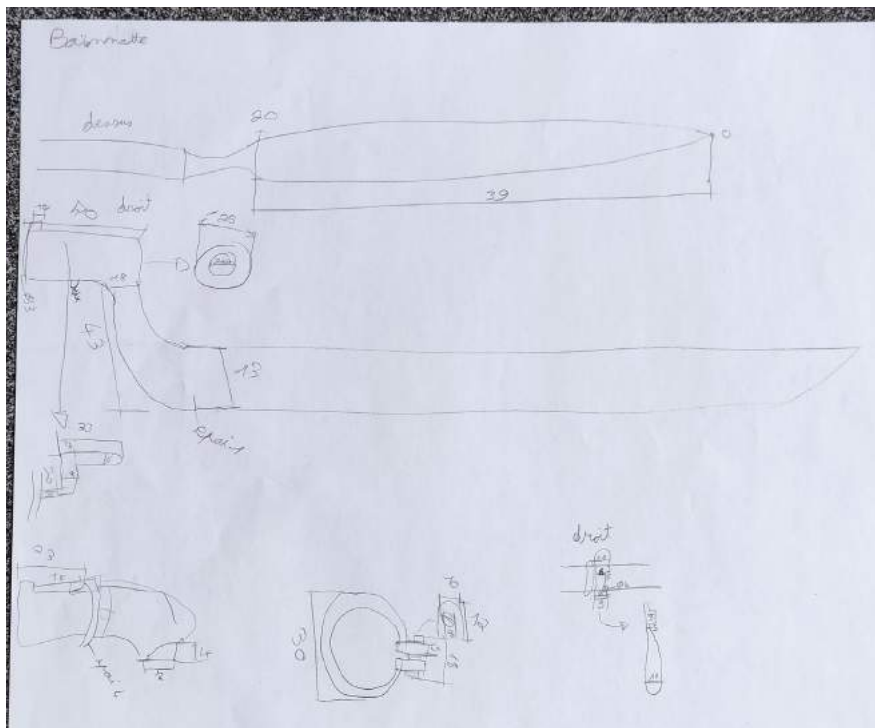


A présent, le site de l'asbl 7rapl comprend 5 pages principales et 6 sous-pages. La partie layout est terminée. Désormais, il va falloir passer à la 3D puis au code.

## 4.2. 3D

Avant de modéliser les armes en 3D, je dois faire de la métrologie. Cette étape est primordiale pour que la modélisation 3D soit la plus fidèle possible à l'arme dans la réalité. Je me suis donc attelée à mesurer la baïonnette et le sabre, les deux armes disponibles. Je n'ai pu les avoir entre les mains qu'une seule fois. De ce fait, j'ai dû faire preuve d'efficacité et de perspicacité dans ma prise de note sur le matériel.

### Baïonnette



Voici mes relevés de mesures pour la baïonnette et quelques photos. Cette arme est en acier et elle est complexe au niveau de son « coude ». J'ai décidé de réaliser mes modélisations sur Solidworks, ce qui m'a valu pas mal de contraintes au niveau des courbes.

La baïonnette comporte 3 pièces > une lame, une attache et une vis.

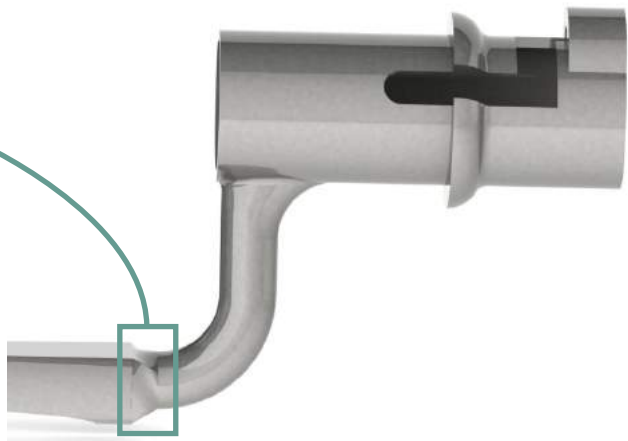


La lame est triangulaire. Pour s'attacher à un fusil, elle a un cylindre percé par un chemin. Un coude relie la partie triangulaire à la partie cylindrique.

La lame est modélisée grâce à une esquisse que j'ai extrudée avec un paramètre de dépouille pour finir en pointe.

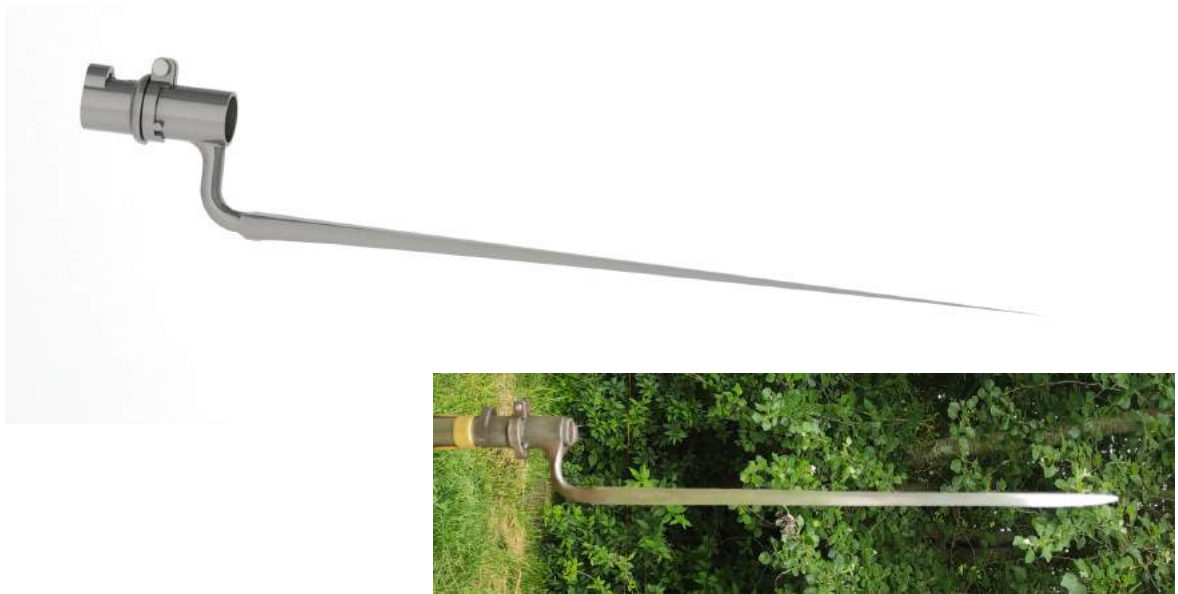
Le coude est l'élément le plus complexe que j'ai dû modéliser sur cette arme. C'est grâce à la fonction « base lissée » que j'ai pu réaliser la jonction entre la lame triangulaire et le début du coude.

Afin de dessiner correctement la courbure, j'ai effectué un balayage composé d'un profil et d'une ligne directrice.



Pour les deux autres éléments, ils sont composés d'extrusions et d'enlèvements de matières principalement. Afin que la vis tienne dans l'attache, un filetage est effectué sur les deux parties. Cette fonction est très utile, je l'ai apprise lors de mon stage.





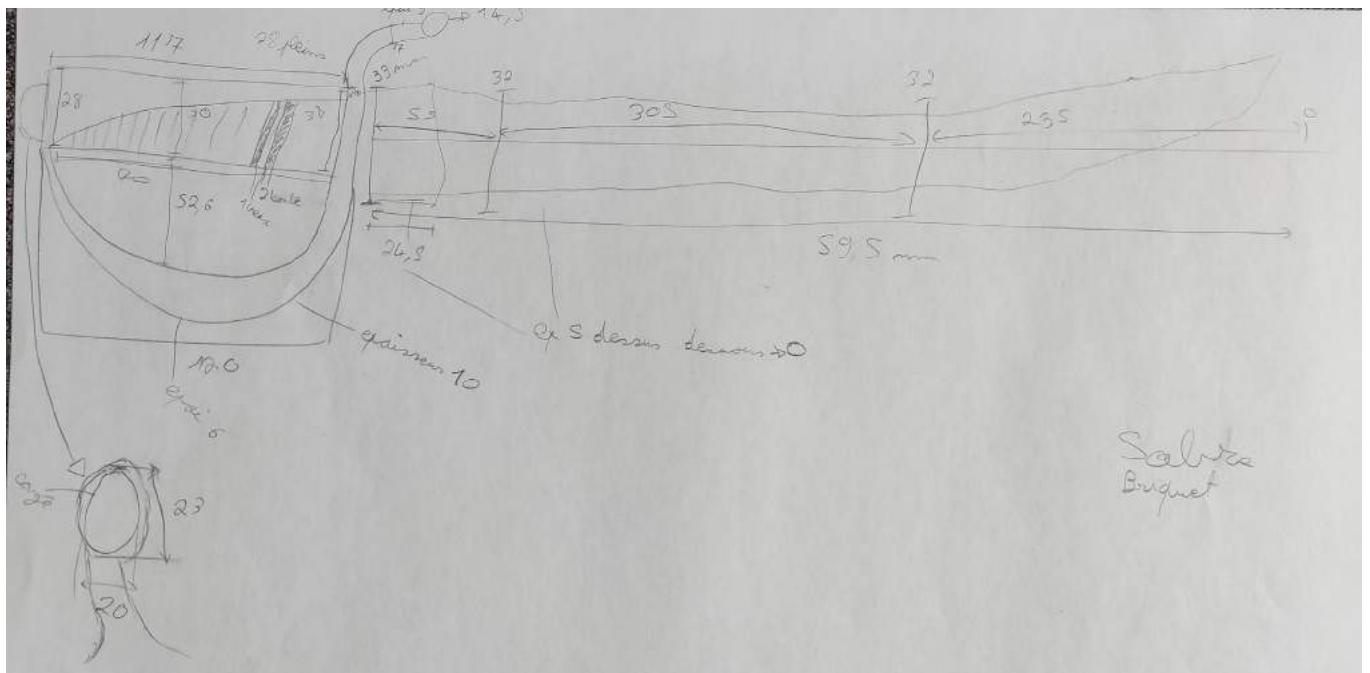
Suite à l'assemblage des 3 pièces, la baïonnette est finalisée.

### Sabre Briquet



Attaquons-nous désormais à la seconde arme qui se nomme le sabre briquet. Contrairement à une épée, la lame est courbée. Sur les photos ci-dessus, vous pouvez clairement voir que le sabre se compose de deux parties avec deux matières différentes. Une première en acier et une seconde en laiton. Le tissu que vous voyez ne sera pas modélisé car il change en fonction du grade de la personne possédant l'arme.





La métrologie a été assez périlleuse car il est difficile de mesurer des courbes sans matériel de précision.

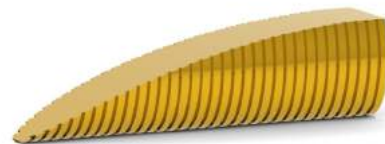


Pour la partie acier, la lame, commence de manière rectangulaire et finit par une pointe. Afin de réaliser le coupant j'ai chanfreiné les deux cotés d'un rectangle.

La partie laiton, quant-à-elle, je l'ai divisée en deux éléments. Après avoir fait une arme assez complexe durant le cours de Bac 3 CAO et après mes 3 mois de stage, je connais bien Solidworks.

Pour éviter la surcharge du logiciel, je fais une pièce qui représente la poignée sans l'intérieur et une autre représentant uniquement l'intérieur.

Une fois l'assemblage réalisé, on ne remarquera pas que ce sont deux pièces séparées.





L'assemblage réalisé, le sabre briquet est finalisé.

Pour chacune des pièces, il faut désormais réaliser une mise en plan, vous pouvez les retrouver en annexe. Grâce à mon stage chez BBK, j'ai appris à optimiser les mises en plans, c'est pour cela qu'elles ne sont pas forcément pareilles que celles vues au cours de DAO.

Ces deux modélisations vont apporter de la dynamique au site web. Vous verrez avec quelle technologie les armes ont été ajoutées. Prochaine étape de mon TFE et ce n'est pas la plus simple ! Le codage des pages du site internet.

J'ai décidé de réaliser les rendus sur Solidworks car je trouve que c'est une bonne qualité de rendu et que les matériaux, ici acier et laiton, sont ressemblant. Bien évidemment ce ne sont que des rendus, mais avec le logiciel on peut voir le poids de l'arme, son centre de gravité,...

Il est vrai que j'aurais pu utiliser Keyshot qui est un logiciel spécialement fait pour tout ce qui est rendu. Mais, n'ayant pas la licence, je ne souhaitais pas l'utiliser. Je l'avoue, mon côté cartésien l'a emporté.

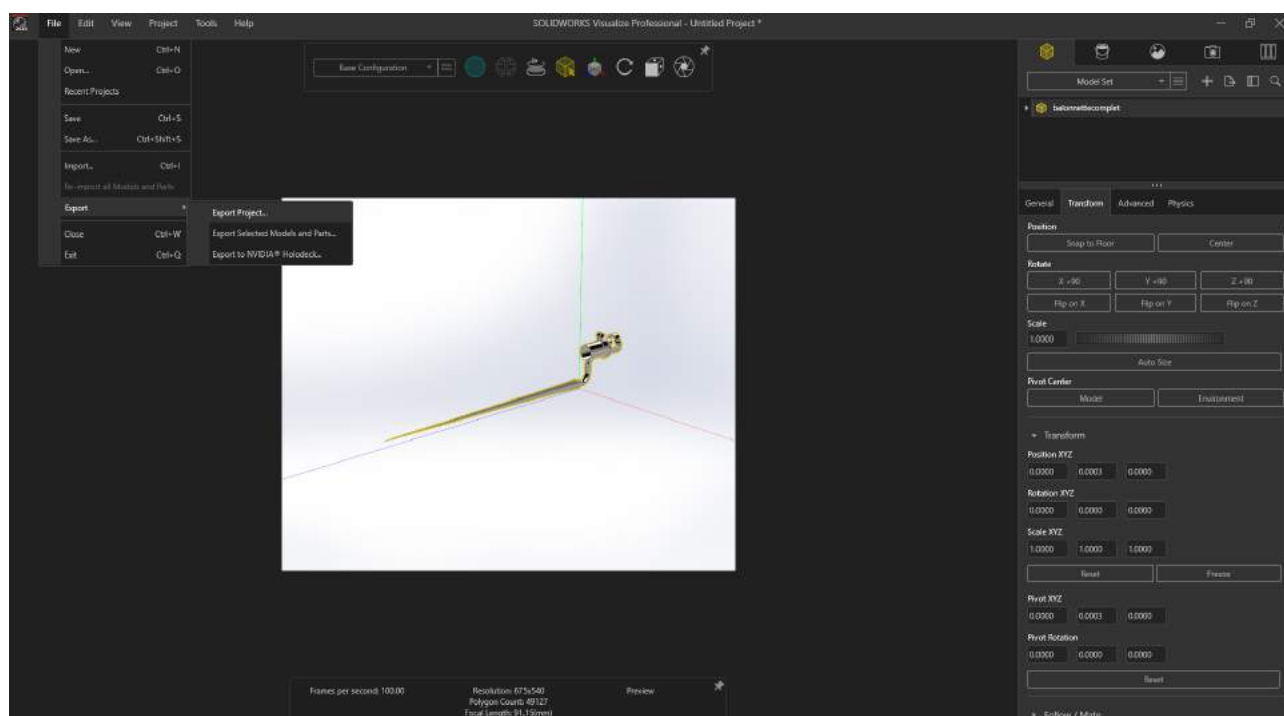
Apartée >

Je souhaite vous expliquer pourquoi j'ai décidé de travailler sur un logiciel tel que Solidworks au lieu de Maya. Il faut savoir que personnellement, je suis un peu comme les membres de l'association 7rapl, c'est-à-dire une personne avec une personnalité rationnelle. J'aime que tout soit correctement réalisé et correspondant le plus possible à la réalité. Maya ne correspondait pas à ma personnalité et donc au Travail de fin d'études que je souhaitais rendre.

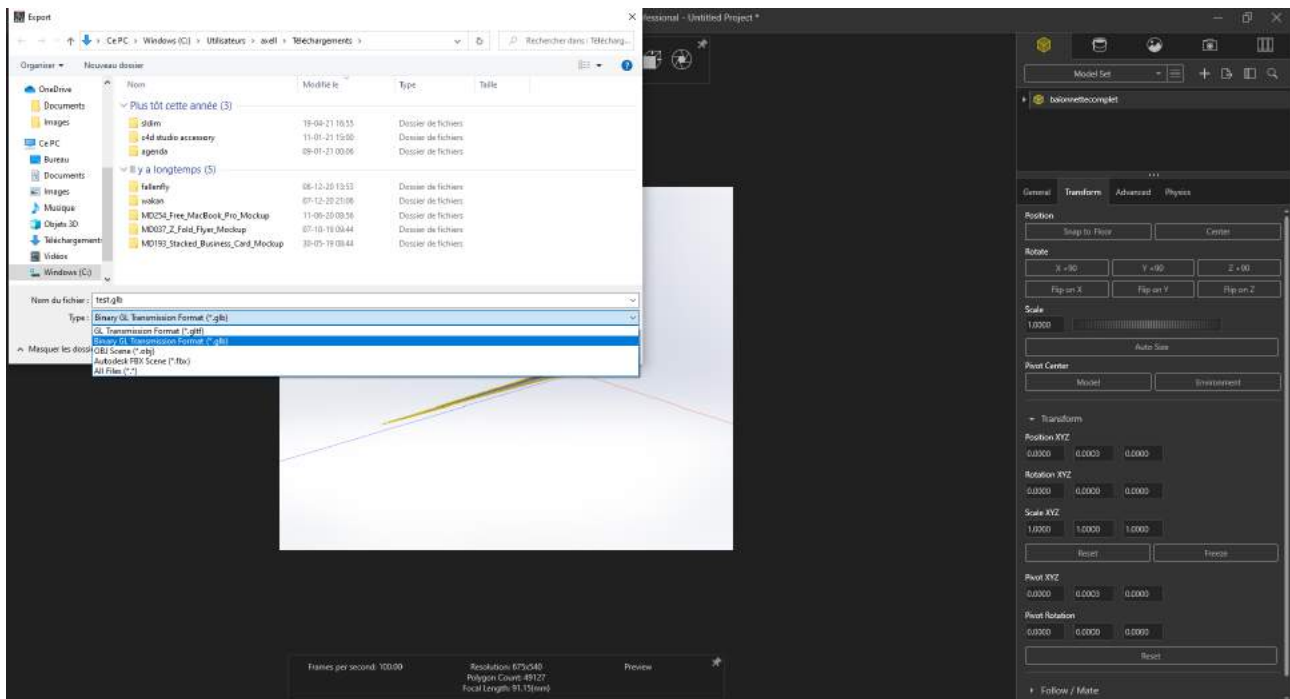


Au vu du code en [three.js](#), les fichiers Solidworks doivent être dans un certain format. Au lieu de l'extension `.SLDPRT`, ce doit être `.GLB`.

Afin d'obtenir celle-ci, j'ouvre Solidworks Visualize. J'importe mon arme dans ce logiciel. Après quelques réglages, je procède à l'export.







Plusieurs exports sont possible mais celui qui nous intéresse est le **.GLB**. Ce format transforme la 3D en fichier binaire ce qui la rend exploitable sur un site web. L'avantage est que ce fichier est très léger, il affichera son contenu rapidement car c'est un document compact.

A titre d'exemple, mon fichier «sabre.glb» a un poids de 439ko.

Désormais nous sommes prêts à attaquer la partie web.

## 4.3. Web

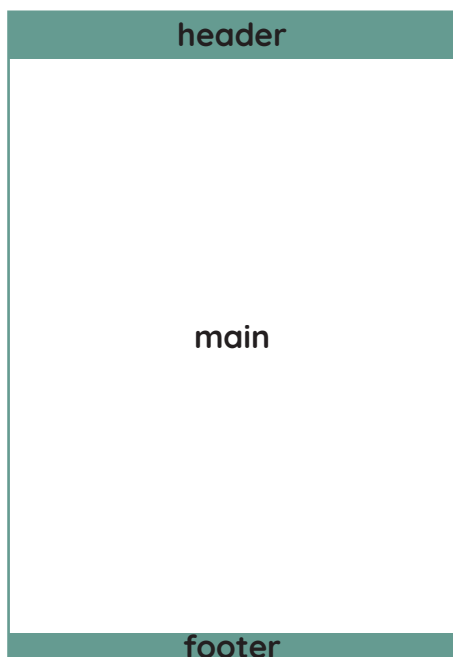
L'univers du code est vaste. De multiples technologies et langages de programmations existent. J'ai fait le choix de tout coder par moi-même, sans utiliser de plug-in ou de sites comme Wordpress et autres.

Je vais coder en HTML, CSS, Javascript, JQuery et PHP donc avec 5 langages appris durant mon cursus. Pour la 3D interactive je vais utiliser une bibliothèque de Javascript qui s'appelle Three.js.

Mais avant de voir ce que c'est, je vais brièvement vous expliquer comment se nomme les différentes parties d'un site web et des codes spécifiques aux pages.

### Structure d'un site web

Afin de pouvoir comprendre la structure du site il faut savoir qu'il y a 3 parties appelées « header », « main » et « footer ». Ces 3 mots se retrouvent dans le code HTML sous forme de balises.



```

<!DOCTYPE html>
<html lang=«fr»>
<head>
</head>
<body>
  <header>
  </header>
  <main>
  </main>
  <footer>
  </footer>
</body>
</html>

```

Partie invisible sur le site web.

Partie visible du site, divisée en 3 parties.

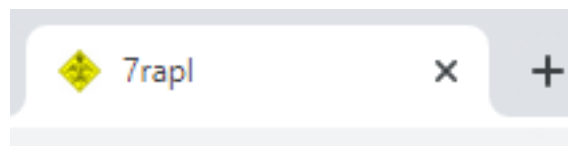
## Contenu Head

Entre ces deux balises, je vais y mettre le titre de la page, des méta-balises qui vont fournir des informations au navigateur et des liens vers d'autres codes.

J'ai choisi des méta-balises telles que :

- > Le format d'encodage des caractères  
`<meta charset=«utf-8»>`
- > Les informations données aux robots d'indexation  
`<meta name=«robots» content=«index, follow»>`
- > Le nom et prénom de l'auteur  
`<meta name=«author» content=«Swertvaegher Axelle»>`
- > Les dimensions d'une page web  
`<meta name=«viewport» content=«width=device-width, initial-scale=1»>`
- > Un résumé sur ce que contient le site  
`<meta name=«description» content=«...»>`
- > Quelques mots clés qui aident le référencement  
`<meta name=«keywords» content=«7rapl, 7RAPL, ...»>`

Au niveau des liens, j'ai joint à chaque page un même fichier s'appelant « style.css ». Il est rédigé en langage de programmation CSS (Cascading Style Sheets ou feuilles de style en cascade). Ce fichier va permettre, comme il se doit dans le graphisme, de donner une mise en page à toutes les pages html.



Cependant, il n'y a pas que le «style.css» qui va être lié aux fichiers html. Il y a également les favicons (c'est l'image placée à côté du titre en haut du navigateur) de toutes les dimensions possibles et pour tous les appareils possibles.

Le logo fera office de favicon. C'est à l'aide d'un site de «favicon generator» que j'ai pu avoir toutes les dimensions et compatibilités possibles.

C'est également dans le `head` que je vais mettre des fonctions javascript et jquery à l'aide d'une balise script. Une fonction sera commune à toutes les pages. C'est celle du chargement, voici le code.

	<code>&lt;script src=«src/js/jquery.js»&gt;&lt;/script&gt;</code>	Avant d'effectuer le script j'incorpore ma bibliothèque jquery.
	<code>&lt;script&gt;</code>	
Lorsque la page est prête, on lance le code.	<code>\$(window).ready(function() {</code>	
	<code>/*écran de chargement*/</code>	Commentaire dans le langage JS.
L'élément dans l'html avec l'id «loader» part avec un effet de transparence.	<code>\$(«#loader»).fadeOut(«1000»);</code>	
	<code>setTimeout(charger,1000);</code>	Ensuite, on fait appel à la fonction « charger ».
	<code>};</code>	
Ouverture de la fonction « charger ».	<code>function charger(){</code>	
	<code>\$(«#loader»).css('display','none');</code>	Changement du code CSS de l'élément correspondant à l'id «loader». Ici le code permet de le faire disparaître.
	<code>};</code>	
	<code>&lt;/script&gt;</code>	

L'id est lié à une section du body dans le code de la page html. Cette section n'est pas seulement dans le body mais est également dans la partie «main» de celui-ci.

```
<section id= «loader»>
  <img src= «src/img/logo.png» alt= «logo page chargement 7rapl» id= «logo»>
  <div id= «cercle»></div>
</section>
```

Comme vous pouvez l'observer, cette section contient une image et un cercle. Ce dernier est animé dans le fichier «style.css» afin de montrer que la page charge. Ce n'est donc pas une image mais une forme que je vais créer et animer en CSS.

Quelques lignes intéressantes dans le fichier css.

La souris se met en mode chargement.

```
#loader{
  cursor: wait;
}
```

Je donne une taille au cercle, un contour, un arrondi, une couleur,... Mais surtout une animation infinie qui s'appelle «spin» et qui envoie au code en dessous.

```
#cercle{
  width: 80px;
  height: 80px;
  border: 2px solid #f3f3f3;
  border-top: 3px solid #659B91;
  border-radius: 100%;
  margin-top: 450px;
  animation: spin 1s infinite linear;
}
```

Voici la fonction d'animation «spin» qui crée une rotation de 0° à 360°.

```
@keyframes spin {
  from{
    transform: rotate(0deg);
  }to{
    transform: rotate(360deg);
  }
}
```

## Contenu Body parties header et footer

Avant de coder ces deux extrémités du site, je vais dimensionner mon `body` dans ma feuille de style en cascade.

```
body{
  display:grid;
  grid-template-columns: 100%;
  grid-template-rows: 50px 1fr 45px;
}
```

Je vais le diviser en 3 lignes tel un tableau. Les colonnes feront toutes la largeur de la page et les lignes ont des tailles définies qui correspondent respectivement au `header`, `main` et `footer`. Le `main` a une dimension de `1fr` ce qui signifie qu'il fait l'écart entre les extrémités par rapport à la hauteur de la page.

Sur toutes les pages, je vais mettre le même header et le même footer. Ceux-ci sont composés de liens renvoyant aux autres pages principales pour le header et aux réseaux sociaux, mails pour le footer.

Comme je l'ai appris durant mon cursus, j'ai codé ces liens à l'aide de balises `<ul>` et `<li>` qui ont pour but de créer une liste avec des puces. C'est avec le css que j'ai enlevé ces éléments graphiques ( `list-style:none;` ). De plus, au lieu de les avoir les uns en dessous des autres, ils doivent être les uns à côté des autres. Les `flexbox` résolvent ce problème.

Les `flexbox` permettent, grâce à un « `display:flex;` » dans le css, de :

- > aligner des éléments de manière horizontale et verticale, au centre, à gauche, à droite, avec des espaces, ...  
`Justify-content:center;`      `align-items:center;`
- > mettre des blocs les uns en dessous des autres ou les uns à côté des autres.  
`Flex-direction:column;`      `Flex-direction:row;`
- > afficher les composants sur une seule ou plusieurs lignes.  
`Flex-wrap:nowrap;`      `Flew-wrap:wrap;`
- > mettre une partie du contenu à gauche du conteneur et une autre partie à droite de celui-ci.  
Sur le conteneur > `Display:flex;`  
Sur le contenu aligné à droite > `margin-left:auto;`

La technologie des `flexbox` est novatrice dans la façon de coder. Elle facilite grandement la mise en page. Je vais beaucoup les utiliser dans mon travail de fin d'études. Le plus gros avantage avec celles-ci, c'est que le responsive est facilité. Si vous souhaitez en savoir plus, je vous invite à aller voir sur le site de « [developer.mozilla.org](https://developer.mozilla.org) ».

En parlant de responsive, un site se doit d'être utilisable sur n'importe quel écran d'une télé, d'un ordinateur à un téléphone. Dans l'html, une seule ligne suffit comme nous l'avons vu précédemment. Ce n'est pas la même histoire dans le css.

```
@media only screen and (max-width:1550px){ ——— petit écran d'ordinateur
}

@media only screen and (max-width:1230px){ ——— tablette en paysage
}

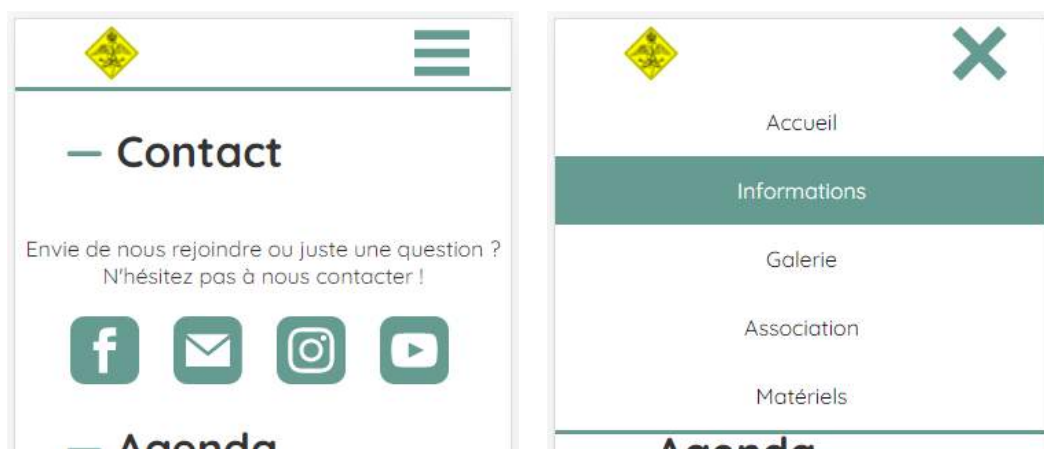
@media only screen and (max-width:900px){ ——— tablette en portrait
}

@media only screen and (max-width:560px){ ——— téléphone
}

@media only screen and (max-width:400px){ ——— petit téléphone
}
```

Au-dessus de ces codes, ce sera la dimension supérieure à 1550px, donc pour de plus grands écrans. A l'intérieur des accolades, se trouvera les éléments dont, par exemple, la dimension change.

Sur téléphone, la navigation du header ne doit pas être la même que sur ordinateur. Je transforme mon menu horizontal en menu vertical qui se déroule lors du clic sur mon icône « burger » grâce à une fonction JQuery.



L'icône est remplacée par une croix. Vous pouvez également voir sur quelle page vous êtes à partir de cette navigation et ce, que ce soit sur téléphone ou autre.

Pour afficher, changer l'orientation de la navigation, je change les `flexbox`. Une fonction jquery doit être créée afin que le menu s'affiche lorsqu'on clique sur l'icône du burger.

	<code>\$(«#croix»).css('display','none');</code>	La croix est cachée dès le début
Animation sous forme de mouvement de glissement vers le bas.	<code>\$(«#burger»).click(function(){</code> <code>\$(«nav»).slideDown();</code> <code>\$(«#croix»).fadeIn();</code> <code>\$(«#burger»).css('display','none');</code> <code>});</code>	Lors du clic sur l'icône burger, on lance l'animation de l'apparition de la navigation. On change d'icône.
Animation sous forme de mouvement de glissement vers le haut.	<code>\$(«#croix»).click(function(){</code> <code>\$(«nav»).slideUp();</code> <code>\$(«#croix»).css('display','none');</code> <code>\$(«#burger»).fadeIn();</code> <code>});</code>	Lors du clic sur l'icône croix, on démarre l'animation de la navigation. Et on change d'icône.  Animation qui joue sur la transparence de l'élément. Ici la valeur passe de 0 à 100% sur la durée par défaut de la fonction.

Etant donné que le contenu du `header` et du `footer` seront des liens, il faut qu'on le remarque facilement et de manière visuelle. Pour cela, je vais changer la couleur au survol et lors du clic sur l'élément. Cela va se coder en CSS dans mon fichier « `style.css` ».

Dans le `header`, vu que de base le fond est blanc, je le passe en vert au survol et en bleu au clic.

```
.lien:hover, .lien:focus{
  background-color: #659B91;
  color:#fff;
}
.lien:active{
  background-color: #BEE0E4;
}
```

Dans le `footer`, vu que de base le fond est vert, je le passe en bleu au survol et en blanc au clic.

```
footer a:hover, footer a:focus{
  color: #BEE0E4;
  cursor:pointer;
}
footer a:active{
  color: #fff;
}
```



Ces deux extrémités sont désormais faites.  
 Passons maintenant à l'essence même du site web,  
 l'intérieur du main. Je vais procéder par page comme,  
 pour les layouts, en vous expliquant les particularités  
 au niveau du code de chacune d'entre elles.

## Contenu Body partie main

Chaque page a une certaine mise en page et un certain contenu. C'est dans la partie main que se concentrent ces éléments. Forcément du css va s'ajouter, et parfois du javascript, du jquery et même du php.

Mais avant de voir les particularités de chaque page, des éléments communs subsistent. Notamment une flèche qui permet de remonter en haut de la page lorsque celle-ci est longue.

Au niveau de l'html, le code tient sur trois lignes.  
 La balise «a» envoie au clic l'utilisateur à  
 l'élément possédant l'id indiqué. (l'exemple est  
 repris de la page association)

```
<a href="#histoire" id="flechetop">

</a>
```

Dans le script du head, après le  
 «\$(window).ready». J'ajoute une condition  
 au scroll afin que la flèche ne s'affiche pas  
 dès le départ.

Lorsque le visiteur est descendu d'au  
 moins 500 pixels, j'affiche la flèche.

```
$(window).scroll(function(){
  if(window.scrollY<500){
    $('#flechetop').css('display','none');
  }
  else{
    $('#flechetop').css('display','block');
  }
});
```

Le second code est commun à toutes les pages possédant des images cliquables. C'est une animation css méconnue que j'ai souhaitée utiliser. Elle se nomme «filter» et a besoin d'un second code pour être compatible sur tous les navigateurs.

```
#contenu img:hover,#contenu img:focus{
  filter: grayscale(100%);
  -webkit-filter: grayscale(100%);
  cursor: pointer;
}
#contenu img:active{
  filter: hue-rotate(50deg);
  -webkit-filter: hue-rotate(50deg);
  cursor: pointer;
}
```

Au survol et au focus, l'image passe en niveaux de gris.

Au clic, les couleurs de l'image vont effectuer une rotation (selon une valeur en degrés) sur la roue colorimétrique.

Désormais nous sommes prêts à voir les codes page après page.

Page « Accueil »

Commençons par le début, la page principale. Je vais la diviser en 2 parties, une pour le texte et une pour le slider. La 1ère va à gauche et la seconde à droite. Je les regroupe dans une balise «div» avec un id contenu. Cela va permettre de cibler cet élément et de le mettre en page à l'aide de flexbox.

Dans cette page, la particularité se trouve sur le slider. Je vais donc vous développer les codes html, css et jquery de cette partie.

Le code html est assez basique, ce ne sont que des div qui sont traités en css pour la mise en forme.

Comme vous pouvez voir il y a deux flèches qui auront comme mission le changement d'image. Il y a 3 slides dont les fonds vont être des images gérées par le css. Et sur la fin du code, il y a 3 cercles qui auront le même but que les flèches.

```
<div id="container">
  <div id="fleche_gauche" class="fleche"></div>
  <div id="slider">
    <div class="slide" id="slide1">
    </div>
    <div class="slide" id="slide2">
    </div>
    <div class="slide" id="slide3">
    </div>
  </div>
  <div id="fleche_droite" class="fleche"></div>
  <div id="dessousimg">
    <div id="rectcircle">
      <div class="circle" id="c1"></div>
      <div class="circle" id="c2"></div>
      <div class="circle" id="c3"></div>
    </div>
  </div>
```

css de la slide 3

```
#container #slide3{
    background-image: url(../img/slider3.webp);
}
```

Format d'image en Webp au lieu de jpg ou png car c'est le format standard sur le web et il est le plus léger.

css de la flèche gauche

```
#fleche_gauche{
    border-width: 30px 40px 30px 0;
    border-color : transparent #659B91 transparent transparent;
    left:10px;
}
```

Il est pratiquement identique pour la flèche droite, seules les dimensions du border changent > 30px 0 30px 40px et dans le border-color, la couleur passe de la deuxième position à la dernière. De plus, au lieu de left ce sera right.

css des cercles

```
.circle{
    width: 20px;
    height: 20px;
    border:3px solid white;
    border-radius: 9px;
    margin:10px;
    cursor:pointer;
}
```

Un des 3 cercles aura l'intérieur coloré afin de montrer visuellement sur quelle slide est le visiteur.

Le code javascript est plus volumineux et complexe. Je vais débiter avant le « `$(window).ready` », déjà présent, par la suppression des deux flèches du slider.

```
$(«#fleche_gauche»).css('display','none');
$(«#fleche_droite»).css('display','none');
```

Ensuite, dans le « `$(window).ready` », je vais ajouter une variable qui va reprendre la largeur de la page. Cela va me permettre de savoir sur quel appareil je suis et, du coup, ne plus afficher les flèches lorsqu'on est sur téléphone. Si je suis sur une résolution d'écran mesurant plus que 560px, je vais afficher les flèches au survol du slider.

```
var larg = (document.body.clientWidth);

if(larg<560){
    $(«fleche»).css('display','none');
}
else{
    $(«#slider»).mouseenter(function(){ — La souris entre dans la zone du slider
        $(«fleche»).css('display','block');
        couleurslide(slide); — Appel vers une fonction que j'explique plus bas
    });
    $(«#slider»).mouseleave(function(){ — La souris sort de la zone du slider
        $(«fleche»).css('display','none');
        couleurslide(slide);
    });
    $(«.fleche»).mouseenter(function(){ — La souris est sur une des deux flèches
        $(«fleche»).css('display','block');
        couleurslide(slide);
    });
    $(«.fleche»).mouseleave(function(){ — La souris n'est plus sur une des deux flèches
        $(«fleche»).css('display','none');
        couleurslide(slide);
    });
}
```

Pour changer de slide, il faut deux fonctions «click» qui se déclenchent lorsque le visiteur clique sur une flèche. Afin de savoir sur quelle slide nous sommes, je vais initialiser à 0 une variable «slide» avant le « `$(window).ready` ». Celle-ci variera en fonction des cliques.

Si on clique sur la flèche de droite, la valeur du «slide» augmente car on va en chercher la prochaine. Une animation de translation vers la gauche est effectuée durant 1000 secondes.

A l'inverse, si on clique sur la flèche de gauche, la valeur de «slide» diminue car on revient sur la précédente.

```
$(«#fleche_droite»).click(function animationd(){
    slide++;
    $(«#fleche_gauche»).css('display','block');
    couleurslide(slide);
    $(«#slider»).animate({
        left:(-100*slide)+'%',
    }, 1000);
});
$(«#fleche_gauche»).click(function animationg(){
    slide--;
    couleurslide(slide);
    $(«#slider»).animate({
        left:(-100*slide)+'%',
    }, 1000);
});
```

```

$(«#c1»).click(function animationd(){
    slide=0;
    couleurslide(slide);
    $(«#slider»).animate({
        left:(-100*slide)+'%',
    }, 1000);
});
$(«#c2»).click(function animationd(){
    slide=1;
    couleurslide(slide);
    $(«#slider»).animate({
        left:(-100*slide)+'%',
    }, 1000);
});
$(«#c3»).click(function animationd(){
    slide=2;
    couleurslide(slide);
    $(«#slider»).animate({
        left:(-100*slide)+'%',
    }, 1000);
});

```

La deuxième manière de changer de slide, est de cliquer sur les cercles situés en dessous du slider.

Si on clique sur le 1er cercle, la valeur de «slide» vaut 1, si c'est sur le second sa valeur vaut 2, et sur le 3ème elle vaudra 3.

L'animation est la même que pour les flèches.

Contrairement aux flèches, cette fonctionnalité reste sur le téléphone. Par ailleurs, ces flèches sont remplacées par une fonction «swipe». Elle correspond au fait de glisser son doigt de gauche à droite pour aller à la slide précédente et de droite à gauche pour aller à la slide suivante.

On met la zone délimitée du slider dans la variable «even».

Appel de la fonction «swipe» avec comme argument les positions.

```

const even = document.querySelector("#container");
even.addEventListener("touchstart",(e)=>{
    startx=e.touches[0].clientX;
})
even.addEventListener("touchend",(e)=>{
    endx=e.changedTouches[0].clientX;
    swipe(startx,endx);
})

```

Détection d'un doigt qui entre dans la zone.

Détection d'un doigt qui sort dans la zone.

Auparavant, j'initialise en dessous de la variable «slide», deux autres variables nommées «startx» et «endx» qui ont comme valeur par défaut 0. «startx» récolte la position du doigt à son entrée dans la zone et «endx» la position du doigt à sa sortie.

Je vais ajouter fonction «swipe» en dessous de la fermeture du « \$(window).ready ». Cette fermeture est représentée comme suit > « ) ».

Dans cette fonction, la variable «debut» correspond à «startx» et la variable «fin» est liée à «endx».

Je vais mettre deux conditions :

- > une pour savoir si le doigt a glissé de gauche à droite.
- > l'autre pour savoir si le doigt a glissé de droite à gauche.

Dans ces dernières, nous retrouvons le même code que pour les flèches.

```

function swipe(debut,fin){
    if(debut<fin){
        slide--;
        couleurslide(slide);
        $(«#slider»).animate({
            left:(-100*slide)+'%',
        }, 1000);
    }
    else if (debut>fin){
        slide++;
        couleurslide(slide);
        $(«#slider»).animate({
            left:(-100*slide)+'%',
        }, 1000);
    }
};

```

Si vous êtes attentifs, vous avez probablement remarqué que peu importe les manières, elles renvoient toutes vers une seule et même fonction «couleurslide» avec en attribut le chiffre du slide.

Cette fonction, je l'ai placée en dessous de «swiper». Elle a pour but de changer la couleur des cercles présents sous le slider. Comme cela on peut voir visuellement sur quel numéro de slide nous nous trouvons.

De plus, si j'arrive à la dernière slide, et que je clique sur la flèche de droite ou si je swipe, je reviens sur la première slide.

Enfin, sur la première slide, il n'y a pas besoin d'avoir la flèche de gauche. De ce fait je l'enlève lorsqu'on se trouve sur la slide 0.

```
function couleurslide(slidenb){
  if(slidenb==0){
    $(«#c1»).css('background-color','white');
    $(«#c2»).css('background-color','#659B91');
    $(«#c3»).css('background-color','#659B91');
    $(«#fleche_gauche»).css('display','none');
  }
  else if(slidenb==1){
    $(«#c1»).css('background-color','#659B91');
    $(«#c2»).css('background-color','white');
    $(«#c3»).css('background-color','#659B91');
  }
  else if(slidenb==2){
    $(«#c1»).css('background-color','#659B91');
    $(«#c2»).css('background-color','#659B91');
    $(«#c3»).css('background-color','white');
  }
  else{
    slide=0;
    couleurslide(slide);
  }
};
```

## Page « Informations »

Je crée 3 sections > une «contact», une «agenda» et une «déplacement». Je vais vous montrer les codes des deux dernières.

Ci-dessous, vous pouvez observer deux titres avec un «&mdash;» à l'intérieur. Cela correspond à un tiret dans le langage html.

Avec ces titres, 2 iframes sont présents. L'un permet d'afficher l'agenda google de l'asbl et l'autre une carte géographique de google montrant les différents bivouacs où le 7rapl s'est rendu.

```
<h2><span class=«tiret»>&mdash;</span> Agenda </h2>
<section id= «agenda»>
  <iframe src=«https://calendar.google.com/calendar/
embed?src=ke9mg217qk8i91t5jjkt4mtco4%40group.calendar.google.com&ctz=Europe%2FBrussels»
style= «border: 0» width=«800» height=«600»>
  </iframe>
</section>

<h2><span class= «tiret»>&mdash;</span> Déplacements </h2>
<section id= «deplacement»>
  <iframe src= «https://www.google.com/maps/d/
embed?mid=1BwkmK7TZhwRVakMykalvVx9mJdPkj131»>
  </iframe>
</section>
```

Contrairement aux deux précédentes pages, celle-ci ne va pas avoir une extension `.html` mais `.php` car il va y avoir du code php à l'intérieur. Lorsque du php est présent dans une page, le programmeur doit utiliser le logiciel avec un environnement comprenant différents serveurs et notamment un `phpMyAdmin`. Il en existe plusieurs, le plus connu est `WAMP` mais personnellement j'ai utilisé `Laragon`.

Cette page n'est pourtant pas très différente des autres hormis le code ci-dessous. Il rend cliquable chaque vignette. Celle-ci envoie l'utilisateur vers la page «album» avec comme argument le chiffre de la vignette qui a été cliquée (voir encadré noir dans le code).

```
<a href=«album.php?album=1»>
  <div class=«album» id=«A1»>
    <img src=«src/img/galerie/A1.webp» alt=«Photo de l'évènement 7rapl en 2019 à Plessis-Bourré»>
    <p class=«text»>2019 - Plessis-Bourré</p>
  </div>
</a>
```

Une fois sur la page album, un code PHP se situe en haut de la page, avant même la première ligne d'html. Il vient récupérer le chiffre et vérifie s'il y en a bien un et s'il correspond à un album. Si c'est le cas, je vais stocker le nombre dans une variable «\$recup». Dans le cas contraire, j'initialise le chiffre à 1.

```
<?php
if (isset($_GET['album'])){
    $recup=(int) $_GET['album'];
    if($recup<1||$recup>6){
        $recup=1;
    }
}
else{
    $recup=1;
}
```

Afin d'afficher l'année et le lieu de l'évènement de l'album sélectionné, je code un «switch» qui donnera cela en fonction du numéro récupéré.

```
switch($recup){
    case 1: $titre=«2019 - Plessis-Bourré»;
        break;
    case 2: $titre=«2018 - Cottenchy»;
        break;
    case 3: $titre=«2017 - Gravelines»;
        break;
    case 4: $titre=«2016 - Seclin»;
        break;
    case 5 : $titre=«2016 - Plessis-Bourré»;
        break;
    case 6 : $titre=«2016 - Wavre»;
        break;
}
?>
```

Je vais l'utiliser directement dans le «title» du code html. Voici ce que ça donne :

```
<?php echo '<title>7rapl Album '$titre,'</title>'>
```

Plus bas dans le code, dans le main, je vais chercher toutes les images se trouvant dans le dossier ayant le même numéro que la variable «\$recup».

```
<?php
$rep=opendir('src/img/galerie/A'.$recup) or die («erreur ouverture du
dossier» );
while ($file = readdir($rep)) {
    if (($file != '..') && ($file != '.') && ($file != '')) {
        echo '<div class=«image» id=«',$file,'»>
            <img src=«src/img/galerie/A'.$recup.'/'.$file,'»
            onclick=«fonction clic(src)»
            alt=«image de la galerie 7RAPL dans l\'album \''.$title,'»>
        </div>';
    }
}
closedir($rep);
?>
```

opendir ouvre le dossier avec le nom donné. Si aucun dossier n'a été trouvé, die entre en action et affiche sa phrase.

Partie qui affiche l'image à l'aide d'un chemin, la balise contient également un indicateur d'évènement JS, «onclick» et un alt qui a la même mission que le die.

L'évènement Javascript fait appel à la fonction «fonction clic(src)» qui a pour attribut la source de l'image cliquée. Cette fonction a pour but d'agrandir l'image, de la voir entièrement. Comme pour les flèches sur la page index, je limite cette fonction à une largeur de minimum 900px. A partir de cette dimension, plus besoin de cliquer sur les images, elles s'affichent en grand directement.

Afin de l'agrandir sur la même page, je vais supprimer tous les éléments et ajouter, à l'aide d'un «append» (voir rectangle noir), une balise «img» dans le code html dans le div qui possède comme id «bigimage». Cette image a pour chemin la source reçue en attribut.



```

<script>
Javascript — function fonctionclic(source) {
                /*largeur de la fenêtre du navigateur*/
                var larg = (document.body.clientWidth);
                /*agrandissement de l'image bloqué*/
                if(larg>900){
                    $(«#contenu»).css('display','none');
                    $(«header»).css('display','none');
                    $(«footer»).css('display','none');
                    $(«#titrealbum»).css('display','none');
                    $(«#grand»).css('display','block');
                    $(«body»).css('display','block');
                    $(«main»).css('height','100%');
                    $(«#bigimage»).append('<img src=«'+source+'» alt=«image grande»>');
                }
            }
            </script>
            <section id=«grand»>
            <img src=«src/img/icon-croix.svg» alt=«retour» id=«croixbig»>
            <div id=«bigimage»></div>
            </section>
            Html —

```

Ce code est placé après la fermeture de la section «contenu».

Plusieurs particularités sont regroupées sur cette page. Tout d'abord, la partie histoire est sous forme de slider. Celui-ci a la même logique que celui de la page d'accueil. Cependant, ce n'est pas seulement une image. Il y a certes une photo mais également du texte et une certaine mise en page css. Au lieu de faire une translation vers la gauche, comme sur la première page, je vais effacer la slide pour afficher la suivante ou la précédente grâce à des conditions.

```

else if(slidenb==2){
    $(«#slide2»).css('display','none');
    $(«#slide3»).css('display','flex');
    $(«#slide4»).css('display','none');
    $(«#fleche_droite»).css('display','block');
}

```

Ce code correspond à la 3ème slide. On peut voir qu'on efface la slide 2, on affiche la slide 3, on efface la slide 4 et on affiche la flèche droite. Pourquoi tout cela ? Il faut tenir compte du fait que le visiteur peut venir de la slide 2 et de la slide 4 ! De plus sur la slide 4 la flèche de droite n'est pas affichée car il n'y a pas de 5ème slide.

La partie membre possède un affichage totalement dynamique. En premier lieu, le code PHP va afficher les photos des membres ainsi que les noms, prénoms et grades.

```

<section id= «membre»>
Initialisation      <?php
variables à 0.      $i=0;
                    $g=0;
Boucles de 10.      while($i>=0&&$i<10){
Tableau PHP des noms.  $nom = array(«Yves VANDEKERCKHOVE», «Loïc MICHAUX»,«Philippe DERYCKE»,
                        «Philippe POLLART», «Julien LIMBOURG», «Thierry NORMAND»,«Yves VANDEWIELE»,
                        «Quentin DEPAUW», «Yannis BOUCKAERT», «Sylvie GOLARD»);
Tableau PHP des grades.  $grade = array(
                        0=>«Lieutenant / Sergent-Major»,
                        1=>«Sergent»,«Caporal-Fourrier»,
                        2=>«Philippe POLLART»,
                        3 =>«Canonnière 1ère classe»,
                        5 =>«Canonnière 2ème classe»,
                        8=>«Tambour»,
                        9=>«Vivandière»);
                        if($i==5) echo '<div id=«trois»>';
                        if($i==3||$i==8) echo '<div class=«deux»>';
                        if($i==6||$i==7) $g=5;
                        if($i==4) $g=3;
Affichage des éléments.  echo '<div class=«profil» id=«', $i, '»><img src=«src/img/M', $i+1, 'C.webp» alt=«Photo
                        ', $grade[$g], ' - ', $nom[$i], '» id=«', $i+1, '»>';
                        echo '<p><b>', $grade[$g], '</b><br />', $nom[$i], '</p></div>';
Fermetures afin d'arrêter la mise en page au format ligne en css.  if($i==4||$i==7||$i==9){
                        echo '</div>';
                        $g=$i;
                        }
Incrémentation des variables.  $i++;
                        $g++;
                        }
                        echo '</section>';
?>

```

En dessous du code précédent, je vais rédiger un bout de code alliant JQuery et Javascript. Lorsqu'on clique sur un membre, tout disparaît et son profil est affiché. Le profil est composé de la même photo mais agrandie, de son grade, son sobriquet, son année d'incorporation et un petit texte.

Avant de vous montrer le script, voici l'html qui sera placé après le script.

Image croix qui permettra de rétablir l'affichage précédent.	<code>&lt;section id=«contenuprofil»&gt;</code>	
	<code>    &lt;div class=«ligne1 haut»&gt;&lt;/div&gt;</code>	
	<code>    &lt;div class=«ligne2 haut»&gt;&lt;/div&gt;</code>	
	<code>    &lt;img src=«src/img/icon-croix.svg» alt=«retour» id=«croixbig»&gt;</code>	
	<code>    &lt;div id=«phototext»&gt;&lt;div id=«presentation»&gt;&lt;/div&gt;&lt;/div&gt;</code>	Contiendra tout sauf le texte
Contiendra le texte descriptif.	<code>    &lt;div id=«descrip»&gt;&lt;/div&gt;</code>	
	<code>    &lt;div class=«ligne2 bas»&gt;&lt;/div&gt;</code>	
	<code>    &lt;div class=«ligne1 bas»&gt;&lt;/div&gt;</code>	
	<code>&lt;/section&gt;</code>	

A présent voici le début du script, comme pour la page album, lorsqu'on clique sur l'élément de la classe «profil», on enlève le contenu de toute la page et on affiche la section «contenuprofil».

```
<script>
$(«profil»).click(function(){
$(«header»).css('display','none');
$(«#histoire»).css('display','none');
$(«#membre»).css('display','none');
$(«#ancien»).css('display','none');
$(«footer»).css('display','none');
$(«h2»).css('display','none');
$(«body»).css('display','block');
$(«main»).css('height','100%');
$(«#contenuprofil»).css('display','flex');
```

Après je vais stocker l'id du profil cliqué dans la variable «recup» et «recupimg» qui lui, va être incrémenté de 1.

```
var recup = $(this).attr(«id»);
var recupimg = $(this).attr(«id»);
recupimg++;
```

Dans la même idée que le php, je vais mettre les données dans des tableaux JS comme par exemple >

```
var nom = [«Yves VANDEKERCKHOVE», «Loïc MICHAUX»,«Philippe DERYCKE», «Philippe POLLART», «Julien LIMBOURG», «Thierry NORMAND»,«Yves VANDEWIELE», «Quentin DEPAUW», «Yannis BOUCKAERT», «Sylvie GOLARD»];
```

Pour finir, j'ajoute mes différents éléments dans l'html par des méthodes javascript.

Nous retrouvons `append` comme pour la page album mais il y a également `prepend`. Les deux se ressemblent.

La différence est que `prepend`, au lieu d'ajouter à la suite, positionne l'élément en premier, juste après le parent.

J'ai également placé une condition sur le texte descriptif car certains membres n'en ont pas.

```
$(«#phototext»).prepend('<img src=»src/img/M'+recupimg+'.webp» alt=»image représentant le membre de l\'asbl 7rapl»>');
```

```
$(«#presentation»).append('<h1>'+nom[recup]+'</h1>');
```

```
$(«#presentation»).append('<p>Grade > '+grade[recup]+'</p>');
```

```
$(«#presentation»).append('<p>Incorporation > '+date[recup]+'</p>');
```

```
$(«#presentation»).append('<p>Sobriquet > '+surnom[recup]+'</p>');
```

```
if(texte[recup]!=""r){
    $(«#descrip»).append('<p id=»text»>'+texte[recup]+'</p>');
}
```

Cette page n'a en soi rien de novatrice, elle contient 3 blocs qui renvoient vers diverses sous-pages. Elles possèdent les mêmes `header` et `footer` que les autres pages. Ce qui est plus intrigant est ce qu'elles contiennent.

```
<form method=«post» action=«uniform.php»>
  <select name=«uniforme»>
    <option value=«0»>Choisir un uniforme</option>
    <option value=«1»>Lieutenant</option>
    <option value=«2»>Sergent-major</option>
    <option value=«3»>Sergent</option>
    <option value=«4»>Caporal-fourrier</option>
    <option value=«5»>Caporal-Armurier</option>
    <option value=«6»>Caporal</option>
    <option value=«7»>Canonier 1ère Classe</option>
    <option value=«8»>Canonier</option>
    <option value=«9»>Tambour</option>
  </select>
  <input type=«submit» name=«submit» value=«sélectionner»/>
</form>
```

La première sous-page regroupe les uniformes portés par les membres de l'asbl. Je crée un formulaire de méthode `«post»` car c'est plus sécurisé. A l'intérieur, j'y glisse un `«select»` qui va permettre de changer facilement les grades et qui est compréhensible par tout le monde.

Afin d'identifier quel uniforme est sélectionné, j'incorpore une valeur chiffrée à chaque `«option»`.

Vérifie si le formulaire a été envoyé.

```
<?php
```

```
if(isset($_POST['uniforme'])){
    $select = $_POST['uniforme'];
```

«\$select» reçoit la valeur envoyée par le formulaire.

En fonction du nombre reçu, je vais afficher l'uniforme correspondant au grade sélectionné.

```
switch ($select) {
```

```
case 0:
```

```
    echo '<p>Veuillez choisir un uniforme</p>';
```

```
    break;
```

```
case 1:
```

```
    echo '<div id= «img1» class=«imageunif»></div>';
```

```
    echo '<div class=«droite»>';
```

```
    echo '<div class=«cadre»>';
```

```
    echo '<h1>Lieutenant</h1>';
```

```
    echo '<div class=«lignevert»></div>';
```

```
    echo '<div class=«lignehoriz»></div>';
```

```
    echo '</div><ol>';
```

```
    echo '<li>Rang > Officier</li>';
```

```
    echo '<li>Appellation > Mon lieutenant</li>';
```

```
    echo '<li>Fonction > Etat major</li>';
```

```
    echo '<li class=«descrip»>Description > L'officier
```

```
    est responsable de l'ensemble des hommes mis sous son
    commandement. Il reçoit les comptes-rendus de ses subalternes
    et décide de la marche générale des opérations. C'est le regard
    « externe » de la compagnie.</li></ol></div>';
```

```
    break;
```

Par défaut, le formulaire demande de choisir un uniforme

Si «\$select» vaut 1, c'est le grade Lieutenant. J'affiche tous les éléments correspondants.

Les autres «case» sont identiques, juste les données changent. J'aurais pu faire la même méthode que le code d'affichage des membres sur la page association. Je souhaitais faire différemment pour montrer qu'il y a une multitude de possibilités pour afficher des éléments.

La page uniforme n'est pas complexe en soi, la page news non plus d'ailleurs. Cependant, la page équipements m'a donnée du «fil à retordre».

Cette page comprend les noms des armes, des caractéristiques, une description mais surtout des boutons qui renvoient vers de la 3D interactive. C'est avec la technologie «THREE.JS» que j'ai pu rendre les armes interactives directement sur un site web.

Avant de voir à quoi ressemble cette bibliothèque javascript qui permet de créer un environnement en 3 Dimensions, voici le code du sabre sur la page équipements. Les autres armes ont le même code avec des données différentes.

```
<h2>Sabre Briquet</h2>
<div class=«lignehoriz»></div>
<div class=«lignevert»></div>
<article class=«imagetext»>
  <div id=«imgsabre» class=«img»><img src=«src/img/sabre.webp» alt=«photo du sabre
  utilisé par l'asbl 7rapl»></div>
  <div class=«texteG»>
    <p>Le briquet est un sabre à lame courbe unie, à dos plat munie d'un seul tranchant. Il
    fut imposé à l'artillerie en 1803 en remplacement de son glaive à tête d'aigle datant de 1775.
    Sa poignée est reconnaissable par ses 28 godrons en enfilade. Il était systématiquement
    accompagné d'une dragonne dont le modèle était raccord avec le grade du porteur.</p>
    <h5>Propriétés</h5>
    <ol>
      <li>Longueur de lame > 595 mm</li>
      <li>Masse > </li>
      <li>Matériaux > Acier et Laiton</li>
    </ol>
    <div class=«bout»>
      <div class=«bouton» id=«photoS» onclick=«changesources('1')»>Photo</div>
      <div class=«bouton» id=«renduS» onclick=«changesources('2')»>Rendu 3D</div>
      <a href=«src/3d/sabre/sabre.html» class=«bouton»>3D Interactive</a>
    </div>
  </div>
</article>
```

Ici la seule chose un peu spéciale est que je change la source de l'image (premier encadré noir) lorsqu'on clique sur un des 2 boutons (deuxième encadré noir). C'est avec une méthode javascript «onclick» que je vais changer ce chemin.

```
<script>
function changesources(chiffs){
  if(chiffs==1){
    $('#imgsabre img').replaceWith('<img
src=«src/img/sabre.webp» alt=«photo du sabre utilisé
par l'asbl 7rapl»>');
  }
  else if(chiffs==2){
    $('#imgsabre img').replaceWith('<img
src=«src/img/sabre3d.webp» alt=«le sabre en 3D
utilisé par l'asbl 7rapl» class=«larg»>');
  }
}
</script>
```

Si on clique sur le 1er bouton, la valeur de «chiffs» vaut 1 donc la source est la photo du sabre.

Si on clique sur le second bouton, la valeur vaudra 2 ce qui correspond à l'image du rendu sorti de Solidworks.



L'encadré vert dans le code html précédant envoie le visiteur vers une page «sabre.html». Celle-ci est dans un dossier 3d et un dossier sabre comme on peut le voir dans le chemin : «src/3d/sabre/sabre.html». Ce n'était pas possible de mettre cette page dans le même dossier que les autres et voici pourquoi.

Il faut diverses bibliothèques javascript :

> En premier lieu, celle dont je vous parle depuis le début, [three.min.js](#).

> Ensuite, pour insérer mon objet sorti en format .GLB, il me faut la librairie [GLTFLoader.js](#).

> Enfin pour avoir de l'interactivité entre la souris et l'objet, c'est le [OrbitControls.js](#) qui doit être présent.

Bien évidemment il est primordial de faire un fichier [script.js](#) lié à l'html.

```
<script src=«three.min.js»></script>
<script src=«OrbitControls.js»></script>
<script src=«GLTFLoader.js»></script>
<script src=«script.js»></script>
```

Avant d'entrer dans le script, regardons l'html. Comme je vous l'ai expliqué précédemment, j'ajoute les bibliothèques javascript dont j'ai besoin une par une.

```
<a href= «../..../équipement.html»><img src=«../..../img/
icon-croix.svg» alt=«retour» id=«croix»></a>
```

Je mets également une image de croix qui permet de retourner à la page équipement.

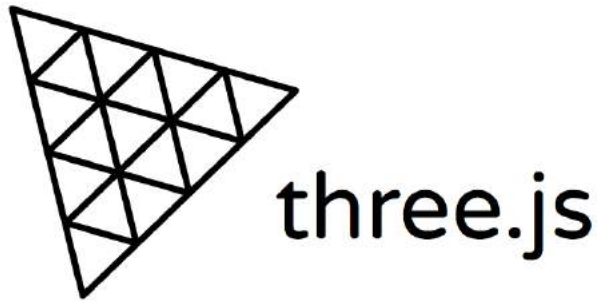
```
<h1 id=«text»>Bougez l'arme à l'aide de la souris ou de
votre doigt</h1>
```

Afin que tout le monde puisse comprendre comment voyager dans l'environnement 3D, je mets un texte l'expliquant. Pour éviter qu'il reste, je vais coder un petit script.

J'insère une autre bibliothèque, qui est présente sur tous les fichiers jusqu'à présent, la «jquery.js».

Je code deux fonctions qui ont pour but d'enlever le texte lorsqu'on clique ou touche l'écran.

```
<script src=«../..../js/jquery.js»></script>
<script>
document.body.addEventListener('touchmove', process_touchstart, true);
$( document.body ).click(function() {
    $(«#text»).css('display','none');
});
function process_touchstart(){
    $(«#text»).css('display','none');
}
</script>
```



Sur le site de [threejs.org](https://threejs.org), Il y a des exemples et beaucoup de documentations.

Il est donc facile de comprendre les codes lorsque l'on a un minimum de base.

Voici les codes de mon objet en 3 dimensions.

D'abord, il faut créer une scène, un environnement avec une couleur d'arrière plan qui ici sera verte.

```
var scene = new THREE.Scene();
scene.background = new THREE.Color(0x659B91);
```

Ensuite, j'utilise la même fonction que pour la page index pour avoir la largeur de la fenêtre.

```
var larg = (document.body.clientWidth);
```

Forcément comme dans toute scène, il faut une caméra. Celle-ci aura comme attributs, 4 pour le champ de vision, la largeur reprise précédemment divisée par la hauteur afin d'avoir un rapport de grandeur et, les deux derniers attributs, concernent la délimitation du proche et de l'éloigné.

```
var camera = new THREE.PerspectiveCamera(4, larg / window.innerHeight*1.1, 1, 50 );
```

```
var chfflight;
if(larg>1200){
    camera.position.set(-7.5,0,1);
    chfflight=1;
}
else if(larg>900&&larg<1200){
    camera.position.set(-10.5,0,1);
    chfflight=3;
}
else if(larg>560&&larg<900){
    camera.position.set(-17.5,0,1);
    chfflight=10;
}
else if(larg<=560){
    camera.position.set(-27.5,0,1);
    chfflight=20;
}
```

Comme pour toute page web, elle se doit d'être responsive. Afin de pouvoir voir l'arme entière sur n'importe quel écran, je vais faire varier la position x, y et z de la caméra.

Au plus la largeur de l'écran sera petite, au plus la caméra sera éloignée.

La variable «chfflight» va être utile pour le positionnement des lights que nous verrons ultérieurement. Encore une fois, au plus la largeur de la fenêtre sera petite, au plus les lumières seront éloignées.

```
var loader = new THREE.GLTFLoader();

loader.load( 'scene.glb', function ( gltf ) {
    var mesh = gltf.scene;
    mesh.position.set(0,-0.07,0);
    scene.add( gltf.scene );
}, undefined, function ( error ) {
    console.onerror( error );
});
```

Maintenant, je vais ajouter mon objet 3D à ma scène. Je décris au navigateur qu'il est en format binaire. Il est codé, ici c'est en `gltf` ce qui correspond au `JSON`.

La variable «`mesh`» va positionner l'objet dans la scène pour ensuite pouvoir l'ajouter dans celle-ci.

Si l'objet n'est pas trouvé, on va afficher une erreur dans la console.

Cependant, l'objet n'apparaîtra pas car il manque le rendu. Je crée donc une variable «`renderer`» qui aura comme rôle de contenir le chemin du script dont nous avons besoin. Je donne les dimensions de la fenêtre au rendu, je décris la texture de mon objet et je lui indique que c'est l'enfant de `body`.

```
var renderer = new THREE.WebGLRenderer();
renderer.setSize( window.innerWidth, window.innerHeight );
renderer.toneMapping=THREE.ReinhardToneMapping;
renderer.toneMappingExposure = 60.3;
document.body.appendChild( renderer.domElement );
```

```
animate();

function animate() {
    requestAnimationFrame( animate );
    renderer.render( scene, camera );
}
```

Dans le bas de mon script je crée un appel de fonction. Celle-ci va afficher la scène et la caméra.

En parlant d'animation, je vais coder le fait de bouger l'objet.

```
var controls = new THREE.OrbitControls(camera,renderer.domElement);
```

Dans la réalité du code, ce ne sera pas l'objet qui va bouger mais bien la caméra. On penserait que le code va être long et laborieux mais non une seule et unique ligne suffit.

Grâce à la bibliothèque Javascript, il n'est pas difficile de faire cette manipulation, il faut mettre en argument la caméra et le rendu. Si la caméra est correctement positionnée selon les axes X, Y et Z par rapport à l'objet, l'animation se fera sans problème.

Il y a la possibilité de faire bouger l'objet de quelques degrés par seconde ou même de limiter le zoom. Je n'ai pas souhaité le faire car mon but était de donner la totale liberté au visiteur.

L'objet va apparaître noir, sans couleur ni matière. Il manque des lumières !

Cette partie du code est très complexe dans le sens où il faut absolument trouver la position exacte des lumières pour éclairer tous les côtés de l'objet et ce même lorsqu'il est manipulé.

Dans un premier temps je vais changer la position en Z de la caméra afin d'être sûre que l'objet ne soit pas à la même place qu'elle.

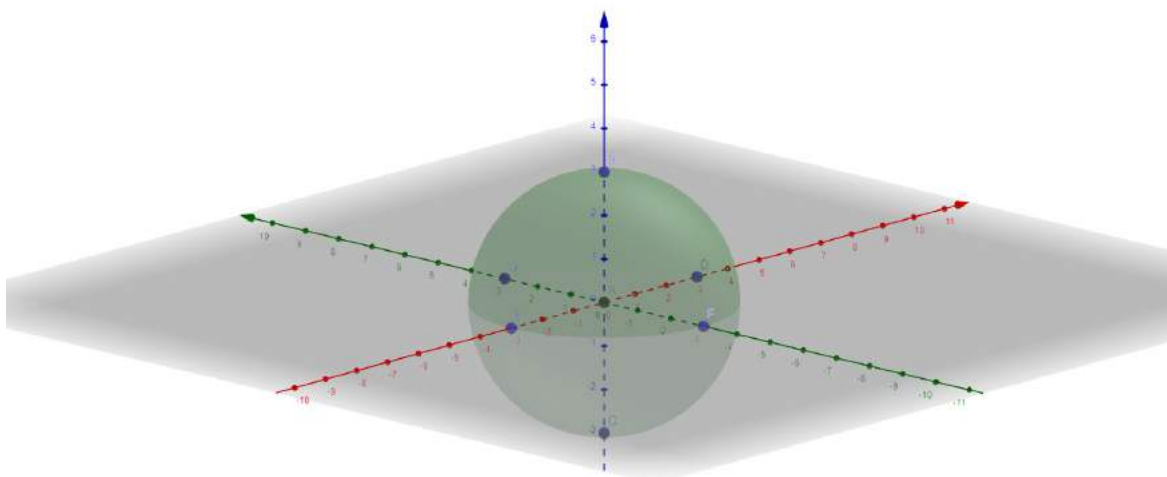
```
camera.position.z = 1;
```

```
var followLight = new THREE.DirectionalLight(0xFFFFFF,10);  
followLight.position.copy(camera.position);  
camera.add(followLight);
```

En dessous, je vais ajouter une première lumière qui va faire office de «flash» d'appareil photo. Je choisis une `DirectionalLight` avec une couleur blanche et une intensité de 10. Je la place en temps qu' enfant de la caméra, comme cela elles sont liées.

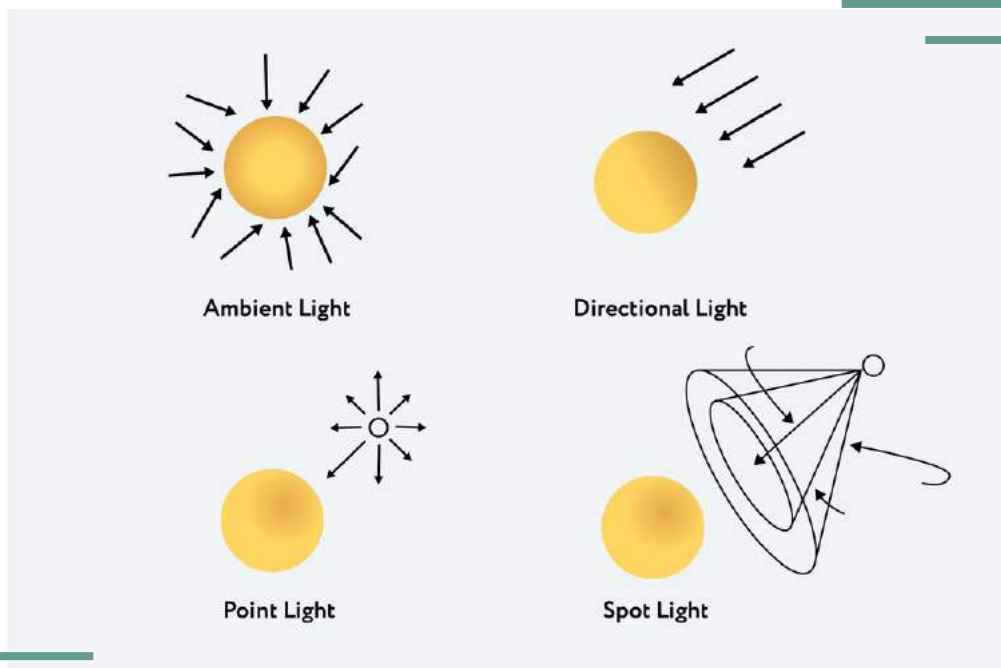
Pour le reste des lumières, il faut imaginer une sphère avec les axes X, Y et Z dont le point 0 est le centre de la forme.

Sur le site de geogebra, il y a possibilité de créer un environnement 3D avec ce dont j'ai besoin.



Le point A est le centre de la sphère et l'intersection entre les axes. C'est là où devrait être notre objet. Les points B et C (sur l'axe bleu), E et D (sur l'axe rouge), G et F (sur l'axe vert) sont les emplacements des lumières que nous allons créer.

Il y a 4 types de lumières qui existent dans le **THREE.JS**. Les voici sous forme de schéma afin de bien les comprendre. Mon choix se porte sur une **ambient light** et des **directional light**.



```
var ambientLight = new THREE.AmbientLight(0xFFFFFF, 110);
scene.add(ambientLight);
```

ambient light

```
var light = new THREE.DirectionalLight(0xFFFFFF,10);
light.position.set(chiffflight,0,0);
scene.add(light);
```

```
var light1 = new THREE.DirectionalLight(0xFFFFFF,10);
light1.position.set(0,chiffflight,0);
scene.add(light1);
```

```
var light2 = new THREE.DirectionalLight(0xFFFFFF,10);
light2.position.set(0,0,chiffflight);
scene.add(light2);
```

```
var light3 = new THREE.DirectionalLight(0xFFFFFF,10);
light3.position.set(0,0,-chiffflight);
scene.add(light3);
```

```
var light4 = new THREE.DirectionalLight(0xFFFFFF,10);
light4.position.set(0,-chiffflight,0);
scene.add(light4);
```

```
var light5 = new THREE.DirectionalLight(0xFFFFFF,10);
light5.position.set(-chiffflight,0,0);
scene.add(light5);
```

Directional light avec les positions que nous avons vues précédemment.

Cette partie 3D interactive est désormais terminée. Il suffit de changer le fichier «**scene.glb**» pour changer d'arme. Le code restera le même.



# Résultats



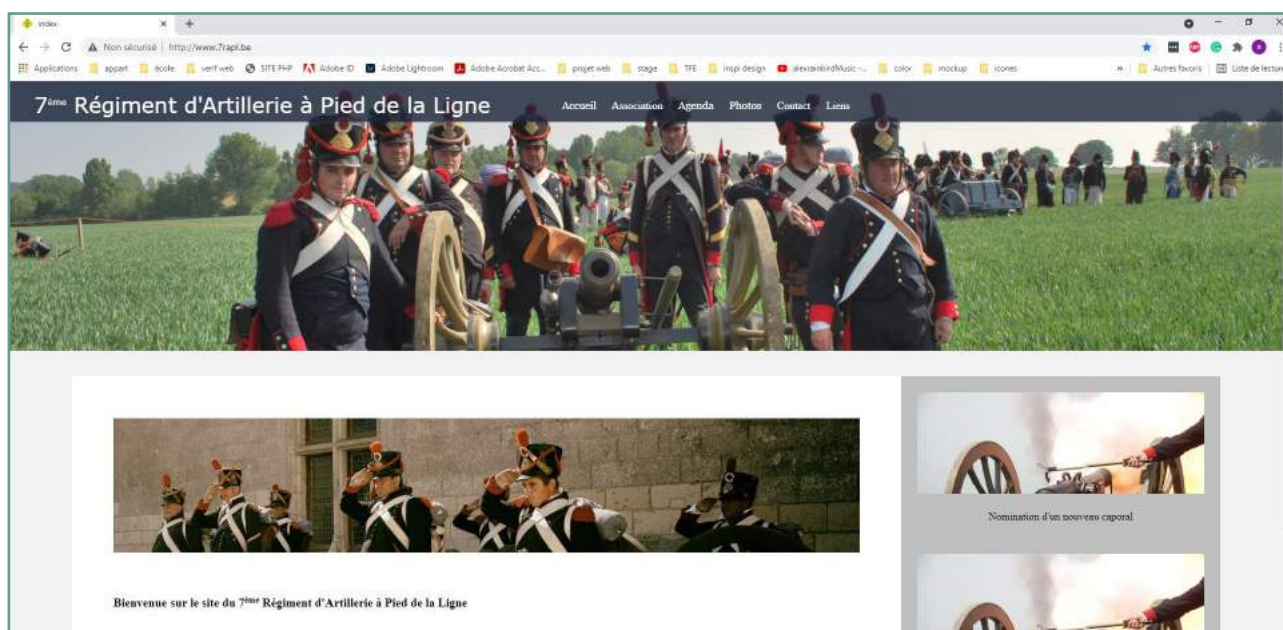
5



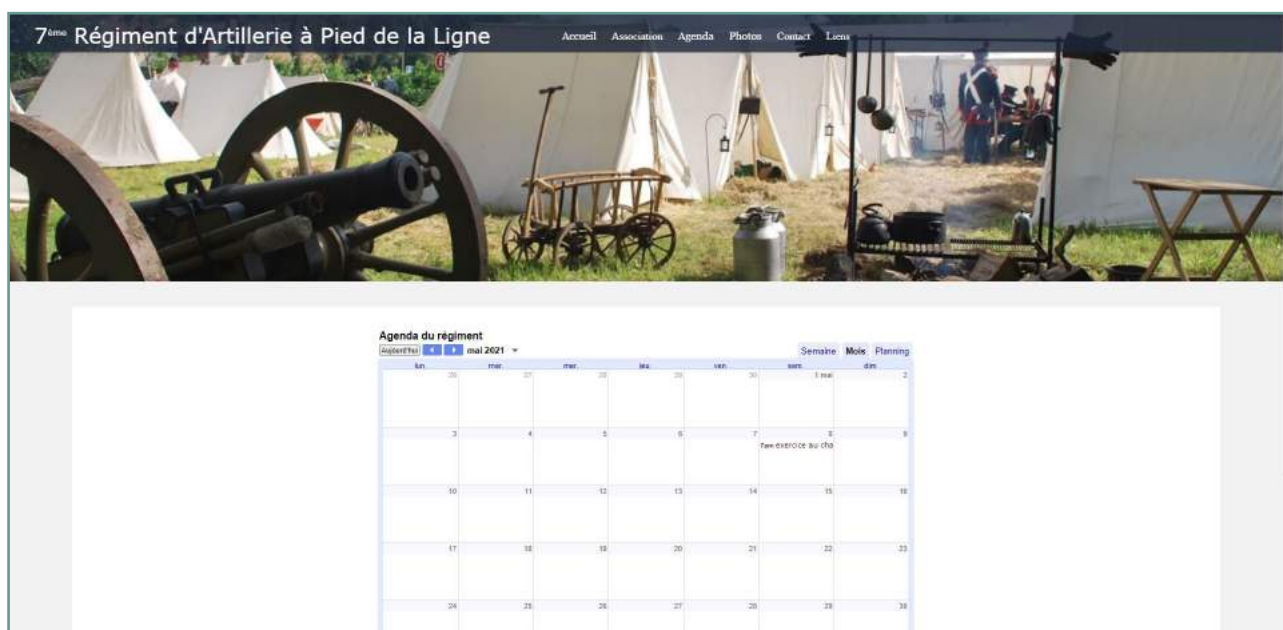
Le résultat de mon TFE est visionnable sur mon site personnel [axelleswert.be/7rapl](http://axelleswert.be/7rapl) et sur le site de l'association [7rapl.be](http://7rapl.be).

Voici quelques screens comparatifs du avant et après. Quelques photos sont encore susceptibles d'être changées.

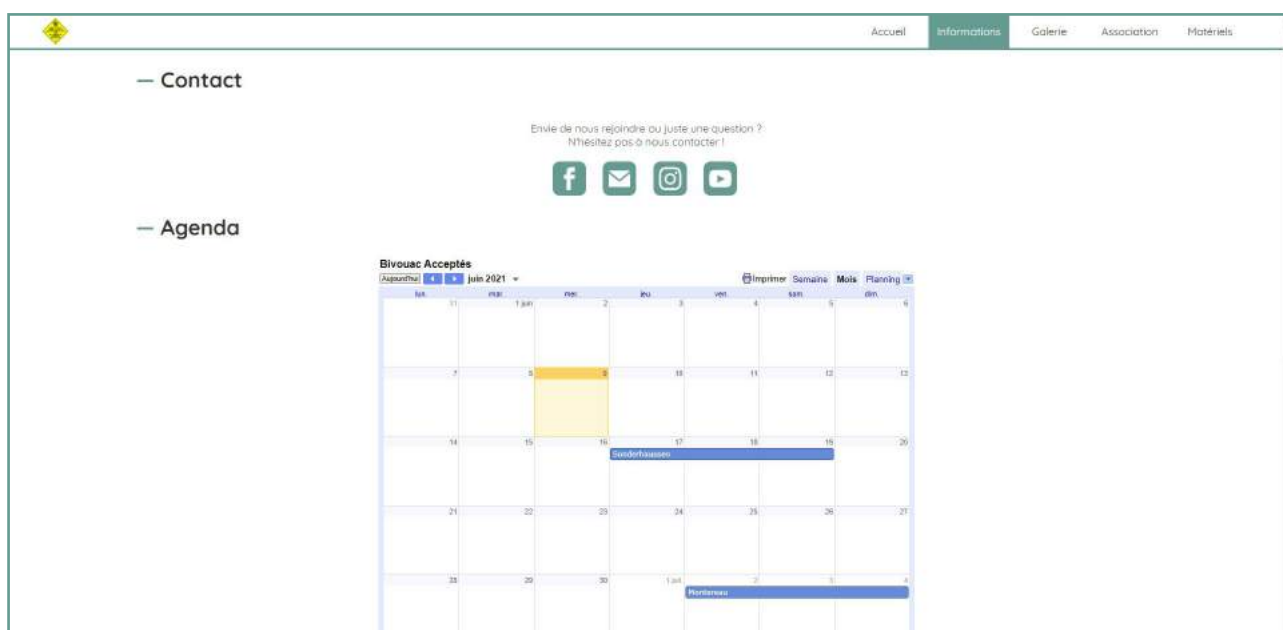
Page « Accueil »



Une belle harmonie s'est créée. La photo est mise en avant d'une meilleure manière. On voit directement les informations importantes.

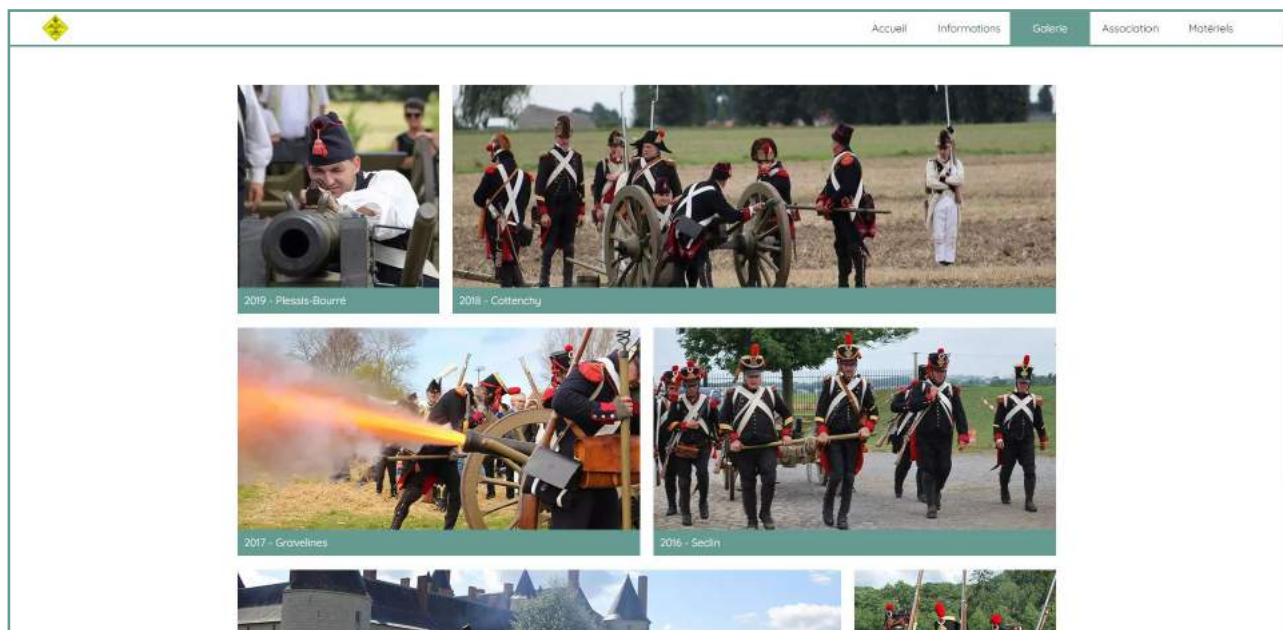


La page est plus lumineuse et regroupe plus d'informations. Au lieu d'avoir une page contact, une page agenda et une page déplacement, j'ai une page informations avec tout. C'est plus simple pour savoir la date et le lieu d'un évènement.



## Page « Galerie »

La page n'existait pas auparavant. Le tri des photos et leur mise en avant a été une vraie mission étant donné que je devais choisir parmi des centaines de photos «amateurs». (Pour la suite, je leur ai indiqué le format et les dimensions préférables)



## Sous-page album

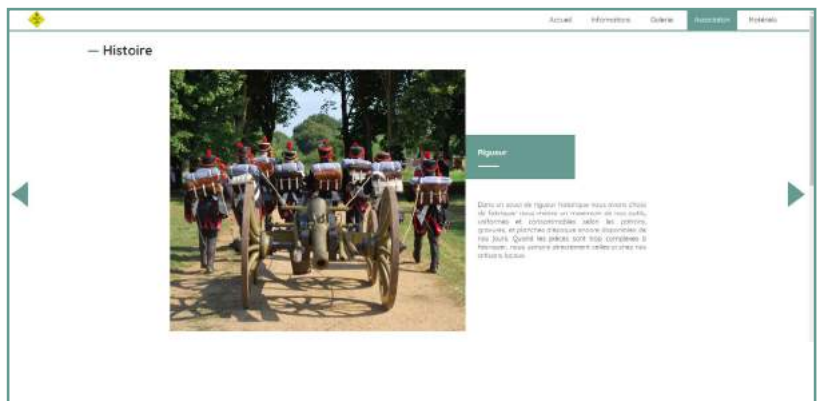






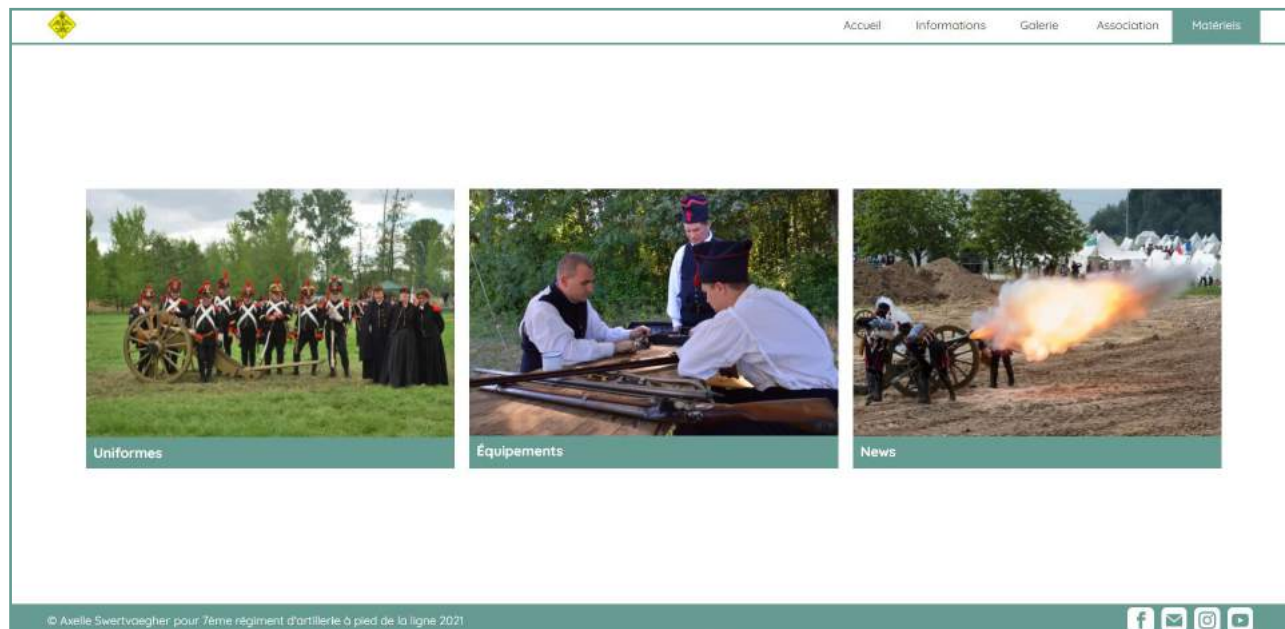
L'histoire de l'association est plus agréable à lire. C'est également plus rapide. On a changé les textes à rallonge par des textes courts et efficaces.

La couleur verte apporte vraiment une harmonie et adoucit l'image de l'asbl qui parle de reconstitutions de batailles.



La hiérarchie des membres est visible en entier, ils sont cliquables. On peut en apprendre plus sur l'ambiance qui règne au sein de l'asbl grâce à leurs descriptions présentes dans leurs profils.





L'ancien site n'avait pas de page regroupant les news, les uniformes et les équipements. Il avait des petites news sur la page d'accueil et une page explicative sur leur canon.

### Sous-pages uniformes et news au format téléphone



## Sous-page équipements

7<sup>ème</sup> Régiment d'Artillerie à Pied de la Ligne
Accueil Association Agenda Photos Contact Liens

### La pièce d'artillerie du régiment

Type: Canon de 4 livres

Modèle: Gribeauval 1765

Echelle: 1/1

Mise en service: 2013

Munition:  
Boulet de 4 livres (1,958 kg)  
Boite à balles (mitraille)

Portée:  
800m théorique  
400m efficace

Personnel: 8 hommes

Masse à vide: 1 tonne

Le canon était alors seul sur une page. Avec mon promoteur externe, nous avons décidé d'ajouter d'autres armes et même une partie innovante avec la 3D interactive.

gardant un esthétisme à la hauteur de l'artillerie.

**Propriétés**

Modèle > suivant l'an IX  
Masse à vide > 4,4kg  
Calibre > 17,5mm  
Longueur > 1028,3mm

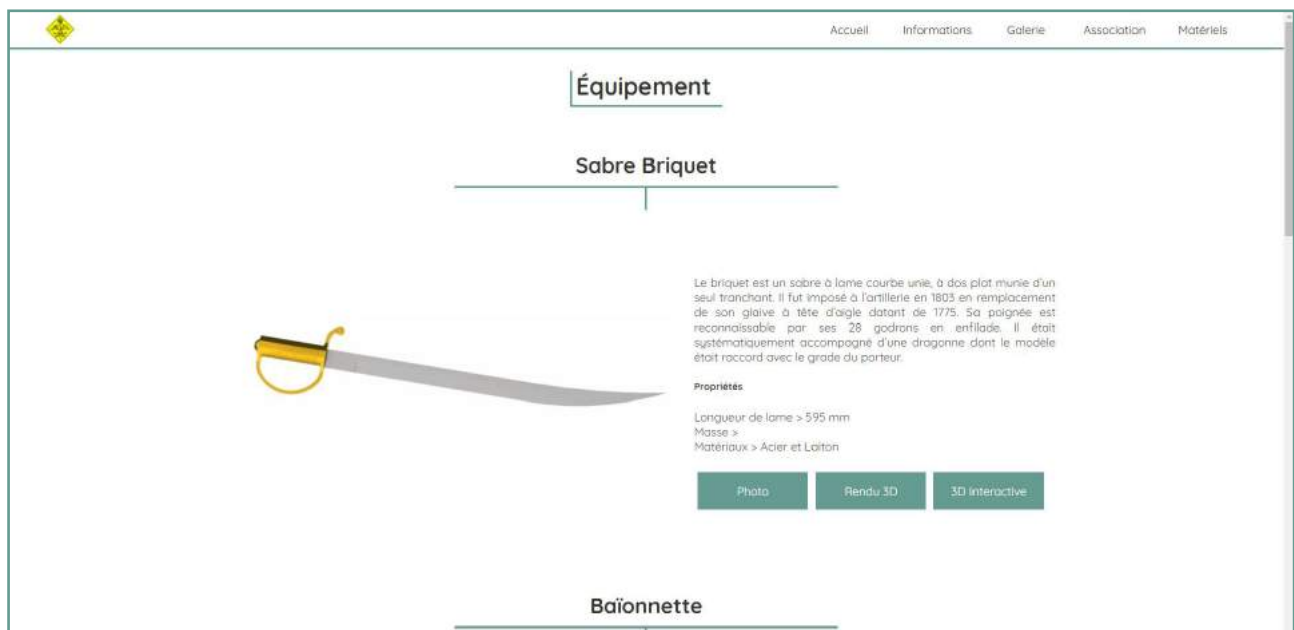
## Canon

La pièce de 4 était, sous l'Empire, la pièce par excellence qui accompagnait l'infanterie de ligne. Mobile et légère, elle permettait d'avancer en même temps que les lignes pour soutenir ces dernières grâce à sa puissance de feu tant redoutée par l'infanterie ennemie. Avec une cadence de feu moyenne allant de 2 à 3 coups par minute, elle permettait dans un premier temps de frapper des colonnes entières d'hommes et de chevaux pour ensuite terminer les pelotons à la mitraille quand ceux-ci arrivaient à trop près des lignes françaises. En campagne, la pièce de 4 était tractée par 3 chevaux qui laissaient leur place à 8 hommes dès que la bataille commençait. Elle était suivie de près par un coffre à munition qui permettait de lui délivrer 170 coups sans être ravitaillé par le train d'artillerie qui suivait le détachement.

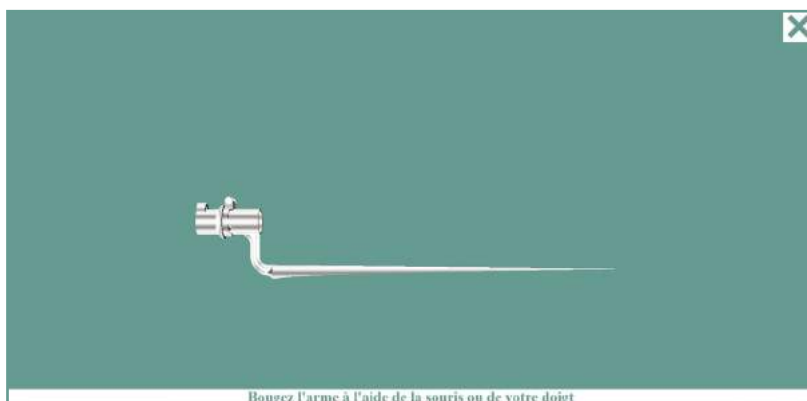
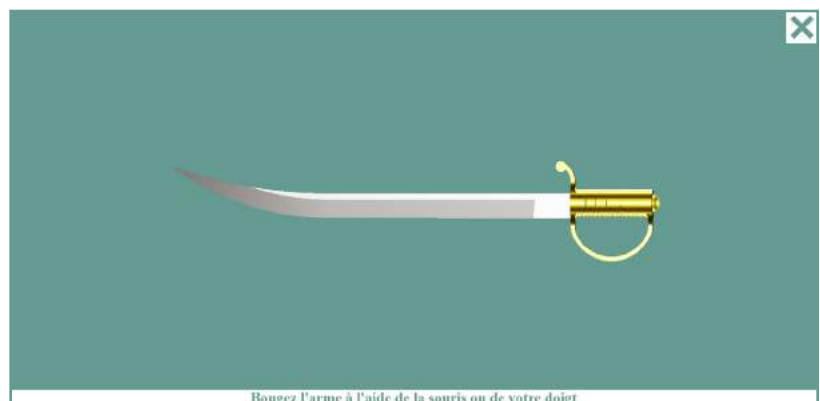
**Propriétés**

Masse à vide > 1 tonne  
Modèle > Gribeauval 1765  
Mise en service > 2013  
Munitions > Boulet de 4 livres (1,958 kg)  
Portée > 800m (théorique), 400m (efficace)

© Axelle Swertvaegher pour 7<sup>ème</sup> régiment d'artillerie à pied de la ligne 2021



## Pages 3D interactive

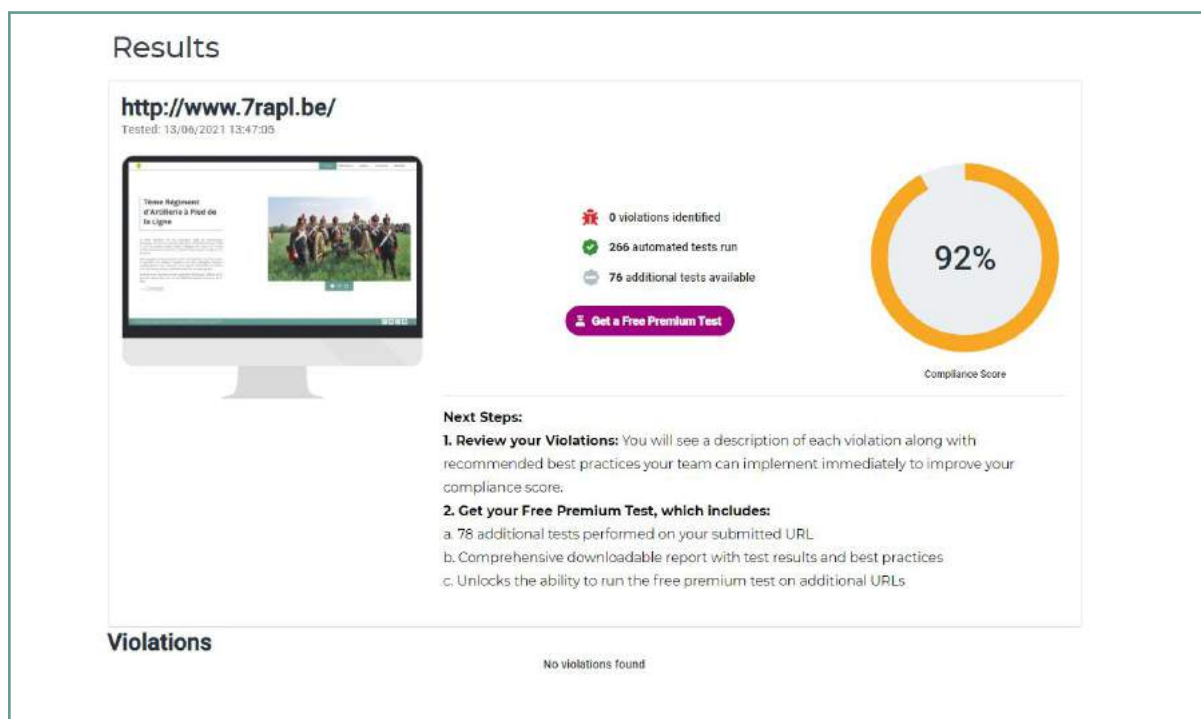




Une fois le site en ligne, je vais procéder à diverses vérifications de mon code voici quelques unes d'entre elles.

## web accessibility

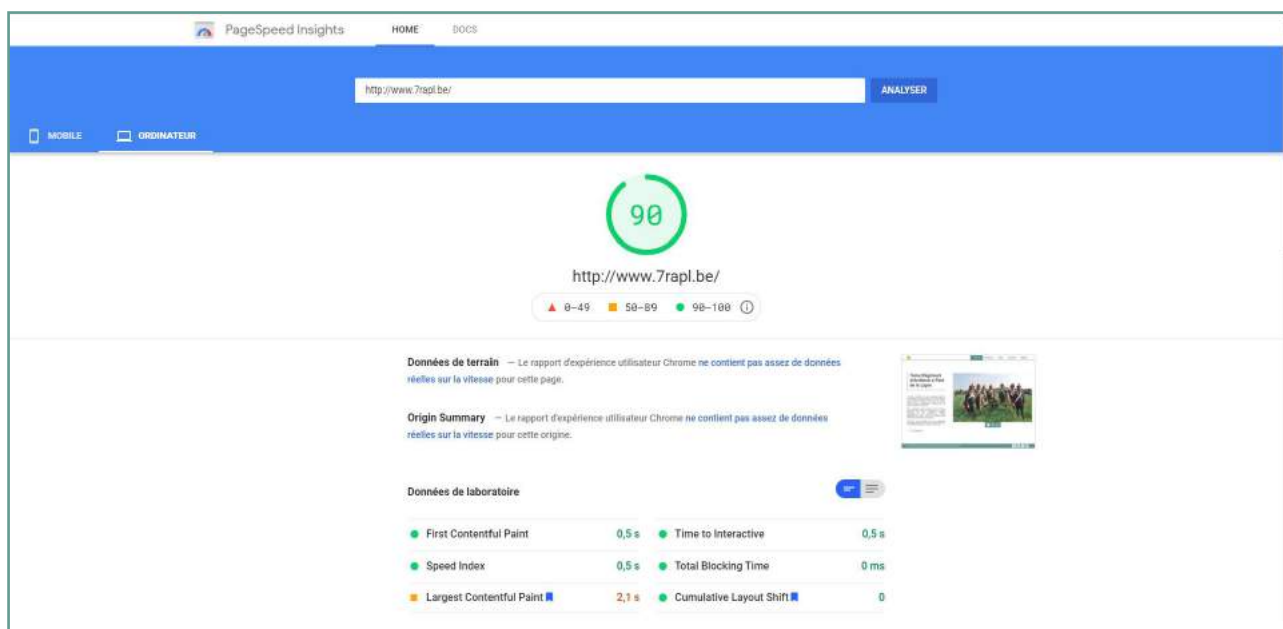
Ce site permet de tester si il est accessible aux personnes handicapées. Ici, 92% est très correct.



A titre d'exemple, lorsque je devais changer la taille de la police dans mon code, j'utilisais une unité de mesure en rem au lieu de px. Celle-ci est changeable selon les paramètres de préférence que le visiteur a indiqué à son navigateur.  
«font-size:1.2rem»

## PageSpeed Insights

A l'aide de ce site internet, je vais pouvoir savoir combien de temps il faut au site pour s'afficher correctement. 90% sur un ordinateur est convenable.



## Alyze

Sur ce site d'analyse de référencement naturel, je peux voir que j'ai correctement codé le site.

<div> <div>Classement</div> <div>Associations</div> <div>Suggestions</div> </div>					
Afficher les expressions composées de : <input checked="" type="checkbox"/> 1 mot <input checked="" type="checkbox"/> 2 mots <input checked="" type="checkbox"/> 3 mots et plus   ?					
N.	Expressions	Densité pondérée (%) ▼	Densité brute (%) ▼	Occ.	Trouvée dans les balises suivantes
1	7rapl	32.8	5	12	DTB TB url.host • title • meta.keywords <sup>x2</sup> • header/a/img.alt <sup>x2</sup> • main/img.alt • main/article • footer/a/img.alt <sup>x4</sup>
2	7ème régiment d'artillerie à pied	9.39	7.5	3	DP header/a/img.alt • main/article/h1 • footer
3	artillerie	9.09	2.5	6	DB meta.description • meta.keywords • header/a/img.alt • main/article/h1 • main/article • footer
4	7ème	7.34	1.67	4	DM header/a/img.alt • main/article/h1 • main/article • footer
5	artillerie à pied de la ligne	7.28	7.5	3	DP header/a/img.alt • main/article/h1 • footer
6	régiment	5.13	1.25	3	DP header/a/img.alt • main/article/h1 • footer
7	ligne	5.13	1.25	3	DP header/a/img.alt • main/article/h1 • footer
8	pied	5.13	1.25	3	DP header/a/img.alt • main/article/h1 • footer
9	époque	0.85	0.83	2	DF meta.description • main/article
10	durant	0.85	0.83	2	DF meta.description • main/article

La colonne «expression» correspond aux mots qui seront cherchés par le visiteur sur google dans le but de trouver le site web de l'asbl 7RAPL.



# Conclusion



6

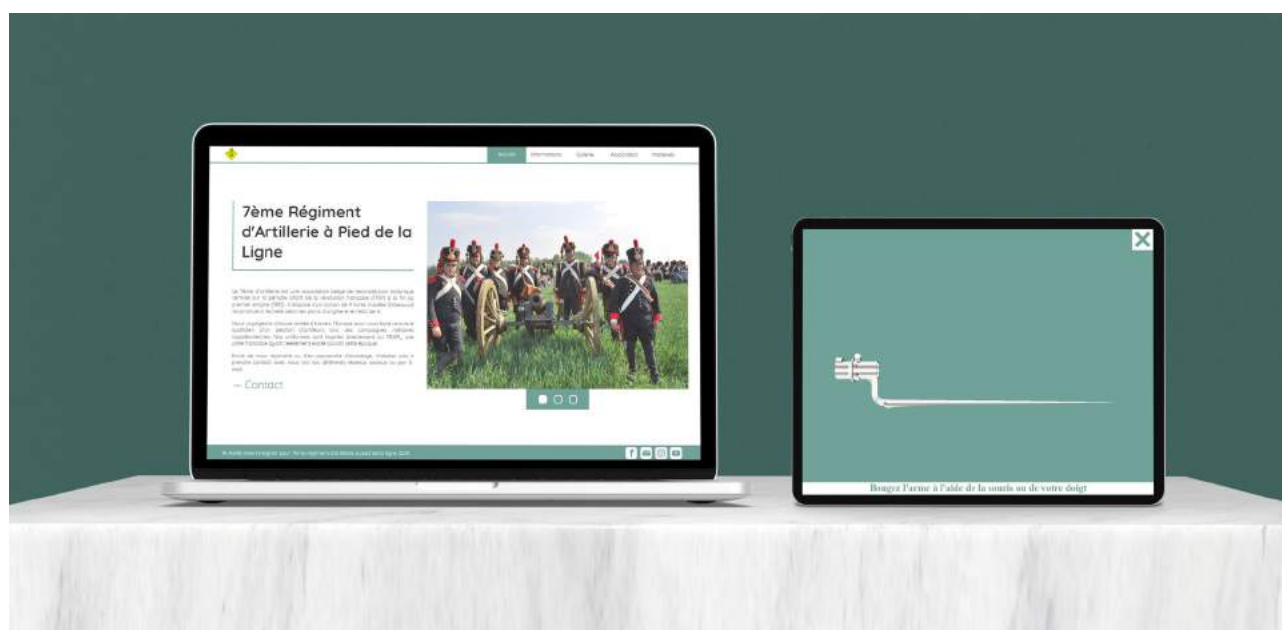
Nous arrivons à la fin de ce rapport, j'ai relevé le défi que je m'étais lancé en choisissant ce projet alliant le graphisme, la 3D et le Web. Je suis contente d'avoir pu démontrer qu'il était possible d'effectuer un Travail de fin d'études complet, basé sur les trois grands axes de ma section.

Ce site web m'a permis d'aller au delà de mes retranchements et de sortir de ma zone de confort. J'ai pu m'appuyer sur mes bases de programmation tout en allant chercher encore plus loin pour optimiser le code, donner plus de fonctionnalités et même rendre une partie interactive.

J'ai acquis de nouvelles compétences autant professionnelles que personnelles. Ce projet m'a confirmé qu'avoir une bonne organisation et une certaine rigueur de travail est primordiale.

Je pense avoir répondu aux attentes et à la demande de mon promoteur externe Monsieur Loïc Michaux. N'ayant pas de page « administrateur », en accord avec l'association, je reste à la disposition de l'asbl afin de pouvoir ajouter des images, des news,...

Je suis fière du résultat et du parcours que j'ai effectué durant mes 3 années d'études. Car oui pour moi ce travail de longue haleine est la « cerise sur le gâteau » de ces 3 ans !



Le site web est fonctionnel sur tous les navigateurs et sur n'importe quel support. Cependant, comme dans tout travail, il y a toujours moyen de faire plus. Voici trois améliorations possibles de mon TFE :

- > Ajouter une page administrateur.
- > Rendre la page news plus détaillée (je n'ai pas pu le faire par manque de contenu)
- > Mettre plus d'armes en 3D.
- > Montage et démontage d'armes en 3D.
- > Coder l'agenda soi-même afin de pouvoir le rendre plus design.



# Sources



7



## Sitographie

<http://www.7rapl.be/>  
<https://www.pinterest.fr/>  
<https://dribbble.com/>  
<https://www.behance.net/>  
<https://www.awwwards.com/>  
<https://coolors.co/659b91-bee0e4-fefe00-444444-ffffff>  
<https://onlineconvertfree.com/fr/complete/jpg-webp/>  
<https://www.geogebra.org/?lang=fr>  
<https://threejs.org/>  
<https://www.php.net/>  
<https://stackoverflow.com/>  
<https://www.intexsoft.com/blog/post/three-js.html>  
<https://developer.mozilla.org/>  
<https://support.google.com/>  
<https://www.w3schools.com/>  
<https://www.youtube.com/watch?v=jmld3mIKBlo&t=195s>  
<https://www.youtube.com/watch?v=-zGbbUPgKdk>  
<https://alyze.com/>  
<https://developers.google.com/speed/pagespeed/insights/?hl=fr>  
<https://validator.w3.org/>  
<https://jigsaw.w3.org/css-validator/>  
<https://www.webaccessibility.com/>  
<https://contrast-ratio.com/>

## Bibliographie

Syllabus des cours de programmations front-end et back-end (Javascript, JQuery et PHP)  
Les cours de design web.  
Syllabus de DAO.



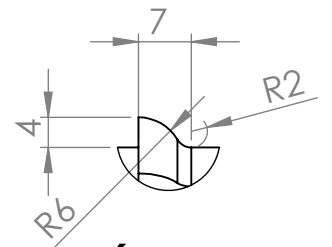
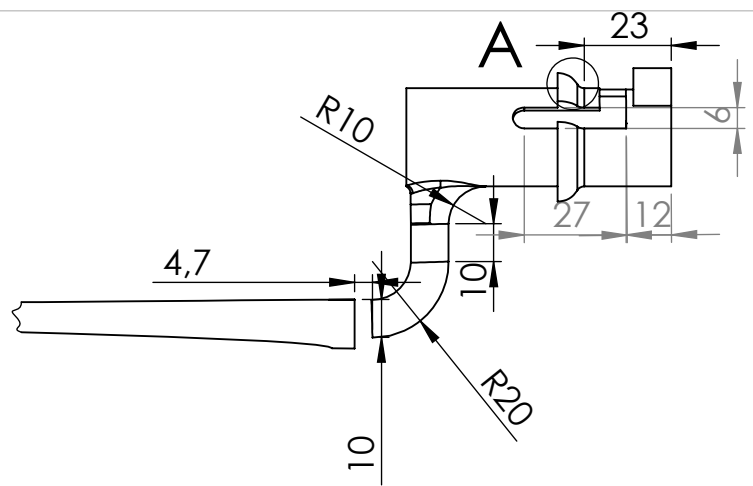
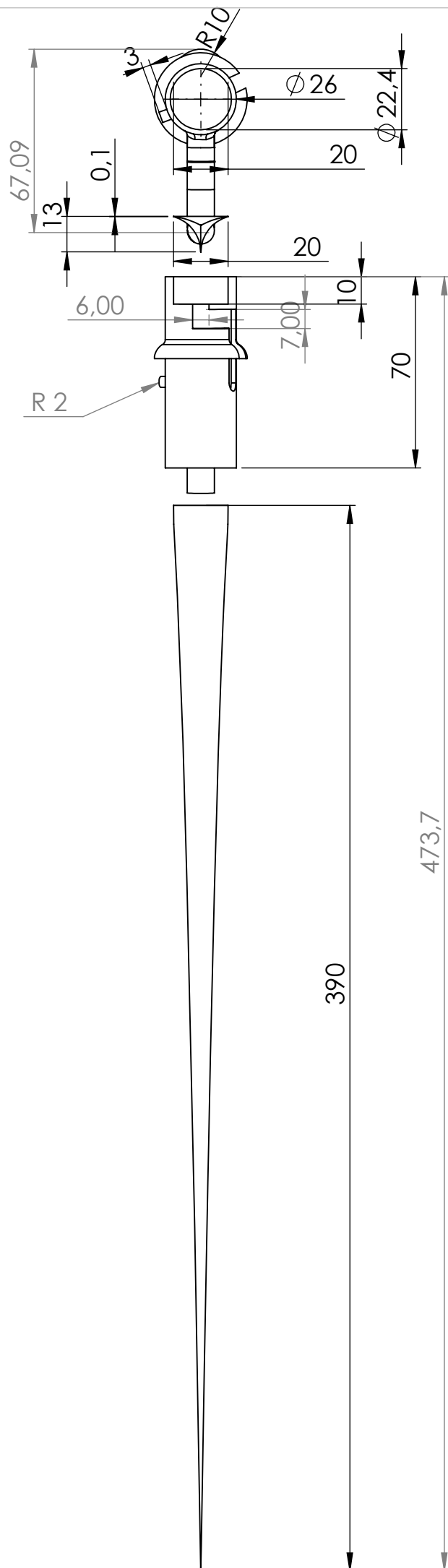
# Annexes



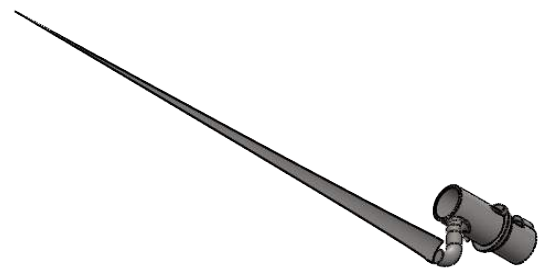
8






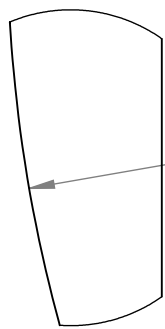


**DÉTAIL A**  
ECHELLE 1 : 1

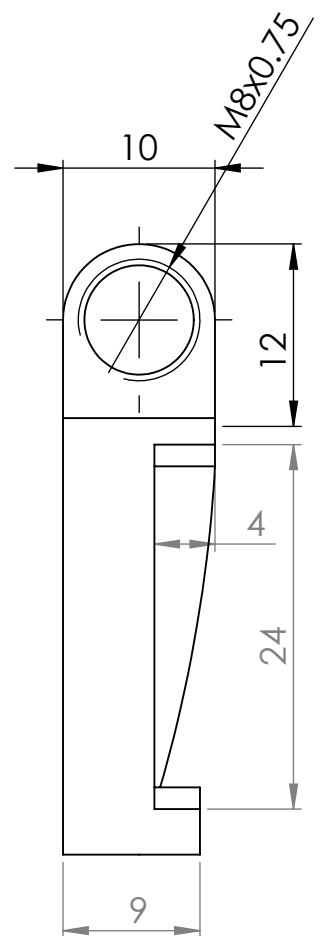
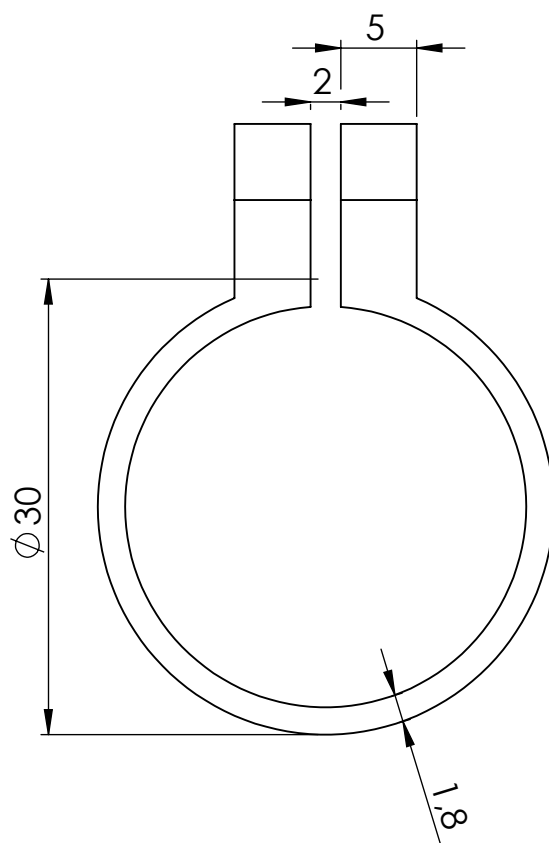
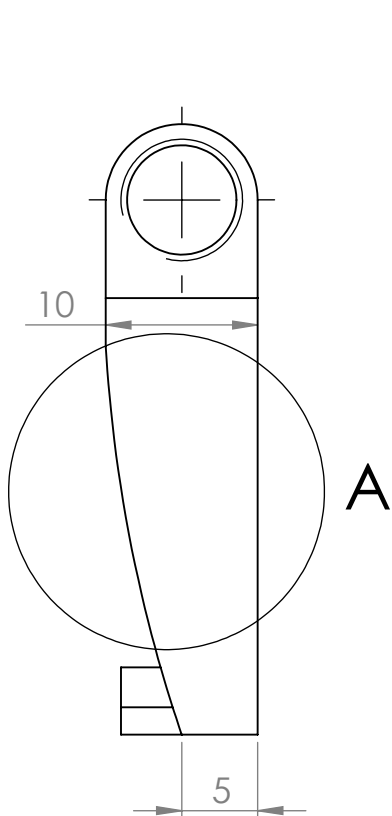


H.E.H. TFE 7rapl	13/05/2021
Baïonnette (lame)	Axelle SWERTVAEGHER
	Ech.: 1/2 
	Annexe 1 Bac. 3

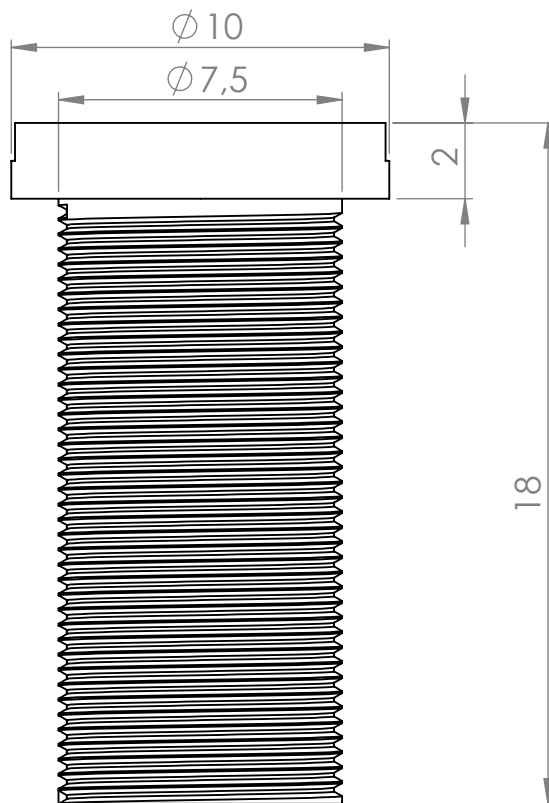
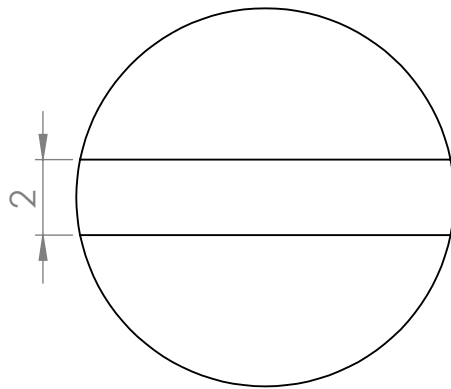
# DÉTAIL A



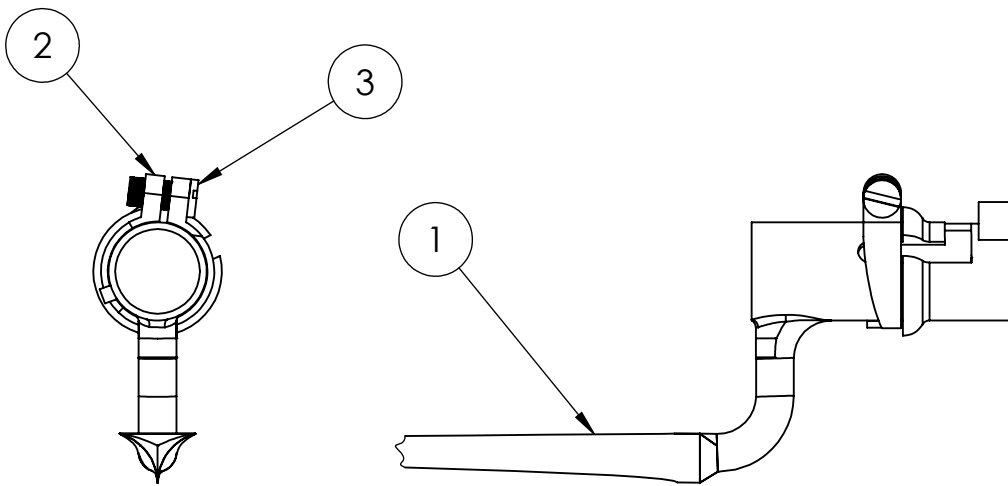
R97



H.E.H. TFE 7rapl	13/05/2021
Baïonnette (Attache)	Axelle SWERTVAEGHER
	Ech.: 2/1
	Annexe 2 Bac. 3

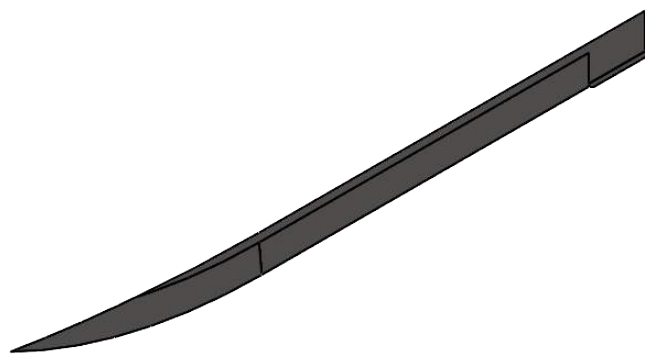
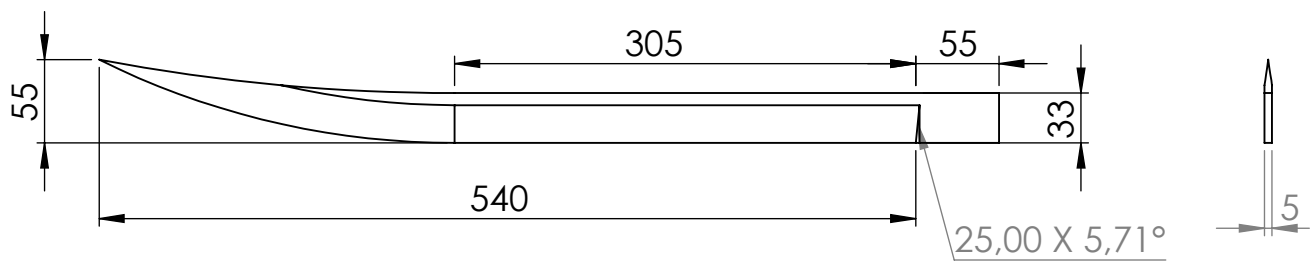


H.E.H. TFE 7rapl	13/05/2021	
Baïonnette (vis)	Axelle SWERTVAEGHER	
	Ech.: 5/1	
	Annexe 3	Bac. 3

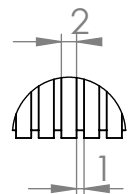
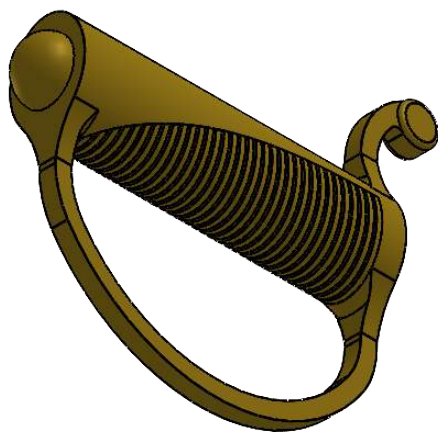
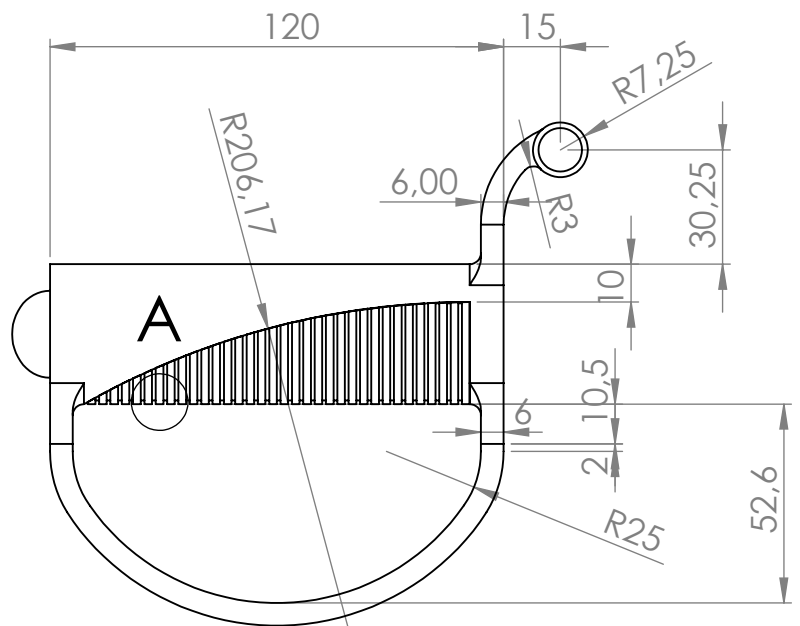
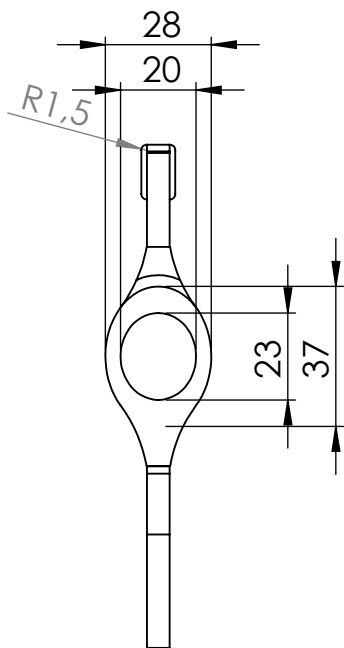


N°	Nom de pièce
1	baïonnettefinal
2	attachebaïonnette
3	vis


H.E.H. TFE 7rapl	13/05/2021	
Baïonnette	Axelle SWERTVAEGHER	
	Ech.: 1/2	
	Annexe 4	Bac. 3

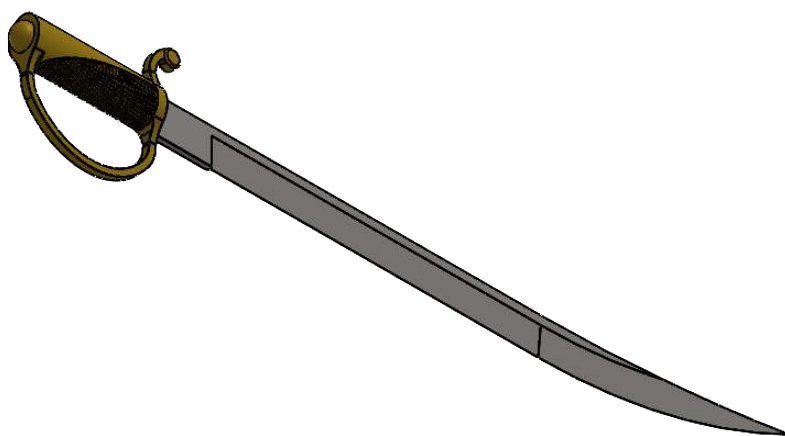
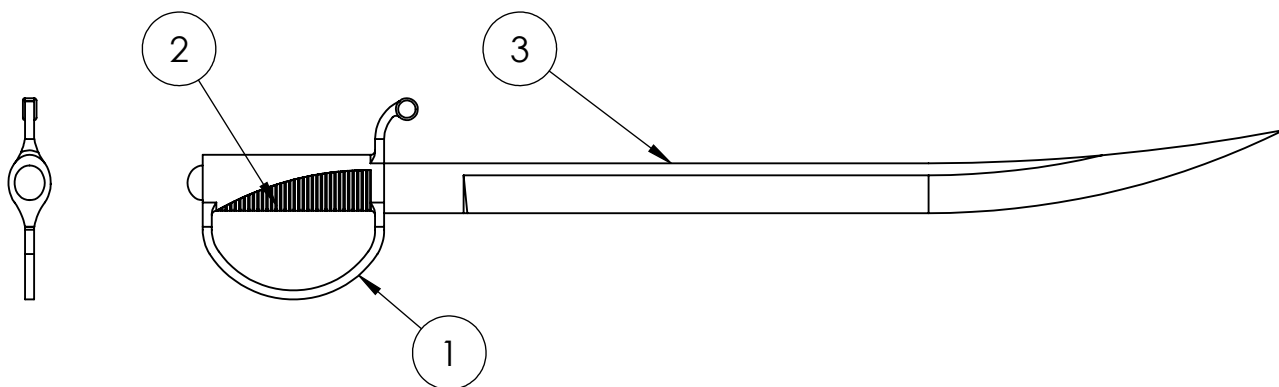


H.E.H. TFE 7rapl	13/05/2021	
Sabre (lame)	Axelle SWERTVAEGHER	
	Ech.: 1/5	
	Annexe 5	Bac. 3



**DÉTAIL A**  
ECHELLE 1 : 1

H.E.H. TFE 7rapl	13/05/2021
Sabre (poignée)	Axelle SWERTVAEGHER
	Ech.: 1/2 
	Annexe 6 Bac. 3



N°	Noms des pièces
1	poignée
2	lame
3	intérieur manche

H.E.H. TFE 7rapl	13/05/2021
Sabre	Axelle SWERTVAEGHER
	Ech.: 1/5 
	Annexe 7 Bac. 3



Ce document reprend les différentes étapes de création de mon travail de fin d'études. Il a pour objectif de créer un site web pour l'association « 7ème Régiment d'Artillerie à Pied de la Ligne ». Mes compétences dans le graphisme, la modélisation 3D et la programmation ont été nécessaires. Je vous explique chapitre par chapitre ces différents secteurs. Pour amener de la dynamique, j'ai fait la modélisation d'armes en 3D que j'ai rendu interactives sur le site web.

This document summarises the different stages of creation of my end-of-study work. Its objective is to create a website for the association «7ème Régiment d'Artillerie à Pied de la Ligne». My skills in graphic design, 3D modelling and programming were necessary. I explain you chapter by chapter these different sectors. To bring dynamics, I designed 3D weapons that I made interactive on the website.