Partie de Eva:

* **Site** :

Pour accompagner notre projet, nous devons réaliser un site Internet. Etant curieuse et ayant soif d’apprendre, j’ai souhaité m’occuper de faire le site à partir de rien sans patron ou espace de travail. Il a fallu s’instruire alors sur sa réalisation ainsi que sur les langages du web. J’ai suivi de nombreux tutoriels pour essayer de comprendre comment cela fonctionne. Les sites HTML Références et CSS Références m’ont beaucoup aidé car chaque élément est expliqué et possède un exemple de son utilisation.

Je n’ai donc utilisé que le HTML et le CSS sur un l’IDE Visual Studio Code pour la réalisation. HTML est un langage de balisage qui est utilisé pour la description d’une page web. CSS est un langage qui définit la présentation d’un document écrit en HTML.

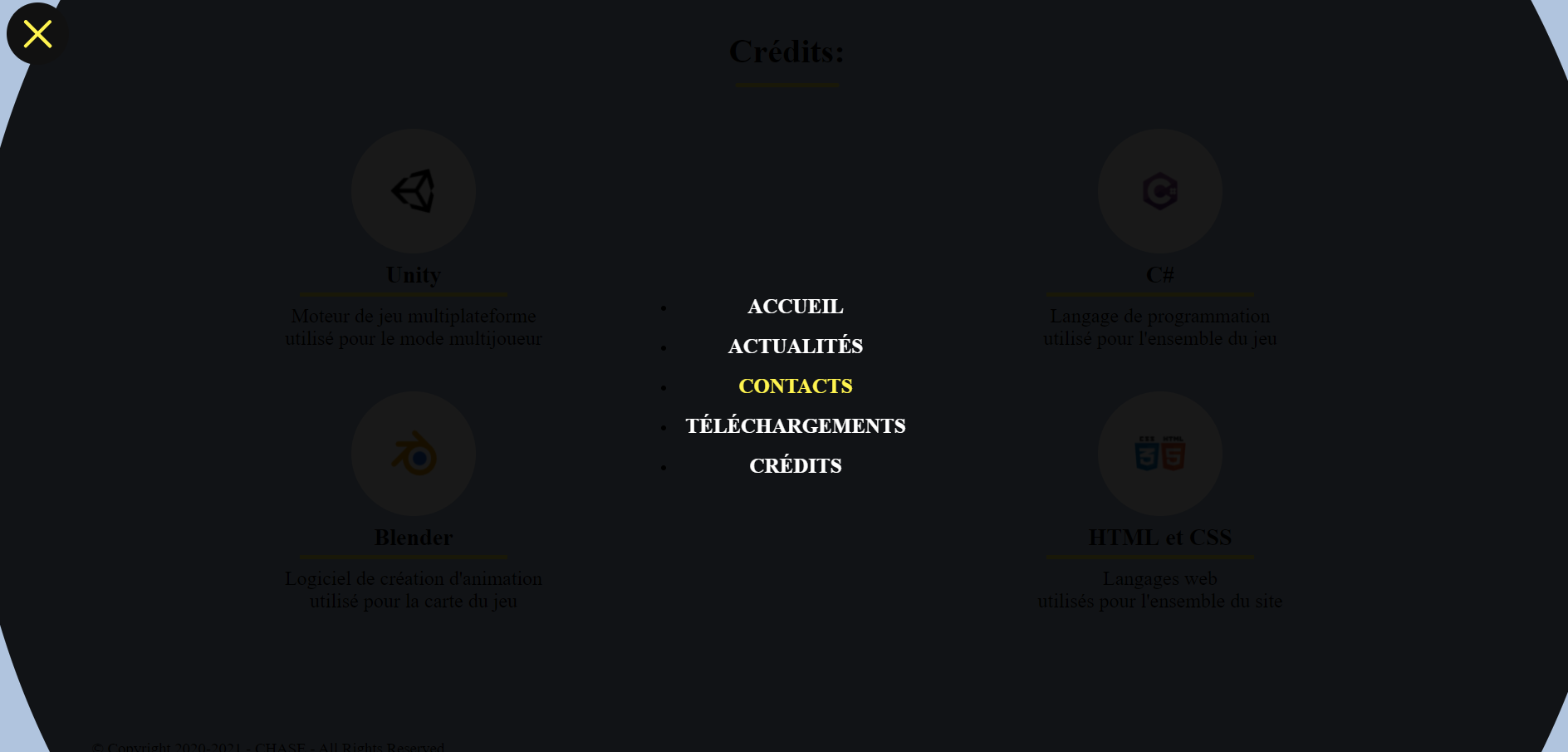
Le site est constitué de différentes pages, chacune associée à une catégorie telle que : accueil, actualités, contacts, téléchargements ou crédits. Toutes sont reliées par un menu appelé « menu hamburger ». J’ai été surprise par le temps de la réalisation de l’ensemble. Je n’ai pas eu beaucoup de difficultés car le temps consacré à l’apprentissage a été conséquent.

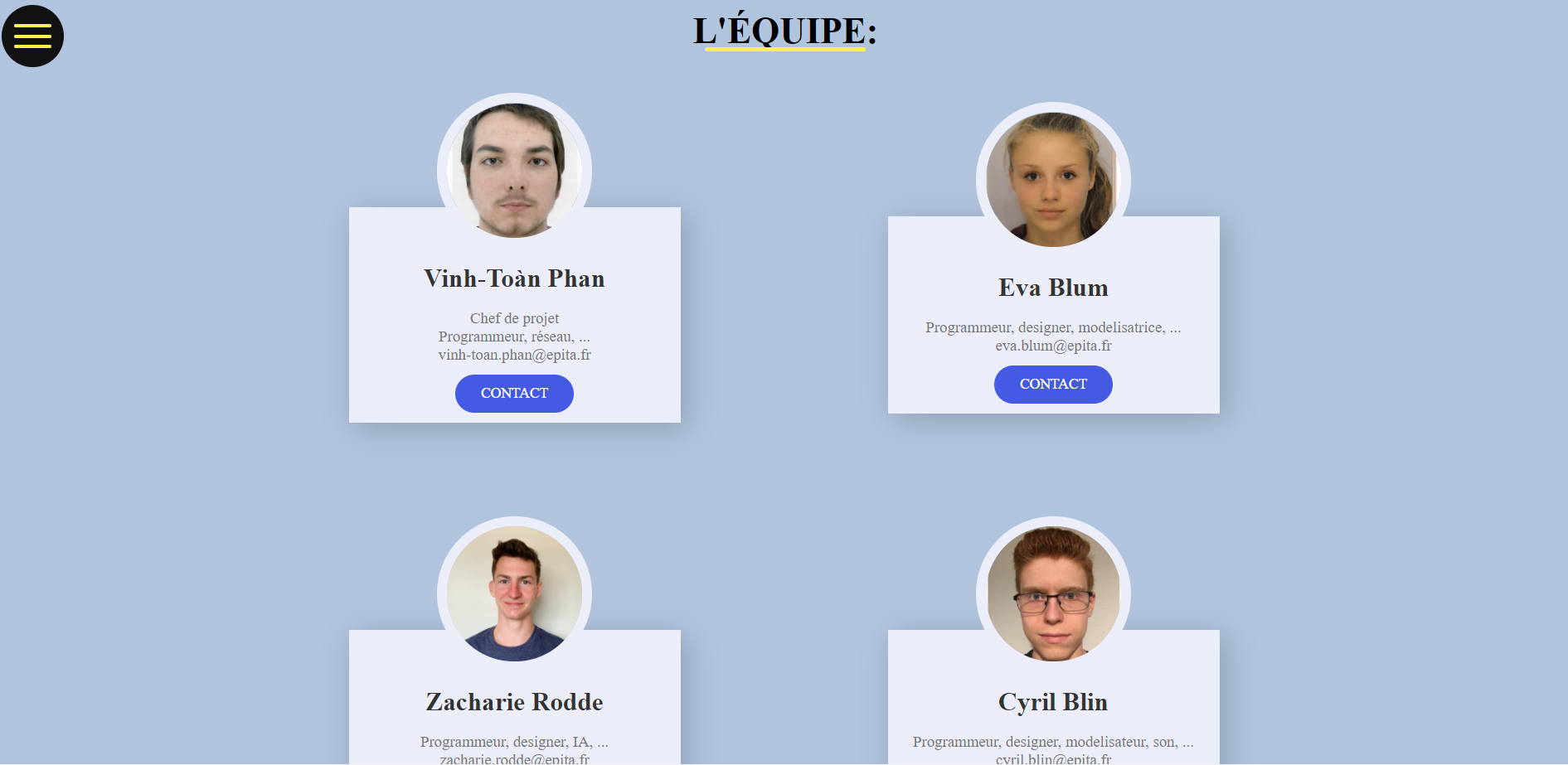
Voici les différentes maquettes du site élaborées auparavant.

* Maquettes : (photos mises dans discord)

Une fois les idées claires, place à la réalisation. Voici une ébauche du résultat après quelques modifications par rapport à la première soutenance :







* Publication :

Pour héberger le site, j’ai choisi d’utiliser GitHub Pages qui permet de transformer un projet en hébergement pour site web. Après quelques manipulations, le site web est visible par tous.

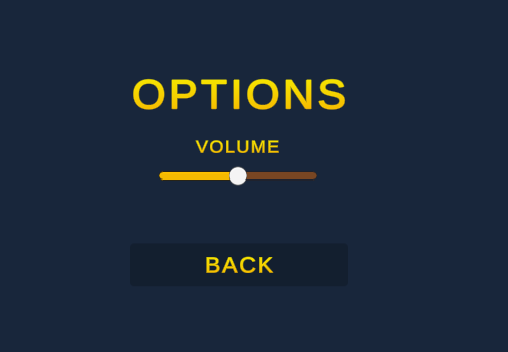
* Menu :

Pour la première fois dans ce projet, nous avons créé un menu principal qui par la suite sera bien entendu amélioré. Créer et implémenter cette ébauche de menu principal a été très facile et rapide grâce aux tutoriels trouvés. On retrouve les boutons principaux : jouer, les options et quitter le jeu.

Les boutons « PLAY » et « QUIT » ont une liaison vers un script nommé « SceneLoader » contenant une fonction pour chaque. Le bouton « PLAY » va lancer la fonction *LoadGame()* qui va charger et lancer simplement la scène pour jouer tout seul pour l’instant. La fonction *QuitGame()* va tout simplement quitter le jeu actuel.



Le bouton « OPTIONS » contient le réglage du volume de la musique du jeu. Pour voir les options proposées, il a fallu appliquer des fonctions. Plus précisément, lorsque l’utilisateur clique sur le bouton « OPTIONS » ou « BACK » c’est tout simplement une fonction qui nous permet de passer de « l’écran principal » à « l’écran » pour voir les options.

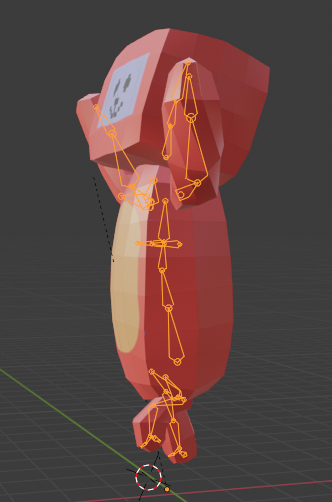


Pour la prochaine soutenance, nous avons pour ambition de faire un fond animé. Il y aura une caméra faisant le tour de la carte.

* Animations :

Lors de la première soutenance, nous avions réussi à créer des animations sur un personnage telles que la course, le saut, et la respiration lorsque le personnage reste statique. Cependant, l’implémentation sur Unity n’avait pas été un grand succès. Le personnage est arrivé déformé et manquait des animations importantes notamment pour ses jambes. De plus, le personnage implémenté n’était pas celui souhaité et pensé au départ. Ainsi, pour cette soutenance, nous avions du travail.

Cette partie animation du personnage nous a permis de comprendre l’importance de ce projet de groupe. En effet, nous avons tous réfléchis sur le problème. Le mouvement d’un bras entrainait directement l’autre. Nous avons finalement compris que le problème était lié au modèle de base. Nous souhaitions à tout prix garder le personnage de départ et avoir des animations faites par nous-même et non importées depuis un site connu appelé Mixamo. Par la suite, j’ai pu refaire les animations sur le bon personnage avec la texture correspondante.



Après avoir créer les animations sur le logiciel de modélisation Blender, il faut les exporter dans le bon format (.fbx) pour les récupérer sur le logiciel de développement Unity. Cependant, à nouveau les animations n’étaient pas bien récupérées et ne fonctionnaient pas. J’avais fait les animations sur un seul fichier donc le problème venait des transitions entre chaque. C’est alors dans la partie animator dans Unity que nous avons réussi à définir séparément les animations pour le saut, la course et enfin la respiration.