

Projets réalisés dans le cadre du passage du Titre Professionnel de
DÉVELOPPEUR WEB et WEB MOBILE

Présenté par *Jessy CHARLET*

La_Plateforme - Toulon 2024

Projet 1/2

ALTERED COUNTER

Application compagnon pour le TCG Altered



SOMMAIRE

Introduction	... 4
Compétences du référentiel couvertes par le projet	... 6
Activité type n°1	... 6
Développer la partie front-end d'une application web ou web mobile sécurisée	... 6
- Maquetter une application	... 6
- Réaliser une interface utilisateur web statique et adaptable	... 6
- Développer une interface utilisateur web dynamique	... 6
Résumé du projet	... 7
Cahier des charges	... 8
Objectifs	... 8
Conception de l'application	... 8
- MVP (minimum Viable Product)	... 8
- Liste des fonctionnalités	... 9
Mise en place de l'application	... 10
- Arborescence	... 10
- Wireframes	... 10
- Charte graphique et logo / Accessibilité	... 12
Utilisateurs et User Stories	... 15
- Public visé	... 15
- Rôles	... 15
- User Stories	... 15
Évolutions éventuelles	... 16

Spécifications techniques	... 17
Versioning	... 17
Choix technologiques	... 18
- Front-End	... 18
- Back-End	... 18
Accessibilité	... 18
- Types d'appareils	... 18
- PWA (Progressive Web App)	... 18
Déploiement	... 19
Le code	... 21
HTML	... 21
CSS	... 24
JavaScript / Jquery	... 25
Webmanifest	... 32

INTRODUCTION

Jessy Charlet, âgé de 35 ans, et je suis passionné par la technologie, la logique et la création. Mon parcours professionnel est riche et varié, marqué par une évolution constante et une adaptation à différents domaines tenant au commerce et à la technologie.

Mon Parcours Professionnel

Début de carrière dans la gravure pour bijoux (4 ans)

J'ai commencé ma carrière professionnelle à l'âge de 16 ans dans la gravure pour bijoux. Ce métier artisanal m'a permis de développer un sens aigu du détail et de la précision.

Évolution vers la maintenance et la création visuelle en ligne

Ensuite, j'ai évolué vers des responsabilités plus techniques et créatives en m'occupant de la maintenance de la boutique en ligne de l'entreprise. J'ai aussi créé la partie visuelle et promotionnelle de cette boutique, ce qui m'a permis de combiner mes compétences techniques et artistiques.

Ouverture d'une boutique d'impression à Hyères (5 ans)

Fort de cette expérience, j'ai ouvert ma propre boutique dans le centre-ville de Hyères, spécialisée dans l'impression sur tous supports et destinée aux particuliers particuliers. Durant 5 ans, j'ai appris à gérer cette entreprise, répondant aux besoins variés de mes clients ainsi qu'en développant mes compétences en gestion d'entreprise.

Expérience chez Micromania (1 an)

Par la suite, je suis resté un an chez Micromania, une expérience qui m'a permis de me familiariser avec le secteur du jeu vidéo et de la pop culture.

Création d'une entreprise de produits dérivés de la pop culture (7 ans)

Ma passion pour la pop culture m'a conduit à ouvrir une entreprise de vente de produits dérivés sur diverses marketplaces comme Amazon, AliExpress, et Fnac. Au cours de ces 7 années, j'ai dirigé cette entreprise, tout en gérant la chaîne d'approvisionnement, le marketing, et les ventes en ligne.

Passion pour la Technologie et le Développement

Tout au long de ma carrière professionnelle, j'ai toujours eu énormément d'attrait pour la technologie et la création. Cette passion m'a conduit à vouloir approfondir mes compétences en développement. Actuellement, je me concentre sur le développement web, tout en m'intéressant également beaucoup au développement d'applications et de logiciels.

Objectifs de Formation

Après avoir travaillé de nombreuses années dans le commerce, j'ai décidé de franchir le pas et de me lancer dans le développement. Je suis actuellement en formation afin d'atteindre cet objectif et j'ai l'intention de continuer cette voie en passant le diplôme de Concepteur Développeur d'Applications (CDA) en alternance dès la rentrée prochaine.

Mon objectif est de combiner mon expérience diversifiée en commerce et en technologie avec des compétences avancées en développement, afin de créer des solutions innovantes et répondre aux besoins actuels du marché.

COMPÉTENCES DU RÉFÉRENTIEL COUVERTES PAR LE PROJET

Altered Counter est un projet développé en Javascript avec la bibliothèque Jquery. Il a été réalisé au moment où j'ai découvert cette technologie lors de la formation. Avec du recul il aurait été bien plus simple de le développer avec le framework Angular (que j'ai découvert bien après).

Activité type n°1

Développer la partie front-end d'une application web ou web mobile sécurisée

- **Maquetter une application**

Il fallait dans un premier temps rédiger un cahier des charges et élaborer tous les documents de conception d'Altered Counter. J'ai rédigé les users stories et défini le MVP(Minimum Viable Product) qui m'a permis de créer le wireframes du projet sous Figma.

- **Réaliser une interface utilisateur web statique et adaptable**

Ensuite, j'ai réalisé une intégration (uniquement HTML / CSS) statique et responsive mobile-first (voir mobile only étant donné que le projet était de concevoir une application mobile, même si l'application desktop fonctionne parfaitement, l'utilité en reste bien moindre)

- **Développer une interface utilisateur web dynamique**

J'ai utilisé JavaScript avec Jquery pour manipuler les valeurs, faire les calculs, les comparaisons et ajuster l'affichage en fonction.

RÉSUMÉ DU PROJET



Altered TCG est un jeu de cartes à collectionner où les joueurs construisent des decks (paquets de cartes) personnalisés et s'affrontent en utilisant des stratégies complexes. Le jeu inclut des expéditions et des biomes, offrant des environnements variés qui influencent les combats et ajoutent une profondeur unique à l'expérience de jeu.

Le projet consiste à créer une application compagnon pour le jeu Altered. L'objectif principal de cette application est d'améliorer l'expérience des joueurs en leur offrant des informations en temps réel et des fonctionnalités pratiques pour mieux gérer leurs parties.

Fonctionnalités de l'Application

- **Aperçu des Forces sur Chaque Expédition :**
 - Les joueurs pourront rapidement voir les forces en présence sur chaque expédition, ce qui leur permettra de mieux planifier leurs stratégies et de prendre des décisions plus rapidement.
- **Informations sur les Biomes de Combat :**
 - L'application fournira des détails sur les biomes où se déroulent les combats. Cela aidera les joueurs à adapter leurs cartes et leurs stratégies en fonction des caractéristiques spécifiques de chaque biome.
- **Suivi de l'Avancement en Temps Réel :**
 - Les utilisateurs pourront voir qui est en tête dans une expédition en cours, ce qui ajoutera une dimension compétitive et interactive à leur expérience de jeu.

Processus de Développement

Pour s'assurer que l'application réponde aux besoins réels des joueurs, une phase de recherche approfondie a été menée. J'ai rejoint plusieurs communautés de joueurs d'Altered sur différents serveurs Discord afin de recueillir des retours d'expérience et identifier les besoins non satisfaits.

Les discussions avec les joueurs ont révélé un désir fort pour une solution qui leur permettrait d'avoir un aperçu clair et rapide de l'état des expéditions et des combats. Ils ont exprimé leur frustration face à la lisibilité globale du jeu et au manque d'outils centralisés pour gérer ces aspects du jeu, et ont montré un grand intérêt pour une application qui pourrait combler ce vide.

Cahier des charges

Objectifs

La création de cette application compagnon pour le TCG Altered vise à enrichir l'expérience des joueurs en leur fournissant des informations cruciales et en temps réel sur leurs expéditions et combats. En intégrant les retours directs des joueurs, l'application promet de devenir un outil indispensable pour la communauté Altered, facilitant une immersion plus profonde et une gestion plus efficace des parties.

Conceptualisation de l'application

- **MVP (Minimum Viable Product)**

Le MVP pour l'application Altered Counter doit se concentrer sur les fonctionnalités essentielles qui répondent aux besoins immédiats des joueurs et offrent une valeur ajoutée significative dès le lancement. Voici les éléments clés du MVP :

1. Tableau de Bord des Expéditions

-
- **Aperçu des forces** : Afficher une vue d'ensemble des forces en présence sur chaque expédition. Cette fonctionnalité permettra aux joueurs de visualiser rapidement la composition des équipes en compétition.
 - **Suivi de l'avancement** : Inclure un indicateur en temps réel de l'avancement des expéditions, permettant aux joueurs de voir qui est en tête et de suivre l'évolution de la situation.

2. Informations sur les Biomes

- **Détails des biomes actifs**: Fournir des descriptions succinctes des biomes de combat, avec des informations clés sur les caractéristiques et les effets spécifiques de chaque biome.
- **Ajout des personnages** : Offrir des conseils et des recommandations sur les stratégies à adopter en fonction du biome, basés sur les retours des joueurs et les meilleures pratiques.

3. Interface Utilisateur Simple et Intuitive

- **Navigation Facile** : Concevoir une interface utilisateur claire et facile à naviguer, permettant aux joueurs d'accéder rapidement aux informations dont ils ont besoin.
- **Compatibilité Multi-Plateforme** : Assurer que l'application fonctionne de manière fluide sur les principaux appareils mobiles (iOS et Android) et éventuellement sur le web.
- **Liste des fonctionnalités**

Du MVP, j'ai établi la liste des fonctionnalités du projet. Ces fonctionnalités concernent différentes parties de l'application.

Le site doit être responsive et adapté à tous les mobiles et tablettes. 95% des utilisations se feront sur ce support.

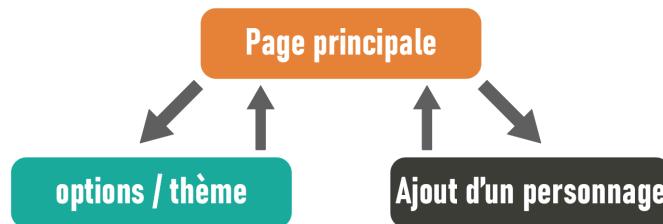
Les joueurs doivent pouvoir :

- activer ou désactiver les biomes d'un seul clic.
- modifier les valeurs à la main rapidement.

-
- ajouter un boost de +1 sur toutes les valeurs d'une expédition d'un seul coup.
 - retirer 1 point à toutes les valeurs d'une expédition d'un seul coup.
 - ajouter un personnage simplement sans avoir à calculer.
 - réinitialiser leurs expéditions d'un seul clic afin de passer au tour suivant.
 - pouvoir identifier à tout moment quel joueur avance sur quelle expédition.
 - choisir un thème représentant leur faction.

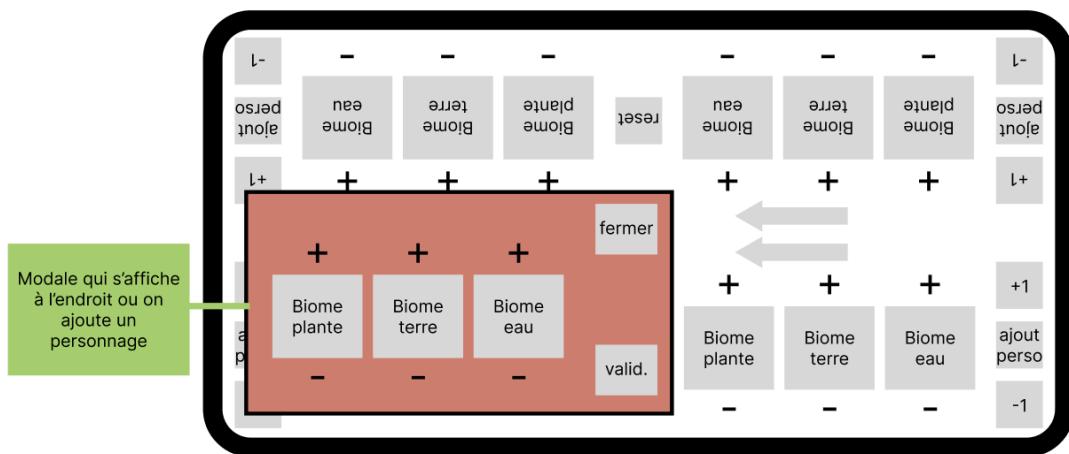
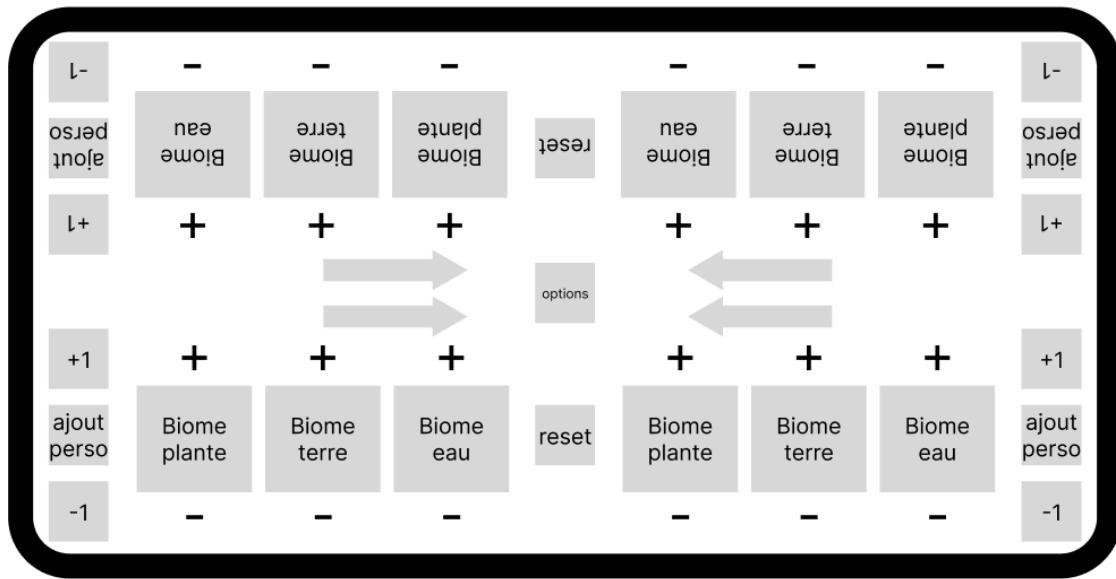
MISE EN PLACE DE L'APPLICATION

Arborescence



Wireframes

Comme évoqué précédemment dans les fonctionnalités, mon souhait était de concevoir une application responsive mobile / tablette. Pour ce faire, j'ai abordé la conception des Wireframes selon une approche mobile-first, et même principalement en mode paysage. Chaque joueur doit pouvoir lire son terrain facilement, ce qui a nécessité de scinder l'écran en deux pour que chaque utilisateur puisse avoir un sens de lecture adapté au mieux.



Charte graphique, logo et accessibilité

Dans le but d'avoir la meilleure accessibilité possible, j'ai choisi des couleurs assez contrastées, tout en gardant l'essence et le design du jeu. Concernant le logo, j'ai modifié ce dernier, notamment en par l'ajout "counter".

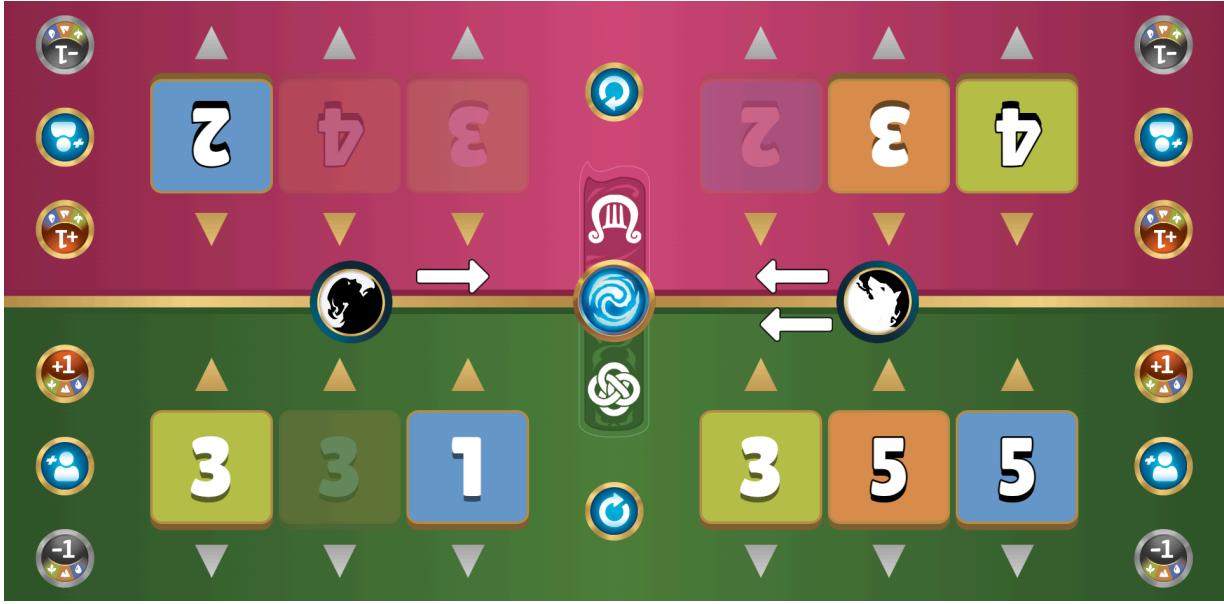
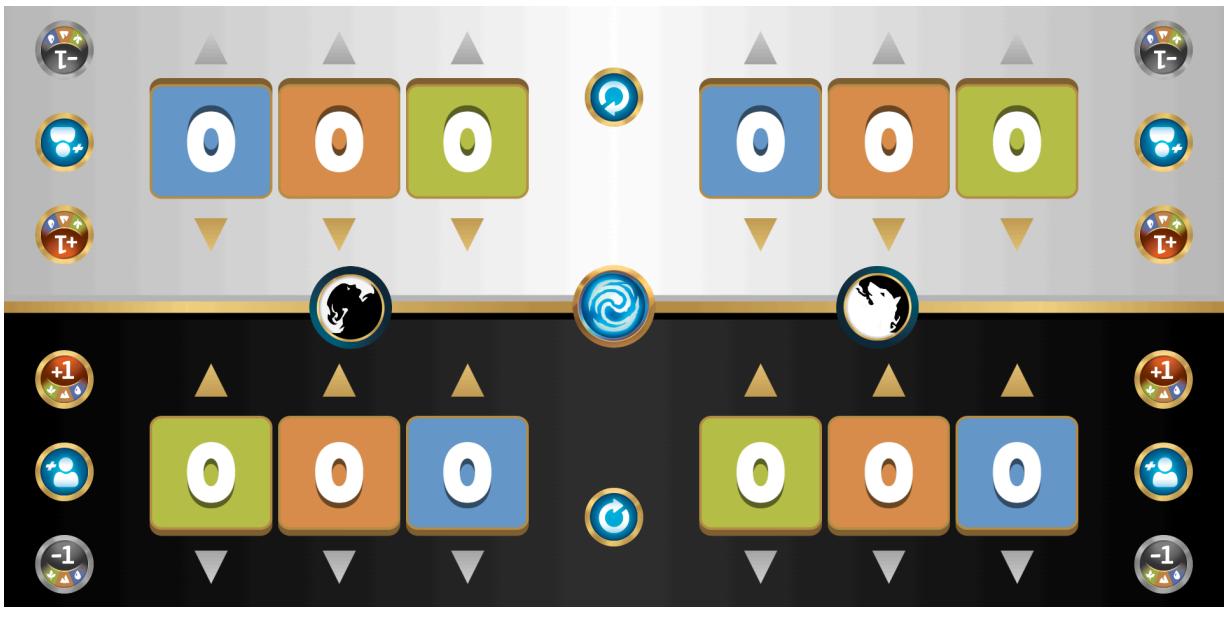
Le fond gris quant à lui, permet de faire ressortir le texte blanc sur le fond blanc de la page.

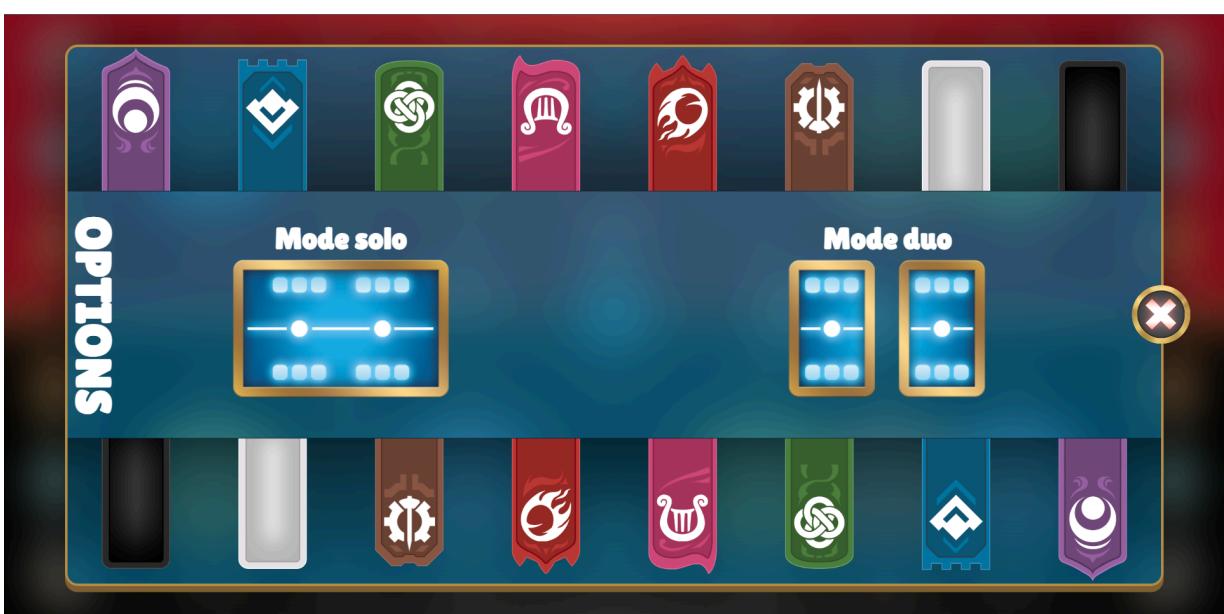


Moodboard



résultat final





UTILISATEURS ET USER STORIES

Public visé

Tous les joueurs du jeu de carte Altered. En conséquence, cela permet de toucher un large public, au regard du succès du jeu ainsi que de son ampleur internationale.
(Il à le record du TCG le plus backé sur kickstarter avec plus de 6 millions d'euros).

Rôles

Il n'y a qu'un seul rôle pour l'application. Le rôle user.

User Stories

En tant que	je souhaite	afin de
utilisateur	modifier les statistiques de mes biomes à la volé	être à jour sur mes expéditions
utilisateur	ajouter +1 /-1 à une expédition d'un seul clic	appliquer un boost ou un malus sans avoir à modifier toutes les valeurs à la main
utilisateur	ajouter un personnage	pouvoir voir et montrer à mon adversaire clairement ce que je fais sans risquer de faire d'erreur
utilisateur	réinitialiser mon board	ne pas avoir à remettre tout à zéro manuellement entre chaque tour
utilisateur	activer ou désactiver un biome en un clic	voir qui remporte l'expédition sans avoir à vérifier le board adverse
utilisateur	personnaliser mon interface	personnaliser un peu plus mon expérience et clarifier l'identification des joueurs pour les spectateurs

EVOLUTIONS EVENTUELLES

De nouvelles cartes avec de nouvelles mécaniques viennent d'être révélées, ce qui me m'incite à mettre à jour l'application pour intégrer au mieux ces évolutions.

En tant que	je souhaite	afin de
utilisateur	pouvoir lancer un dé	d'appliquer l'effet de certaines cartes qui le demandes
utilisateur	pouvoir ajouter un personnage dans mes deux expéditions en même temps	pouvoir plus simplement invoquer les personnages qui disposent de la capacité gigantic
utilisateur	pouvoir inverser mes deux expéditions	d'appliquer l'effet de certaines cartes qui le demandes

De plus, je suis en train de développer une nouvelle fonctionnalité qui n'est dès lors, pas encore achevée. Il s'agit du mode à deux smartphones. Plutôt que d'avoir un seul téléphone au milieu de la table, chaque joueur peut alors poser le sien à l'extrémité d'une expédition en mode portrait. Ce qui permettrait une visibilité accrue pour les deux joueurs.

Par la suite, j'aimerais également la rendre accessible hors ligne grâce à un Service Worker, afin de gérer les fonctionnalités hors ligne, le cache et les notifications push.

Aussi, il faudrait que j'achève la page de présentation de l'application. Elle devrait montrer l'application dans les grandes lignes, indiquer comment la télécharger et présenter un guide d'utilisation. Elle disposerait d'une notification de la dernière version et des ajouts apportés.

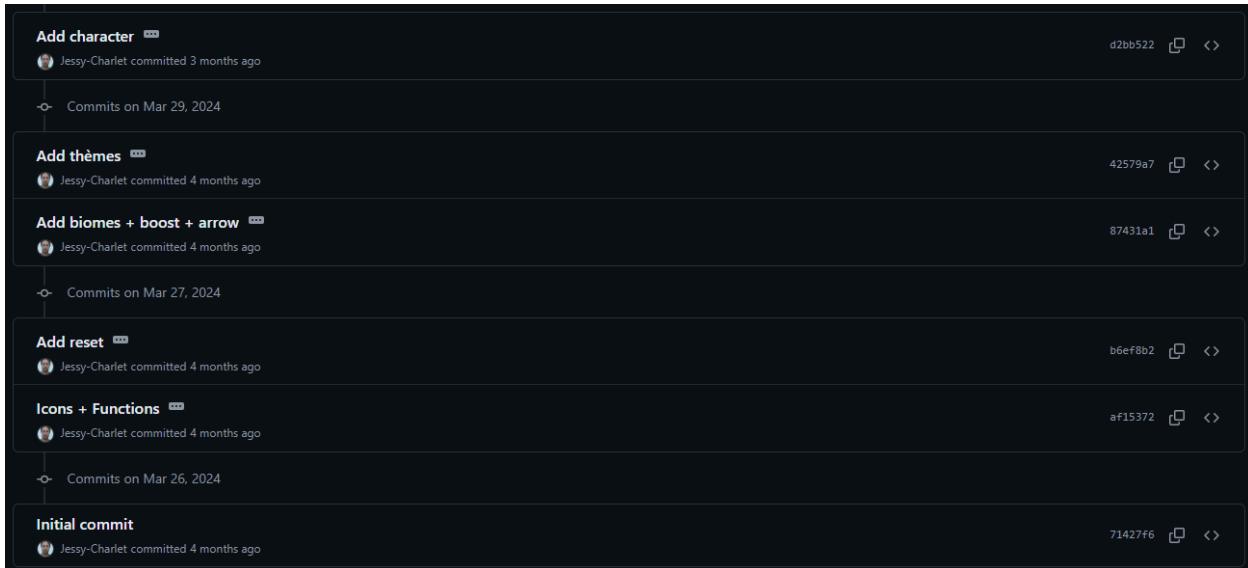
SPECIFICATIONS TECHNIQUES

Versioning

La gestion de version permet de conserver l'historique du code source et de travailler plus facilement en équipe. Même seul, il était très important de garder les versions précédentes du code, notamment en cas de gros problème avec une fonctionnalité suite à des modifications dans le code. Pour cette gestion, j'ai utilisé Git.

J'ai au départ utilisé GitHub puis avec le temps je suis passé directement en ligne de commande. Travaillant seul sur ce projet, je n'ai pas eu à utiliser de Pull Request afin de contrôler les merges de branches, ni même de Git Issues. Malgré tout, leur utilité va s'avérer nécessaire afin de ne pas oublier les mises à jour à faire ainsi que les retours des joueurs.

Les quelques commits que j'ai réalisé :



The screenshot shows a list of six commits from a GitHub repository:

- Add character** (3 months ago) - Commit d2bb522
- Add thèmes** (4 months ago) - Commit 42579a7
- Add biomes + boost + arrow** (4 months ago) - Commit 87431a1
- Add reset** (4 months ago) - Commit b6ef8b2
- Icons + Functions** (4 months ago) - Commit af15372
- Initial commit** (4 months ago) - Commit 71427f6

Each commit includes the author (Jessy-Charlet), the date (e.g., 3 months ago or 4 months ago), the commit hash, and standard GitHub actions icons (copy, diff, etc.). The commits are grouped by date: March 29, 2024; March 27, 2024; and March 26, 2024.

CHOIX TECHNOLOGIQUES

- **Front-End**

Le choix des technologies a été simple. Tout l'intérêt de ce projet était la réactivité, la modification du DOM (Document Object Model) sans rechargement lourd de la page. Exit donc le PHP et vive le JavaScript ! Un langage que j'ai commencé à apprendre une semaine avant de commencer ce projet personnel, mais qui à sa simple découverte m'a mis des étoiles dans les yeux !

Pour la structure bien évidemment j'ai utilisé le HTML5 (HyperText Markup Language) ainsi que le CSS3 (Cascading Style Sheets). Je n'ai pas utilisé Bootstrap ou d'autre librairies CSS car je trouve ça générique et un peu sans âme... et par dessus tout parce que j'aime profondément faire du CSS, ça me détend :P

- **Back-End**

Cette application n'a pas besoin de base de données pour fonctionner, elle ne dispose donc d'aucun back-end.

ACCESSIBILITÉ

Types d'appareils

Altered Counter est compatible avec les smartphones, les tablettes et les ordinateurs (même si les cas d'utilisation desktop seront infimes). En effet, j'ai développé une application responsive et je me suis assuré que l'affichage soit adapté en fonction du type de résolution.

PWA (Progressive Web App)

Une Progressive Web App (PWA) est une application web qui utilise les technologies modernes pour offrir une expérience utilisateur similaire à une application native. Elle peut fonctionner hors ligne, envoyer des notifications push, et être installée sur l'écran d'accueil d'un appareil sans passer par une boutique d'applications. Les PWA sont rapides, fiables, et réactives, adaptant leur interface à différents types de périphériques. Elles combinent les meilleures caractéristiques des applications web et mobiles pour une performance optimale.

Dès le début, je suis parti dans l'optique de développer cette application en PWA. Pour être utilisé par le plus grand nombre il fallait qu'on puisse l'installer sur l'écran d'accueil d'un mobile comme un application native, afin de pouvoir la lancer en un clic.

DÉPLOIEMENT

Le déploiement a été plutôt rapide, grâce à l'hébergement Plesk fournis par la plateforme, j'ai simplement créé un dossier altered-counter et déposé mes fichiers dedans. L'utilisation d'un logiciel FTP tel que FileZilla aurait pu être préférable, mais le système d'intégration de Plesk était suffisamment pratique et intuitif pour ne pas retenir cette option.

LE CODE

HTML

La structure html est assez simple :

```
1  <!DOCTYPE html>
2  <html lang="fr">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="author" content="Jessy Charlet">
7      <meta name="description" content="Compteur pour le jeu Altered TCG">
8      <meta name="viewport" content="width=device-width, height=device-height, initial-scale=1.0, user-scalable=no">
9
10     <link rel="preconnect" href="https://fonts.googleapis.com">
11     <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
12     <link href="https://fonts.googleapis.com/css2?family=Lilita+One&display=swap" rel="stylesheet">
13
14     <link id="css" rel="stylesheet" href="style.css">
15     <link rel="manifest" href="./altered.webmanifest" />
16     <link rel="shortcut icon" href="img/options.png" type="image/png">
17
18     <title>Altered counter</title>
19  </head>
```

- Le Doctype est obligatoire, il informe le navigateur de la version du HTML utilisée.
- Le Head quant à lui, contient les différentes balises meta et link qui ne sont pas affichées à l'utilisateur mais utiles pour les navigateurs et les moteurs de recherche.
- Les meta author et description parlent d'elles même, quand à la meta viewport, celle ci indique au navigateur (dans le cas présent) que la largeur et la hauteur de la page correspond à celle de l'écran, qu'il n'y a pas de zoom et que l'utilisateur de pourra pas zoomer manuellement.
- Les links permettent quant à eux de lier certains éléments à la page.

Nous pouvons voir les links fonts.googleapis qui permettent d'utiliser des polices d'écriture différentes de celles de base.

Le link stylesheet qui pointe vers la feuille de style CSS (celle-ci comporte une id afin

de pouvoir changer la feuille de style à la volée grâce au JavaScript). Le link du manifest qui est utilisé pour la PWA, et enfin le link shortcut icon qui permet de définir le favicon (icon affiché à côté du nom de la page). La balise title quant à elle indique évidemment le nom de la page.

```
21 <body>
22
23   <main>
24
25     <button id="options"></button>
26     <div id="aPlayerArrowRight" class="arrow"></div>
27     <div id="bPlayerArrowRight" class="arrow"></div>
28     <div id="aPlayerArrowLeft" class="arrow"></div>
29     <div id="bPlayerArrowLeft" class="arrow"></div>
30     <div id="hero" class="token"></div>
31     <div id="compagnon" class="token"></div>
32     <div class="road"></div>
```

Vient ensuite l'ouverture du body et celle du main qui contient tout le corps de notre application. On peut y apercevoir le bouton option, les quatres flèches indiquant la progression des joueurs (celles-ci seront masquées au chargement de la page par le JavaScript) ainsi que les deux jetons (hero et compagnon) et la route qui coupe le terrain en deux.

```

34      <!-- Zone de jeu du joueur b -->
35
36  <section id="bPlayerArea" class="playerArea backgroundWhite">
37
38    <div class="boost">
39      <button id="bPlayerUpRight"></button>
40      <button id="bPlayerAddRight"></button>
41      <button id="bPlayerDownRight"></button>
42    </div>
43    <div class="area">
44      <button id="bPlayerUpPlanteright" class="modify mUp">&lt;</button>
45      <button id="bPlayerUpTerreRight" class="modify mUp">&lt;</button>
46      <button id="bPlayerUpEauRight" class="modify mUp">&lt;</button>
47      <div id="bPlayerPlanteright" class="value plante">0</div>
48      <div id="bPlayerTerreRight" class="value terre">0</div>
49      <div id="bPlayerEauRight" class="value eau">0</div>
50      <button id="bPlayerDownPlanteRight" class="modify mDown">&gt;</button>
51      <button id="bPlayerDownTerreRight" class="modify mDown">&gt;</button>
52      <button id="bPlayerDownEauRight" class="modify mDown">&gt;</button>
53    </div>
54    <button id="bPlayerReset"></button>
55    <div class="area">
56      <button id="bPlayerUpPlanteLeft" class="modify mUp">&gt;</button>
57      <button id="bPlayerUpTerreLeft" class="modify mUp">&gt;</button>
58      <button id="bPlayerUpEauLeft" class="modify mUp">&gt;</button>
59      <div id="bPlayerPlanterLeft" class="value plante">0</div>
60      <div id="bPlayerTerreLeft" class="value terre">0</div>
61      <div id="bPlayerEauLeft" class="value eau">0</div>
62      <button id="bPlayerDownPlanteLeft" class="modify mDown">&gt;</button>
63      <button id="bPlayerDownTerreLeft" class="modify mDown">&gt;</button>
64      <button id="bPlayerDownEauLeft" class="modify mDown">&gt;</button>
65    </div>
66    <div class="boost">
67      <button id="bPlayerUpLeft"></button>
68      <button id="bPlayerAddLeft"></button>
69      <button id="bPlayerDownLeft"></button>
70    </div>
71
72  </section>

```

La zone du joueur B (identique à celle du joueur A, hormis les nom des id) est une section qui contient l'affichage des deux expéditions de ce joueur avec les boutons de modification dans la div area ainsi que le bouton reset au centre et les buttons de boost et d'ajout de personnage aux extrémités. Il est possible de remarquer la présence des attributs alt sur les images qui sont très importants pour l'accessibilité des personnes malvoyantes, ainsi que pour le SEO (optimisation pour les moteurs de recherche).

```

114 <section id="optionsWindow" class="blur">
115   <div class="frame option">
116     <button id="close"></button>
117     <h1>Options</h1>
118     <section id="bPlayerSettings">
119       <button id="bPlayerBlack" class="theme"></button>
120       <button id="bPlayerWhite" class="theme"></button>
121       <button id="bPlayerAxiom" class="theme"></button>
122       <button id="bPlayerBravos" class="theme"></button>
123       <button id="bPlayerLyra" class="theme"></button>
124       <button id="bPlayerMuna" class="theme"></button>
125       <button id="bPlayerOrdis" class="theme"></button>
126       <button id="bPlayerYzmir" class="theme"></button>
127     </section>
128     <section id="settings">
129       <h2>Mode solo</h2>
130       <h2>Mode duo</h2>
131       <a id="solo" href=".//home.html"></a>
132       <a id="duo" href=".//homeduo.html"></a>
133     </section>
134     <section id="aPlayerSettings">
135       <button id="aPlayerBlack" class="theme"></button>
136       <button id="aPlayerWhite" class="theme"></button>
137       <button id="aPlayerAxiom" class="theme"></button>
138       <button id="aPlayerBravos" class="theme"></button>
139       <button id="aPlayerLyra" class="theme"></button>
140       <button id="aPlayerMuna" class="theme"></button>
141       <button id="aPlayerOrdis" class="theme"></button>
142       <button id="aPlayerYzmir" class="theme"></button>
143     </section>
144   </div>
145 </section>
146
147 <section id="character" class="blur">
148
149 </section>

```

La section optionWindow quant à elle, affiche la modale des options et des thèmes. Cette dernière est masquée au chargement de la page et est affichée grâce au JavaScript. On peut également remarquer la section character qui est vide... celle-ci sera remplie avec le Javascript quand la modale sera appelée.

```

151   </main>
152   <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>
153   <script src="script.js"></script>
154 </body>
155
156 </html>

```

Enfin, je ferme le main, j'ajoute juste avant la fermeture du body le script me permettant d'utiliser la bibliothèque Jquery ainsi que celui de mon code JavaScript.

Je les place à cet endroit pour qu'ils soient chargés après le reste de mes éléments. J'aurais également pu les déclarer dans le head en y ajoutant la propriété defer.

CSS

Mon fichier CSS faisant 560 lignes, il est compliqué de tout réunir ici, l'idée étant de présenter seulement quelques exemples notables :

```
78  .arrow {  
79    padding: 0;  
80    margin: 0;  
81    position: absolute;  
82    z-index: 9;  
83    top: 50%;  
84    left: 50%;  
85    translate: -50% -50%;  
86    border-radius: 10%;  
87  
88    img {  
89      height: 6vh;  
90    }  
91  }  
92  
93  
94  #aPlayerArrowRight img {  
95    translate: 15vw 4vh;  
96    animation: infinite alternate 2s slideARight;  
97  }  
98  
99  @keyframes slideARight {  
100   from {  
101     translate: 15vw 4vh;  
102   }  
103  
104   to {  
105     translate: 13vw 4vh;  
106   }  
107 }
```

Ce bloc gère les flèches indiquant la progression des joueurs sur les différentes expéditions.

On peut voir la class .arrow qui contient les propriétés communes à toutes les flèches

- les marges internes et externes à 0
- son z-index pour la faire hauteur sur l'axe Z (profondeur) par rapport aux autres éléments qui sont sortis du flux.
- sa position en absolute qui la sort du flux justement pour la positionner au centre (vertical et horizontal) par rapport à l'élément main qui est lui en position relative.
- ses enfants img qui ont une hauteur de 6vh.

J'utilise les vh et vw (viewport width et height) qui correspondent au pourcentage de la largeur et hauteur de l'écran afin de m'adapter à toutes les résolutions.

Chaque flèche à ensuite un id qui permet de la cibler avec JavaScript et de lui attribuer des spécifications (par exemple le translate qui va la placer au bon endroit ainsi qu'une animation). On peut d'ailleurs voir sur cet exemple que l'animation va se jouer en boucle en aller/retour sur un intervalle de 2s et que la nature de celle-ci va être précisée dans le @keyframes slideARight.

```

154  /*** Players areas -----
155
156 .playerArea {
157   box-sizing: border-box;
158   width: 100vw;
159   height: 50vh;
160   display: grid;
161   grid-template-columns: 10vw 1fr 10vw 1fr 10vw;
162   gap: 2vw;
163 }
164
165 #bPlayerArea {
166   transform: rotate(180deg);
167 }
168
169
170
171 /*** Zone boosts */
172
173 .boost {
174   box-sizing: border-box;
175   display: grid;
176   grid-template-rows: repeat(3, 1fr);
177   grid-template-columns: 1fr;
178   gap: 1vh;
179   padding: 5vh 0 0 0;
180
181   button {
182     border: none;
183     background-color: transparent;
184
185     img {
186       height: 10vh;
187       transition: all 200ms;
188     }
189   }
190
191   button:hover img {
192     height: 12vh;
193   }
194 }

```

Sur la zone playerArea on peut remarquer l'utilisation de box-sizing avec la valeur border-box, qui permet de donner la largeur et la hauteur précise de la box, padding et border compris, ce qui est très pratique pour ne pas calculer ces valeurs à chaque modification du design.

On peut également voir l'utilisation de display grid, que j'aime beaucoup. Il permet une mise en page propre, simple et modulaire sans passer par des tableaux archaïques.

La zone de jeux du joueur B étant dans son sens, je l'ai retourné grâce à transform: rotate.

Dans boost on peut apercevoir un tout petit exemple de la syntaxe en cascade très pratique qui permet de cibler des éléments enfant directement dans leurs parents. Ce qui est beaucoup plus lisible et court que d'écrire : .boost button img { }

JavaScript / Jquery

```

1 $(document).ready(function () {
2
3
4   $("#aPlayerArrowRight").hide();
5   $("#aPlayerArrowLeft").hide();
6   $("#bPlayerArrowRight").hide();
7   $("#bPlayerArrowLeft").hide();
8   $("#optionsWindow").hide();
9   $("#character").hide();

```

Pour commencer, tout mon JS est encapsulé dans la fonction `$(document).ready()`, ce qui garantit que le code à l'intérieur de cette fonction ne s'exécute que lorsque le DOM (Document Object Model) est entièrement chargé. Cela signifie que toutes les balises HTML ont été analysées et que les éléments sont accessibles

pour le script.

Ensuite je cache de l'affichage tout ce qui ne doit pas être visible (les flèches, la modale optionWindow et la modale character.

```
42     /*** Diminution des valeurs */
43
44     function Down(elementClick, cible) {
45         $(elementClick).on("click", function () {
46             let value = $(cible).text();
47             if (value > 0) {
48                 value--;
49             } else {
50                 value = 0;
51             }
52             $(cible).text(value)
53             HighterAll();
54             ArrowAll();
55         })
56     }
57     Down("#aPlayerDownPlanteLeft", "#aPlayerPlanteLeft");
58     Down("#aPlayerDownTerreLeft", "#aPlayerTerreLeft");
59     Down("#aPlayerDownEauLeft", "#aPlayerEauLeft");
60
61     Down("#aPlayerDownPlanteRight", "#aPlayerPlanteRight");
62     Down("#aPlayerDownTerreRight", "#aPlayerTerreRight");
63     Down("#aPlayerDownEauRight", "#aPlayerEauRight");
64
65     Down("#bPlayerDownPlanteLeft", "#bPlayerPlanteLeft");
66     Down("#bPlayerDownTerreLeft", "#bPlayerTerreLeft");
67     Down("#bPlayerDownEauLeft", "#bPlayerEauLeft");
68
69     Down("#bPlayerDownPlanteRight", "#bPlayerPlanteRight");
70     Down("#bPlayerDownTerreRight", "#bPlayerTerreRight");
71     Down("#bPlayerDownEauRight", "#bPlayerEauRight");
```

Aperçu de la fonction Down() qui diminue les valeurs lorsque l'on clique sur le bouton correspondant. Il suffit de lui passer en argument le bouton (elementClick) ainsi que la cible dont on souhaite diminuer la valeur (cible).

Je récupère la valeur du texte de l'élément cible que je place dans value, puis si celle-ci est supérieure à 0, alors je l'incrémenter de 1. Le texte de la cible est ensuite changé par la nouvelle value. Pour finir j'appelle les fonctions HighterAll() et ArrowAll() que je vais décrire plus bas.

A noter que je n'utilise pas var pour déclarer la variable value mais plutôt let, ce qui fait partie des bonne pratiques JS. La différence en terme d'utilisation s'effectue surtout au niveau du scope (la portée de la variable) avec var elle à une portée fonctionnelle (c'est à dire qu'elle est accessible uniquement dans la fonction dans laquelle elle a été déclarée)

alors qu'avec let elle a une porté de bloc (elle est accessible uniquement dans le bloc {} dans lequel elle a été déclarée).

```
173     /***** Gestion des victoires ****/
174
175     /*** Detection et changement des biomes les plus hauts*/
176     function Higter(biome, zone) {
177         let aPlayer = $("#aPlayer" + biome + zone).text();
178         let bPlayer = $("#bPlayer" + biome + zone).text();
179         if (parseInt(aPlayer) > parseInt(bPlayer)) {
180             $("#aPlayer" + biome + zone).addClass("higter");
181             $("#bPlayer" + biome + zone).removeClass("higter");
182         } else if (parseInt(aPlayer) < parseInt(bPlayer)) {
183             $("#aPlayer" + biome + zone).removeClass("higter");
184             $("#bPlayer" + biome + zone).addClass("higter");
185         } else {
186             $("#aPlayer" + biome + zone).removeClass("higter");
187             $("#bPlayer" + biome + zone).removeClass("higter");
188         }
189     }
190     function HigterAll() {
191         Higter("Plante", "Left");
192         Higter("Plante", "Right");
193         Higter("Eau", "Left");
194         Higter("Eau", "Right");
195         Higter("Terre", "Left");
196         Higter("Terre", "Right");
197     }
198
199     function Arrow(player, zone) {
200         if ($("#" + player + "Terre" + zone + ":not(.off).higter").text()
201             || $("#" + player + "Plante" + zone + ":not(.off).higter").text()
202             || $("#" + player + "Eau" + zone + ":not(.off).higter").text()) {
203             $("#" + player + "Arrow" + zone).show();
204         } else {
205             $("#" + player + "Arrow" + zone).hide();
206         }
207     }
208
209     function ArrowAll() {
210         Arrow("aPlayer", "Left");
211         Arrow("aPlayer", "Right");
212         Arrow("bPlayer", "Left");
213         Arrow("bPlayer", "Right");
214     }
```

La fonction HigterAll() est simplement un ensemble d'appel à la fonction Higter() avec en paramètre les différents biomes pour chaque expédition (gauche et droite).

Regardons de plus près la fonction Higter() :

Pour chaque joueur, on passe dans une variable le texte de l'élément qui correspond au biome ciblé dans la zone ciblé de ce joueur.

Grâce à la fonction parseInt() qui analyse une chaîne de caractère fournie en argument et renvoie un entier, je compare la valeur des deux textes.

- Si le joueur A est plus grand, l'élément ciblé gagne la classe highter et celui du joueur B perd cette classe.
- Si le joueur B est plus grand, l'élément ciblé gagne la classe highter et celui du joueur A perd cette classe.
- Dans les autres cas (égalité) ils perdent tous les deux la classe highter.

La fonction ArrowAll() fonctionne de la même manière que HigterHall(), un ensemble d'appels à la fonction Arrow() avec en argument un combo joueur + côté.

Pour une de ces combinaisons joueur + côté, la fonction Arrow() va faire plusieurs vérifications :

- Si son biome terre n'a pas la class off et possède la class highter
- OU si son biome plante n'a pas la class off et possède la class highter
- OU si son biome eau n'a pas la class off et possède la class highter

Alors si un de ces cas est vrai, on affiche la flèche d'avancement sur le côté ciblé du joueur ciblé. Dans le cas contraire, on retire la flèche de l'affichage.

Le sélecteur :not() en jQuery est utilisé pour sélectionner des éléments qui ne correspondent pas à un certain sélecteur. Dans le cas de :not(.off), cela signifie que tous les éléments qui n'ont pas la classe off mais qui ont la classe highter seront sélectionnés.

```

217 //***** Ajouter un personnage ****/
218
219 //*** Ouverture et mécanique des quartes menus*/
220 function addCharacter(zone, player) {
221     $("#" + player + "Add" + zone).on("click", function () {
222         let sens = "";
223         if (player == "bPlayer") {
224             sens = "rotate";
225         }
226         $("#" + player + "character").html("<div class='frame " + sens + " " + player + " " + zone + "'>" +
227             "<div class='area">" +
228             "<button id='characterUpPlante' class='modify mUp'>&amplt/button>" +
229             "<button id='characterUpTerre' class='modify mUp'>&amplt/button>" +
230             "<button id='characterUpEau' class='modify mUp'>&amplt/button>" +
231             "<div id='characterPlante' class='value plante'>0</div>" +
232             "<div id='characterTerre' class='value terre'>0</div>" +
233             "<div id='characterEau' class='value eau'>0</div>" +
234             "<button id='characterDownPlante' class='modify mDown'>&lt;/button>" +
235             "<button id='characterDownTerre' class='modify mDown'>&lt;/button>" +
236             "<button id='characterDownEau' class='modify mDown'>&lt;/button>" +
237             "</div>" +
238             "<div class='framebutton'>" +
239             "<button id='characterClose' class='close'><img src='./img/close.png' alt='Fermer' /></button>" +
240             "<button id='"+ player + "CharacterCheck' + zone + "' class='check'><img src='./img/check.png' alt='Valider' /></button>" +
241             "</div>" +
242             "</div>");
243         $("#" + player + "character").fadeIn("fast");
244         Up("#characterUpPlante", "#characterPlante");
245         Up("#characterUpTerre", "#characterTerre");
246         Up("#characterUpEau", "#characterEau");
247         Down("#characterDownPlante", "#characterPlante");
248         Down("#characterDownTerre", "#characterTerre");
249         Down("#characterDownEau", "#characterEau");
250         /** Fermeture des quartes menus*/
251         /** Validation des stats de biomes */
252         $("#" + player + "CharacterCheck" + zone).on("click", function () {
253             console.log(player);
254             let valuePlante = parseInt($("#" + player + "Plante" + zone).text());
255             let valueTerre = parseInt($("#" + player + "Terre" + zone).text());
256             let valueEau = parseInt($("#" + player + "Eau" + zone).text());
257             let characterPlante = parseInt($("#characterPlante").text());
258             let characterTerre = parseInt($("#characterTerre").text());
259             let characterEau = parseInt($("#characterEau").text());
260             valuePlante = valuePlante + characterPlante;
261             valueTerre = valueTerre + characterTerre;
262             valueEau = valueEau + characterEau;
263             $("#" + player + "Plante" + zone).text(valuePlante);
264             $("#" + player + "Terre" + zone).text(valueTerre);
265             $("#" + player + "Eau" + zone).text(valueEau);
266             HighterAll();
267             ArrowAll();
268             $("#" + player + "character").fadeOut("fast");
269         })
270         /** Fermeture */
271         $("#" + player + "characterClose").on("click", function () {
272             $("#" + player + "character").fadeOut("fast");
273         })
274     })
275 }
276
277 addCharacter("Right", "aPlayer");
278 addCharacter("Left", "aPlayer");
279 addCharacter("Right", "bPlayer");
280 addCharacter("Left", "bPlayer");

```

La fonction addCharacter() prend en argument un côté et un joueur.

Au clic sur le bouton add de ce combo expédition + joueur :

- Je déclare une variable sens dont la valeur est une chaîne de caractères vide.
- Si le joueur est le joueur B, alors la valeur de sens devient rotate
- Je sélectionne ensuite la balise section dont l'id est character (celle que l'on a vu

dans la partie HTML) et je remplace tout son contenu HTML par celui de la modale en lui donnant les classes qui correspondent au joueur à sa zone et au sens d'affichage.

- Je fais ensuite apparaître la modale qui était invisible.
- Je fais appel aux fonctions Up et Down pour les boutons de cette modale.
- Je crée une fonction anonyme pour le bouton de validation qui va récupérer comme on l'a vu précédemment les différentes valeurs modifiées afin de les mettre à jour, et je cache à nouveau la modale.

J'aurais pu utiliser les balises HTML template dans le HTML afin de n'appeler que la fonction `.cloneNode()` comme je l'ai fait d'autres projets par la suite (notamment mon projet en stage) mais en l'absence de connaissance relative à cette technique à l'époque et étant donné qu'il y a des variables à intégrer dans le HTML, refactoriser cette partie avec l'utilisation de template + `cloneNode()` ne serait pas spécialement plus court en terme de ligne et pas plus rapide en terme d'exécution.

```

285  **** Menu des options ****
286
287  *** Ouverture du menu ***
288  $("#options").on("click", function () {
289      $("#optionsWindow").fadeIn("fast");
290  })
291
292  *** Fermeture du menu ***
293  $("#close").on("click", function () {
294      $("#optionsWindow").fadeOut("fast");
295  })
296
297  *** Changement de couleur des themes ***
298  function Background(player, color) {
299      $("#" + player + color).on("click", function () {
300          $("#" + player + "Area").removeClass();
301          $("#" + player + "Area").addClass("playerArea background" + color);
302
303      })
304  }
305  Background("aPlayer", "Black");
306  Background("aPlayer", "White");
307  Background("aPlayer", "Axiom");
308  Background("aPlayer", "Bravos");
309  Background("aPlayer", "Lyra");
310  Background("aPlayer", "Muna");
311  Background("aPlayer", "Ordis");
312  Background("aPlayer", "Yzmir");
313  Background("bPlayer", "Black");
314  Background("bPlayer", "White");
315  Background("bPlayer", "Axiom");
316  Background("bPlayer", "Bravos");
317  Background("bPlayer", "Lyra");
318  Background("bPlayer", "Muna");
319  Background("bPlayer", "Ordis");
320  Background("bPlayer", "Yzmir");

```

Voici l'ensemble des fonctions gérant le menu option :

L'ouverture et la fermeture restent classiques, au clic sur un élément, il en est affiché ou caché un autre.

La fonction Background() qui prend en paramètre un joueur et une couleur est assez simple elle aussi. Au clic sur l'élément dont l'id correspond au couple joueur + couleur, on retire toutes les classes de l'arène du joueur puis on lui ajoute ses classes de base plus la classe de la couleur choisie.

Webmanifest

Le fichier .webmanifest, est écrit en format JSON (JavaScript Object Notation). Ce format est léger et facile à lire et à écrire, ce qui le rend idéal pour la configuration des Progressive Web Apps (PWA). Le webmanifest contient des informations essentielles sur l'application, telles que son nom, ses icônes, son thème et la configuration de l'écran d'accueil.

```
1  {
2      "name": "Altered companion app",
3      "short_name": "Altered Counter",
4      "description": "La 'companion app' parfaite pour vos parties d'Altered TCG",
5      "start_url": "./home.html",
6      "display": "fullscreen",
7      "lang": "fr",
8      "background_color": "black",
9      "theme_color": "black",
10     "orientation": "landscape-primary",
11     "icons": [
12         {
13             "src": "img/logo_144.png",
14             "type": "image/png",
15             "sizes": "144x144"
16         },
17         {
18             "src": "img/logo_512.png",
19             "type": "image/png",
20             "sizes": "512x512"
21         },
22         {
23             "src": "img/logo_192.png",
24             "type": "image/png",
25             "sizes": "192x192"
26         }
27     ],
28     "screenshots": [
29         {
30             "src": "img/screen_bw.jpg",
31             "type": "image/jpg",
32             "sizes": "1920x944",
33             "form_factor": "wide"
34         },
35         {
36             "src": "img/screen_lo.jpg",
37             "type": "image/jpg",
38             "sizes": "887x944",
39             "form_factor": "narrow"
40         }
41     ]
42 }
```

Par exemple le "display": "fullscreen" permet de s'affranchir des tous les éléments d'affichage du navigateur afin de se rapprocher au mieux d'une application native. Le tableau de screenshots quant à lui affiche des captures d'écran de l'application au moment de l'installation de celle-ci.