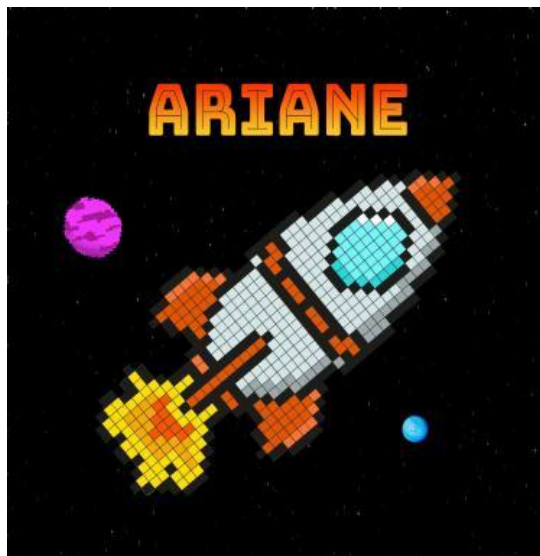




ARIANE

Premier rapport de soutenance, Mars 2021



Groupe CLTN:

- MANONGO Claude Cédric (chef du projet *Ariane*)
- CARION-VIGNAUD Titouan
- HASS Léo
- NDE Daniel Nolan Nathanael

Table des matières:

1/ Introduction: Page 3

-Plan de notre rapport

2/ Avancement du projet, planning et prévision futures: Page 4

→ Editeur 2D/Graphisme	Page 5
→ Gestion du Personnage	Page 7
→ L'interface	Page 10
→ Animateur	Page 14
→ Site Internet	Page 16
→ Ingénieur son et réseau	Page 20

3/ Conclusion: Page 21

1-INTRODUCTION:

En vue de réaliser notre projet de Semestre 2, nous avons créé notre groupe pour mettre en commun nos différentes compétences, nos différentes idées, nos différentes connaissances acquises en cours et pendant les Travaux Pratiques.

Nous nous sommes très vite mis d'accord sur l'idée de faire un jeu vidéo de type platformer 2D. Le jeu se nomme *Ariane* comme le nom du projet en référence au domaine aérospatial et la mythologie grecque.

De plus nous avons trouvé original de faire un jeu où le personnage principal est une femme car pour la plupart des platformer 2D ou même des jeux en général, le personnage principal est un homme ou un garçon.

Notre rapport de projet est composé de deux parties:

La première partie montrera les différents avancements de chaque étape composant notre projet tout en les comparant avec les différents objectifs que nous nous étions fixés pour cette première soutenance.

Ensuite, nous enchaînerons sur les prévisions futures pour notre projet. Nous aborderons les choses développées pour atteindre nos objectifs prévus pour la soutenance intermédiaire.

Enfin, nous terminerons ce rapport par une conclusion sur l'avancement général de ce projet, sur les choses potentielles à améliorer et sur l'état d'esprit général du groupe.

2-AVANCEMENT DU PROJET, PLANNING ET PRÉVISIONS FUTURES :

Tout d’abord voici notre planning et les tâches assignées aux différents membres du groupe :

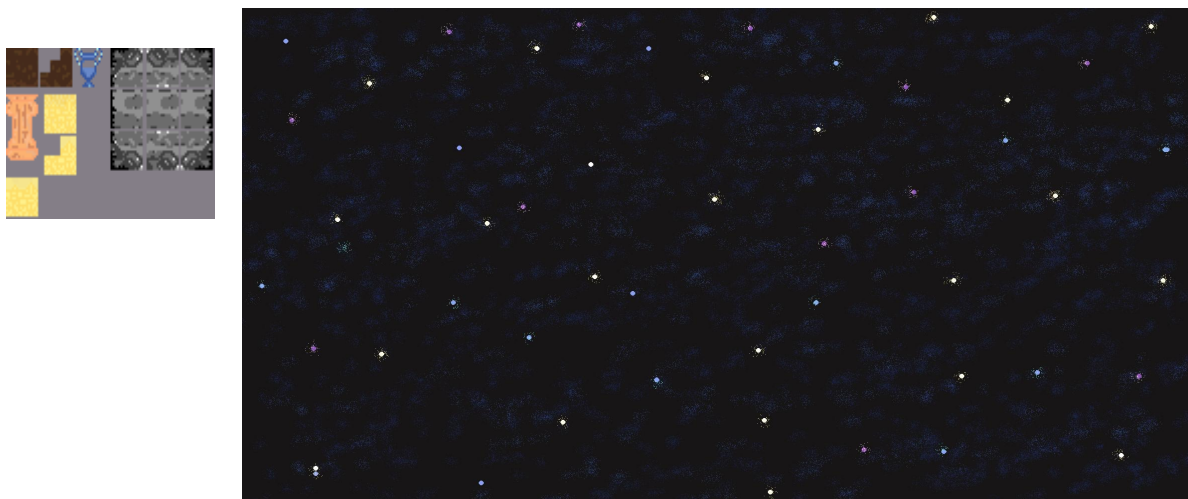
RESPONSABLE	SUPPLEANT	TACHES
NATHANAEL	Léo	Site internet
CLAUDE	Titouan	Editeur 2D/graphiste
TITOUAN	Léo	Animateur
TITOUAN	Nathanael	Ingé son
CLAUDE	Léo	Gestionnaire physique
TITOUAN	Nathanael	Scénariste
CLAUDE	Nathanael	Gestion Perso principal
LEO	Titouan	Gestion Perso mob
CLAUDE	Léo	Gestion Perso Boss
CLAUDE	Nathanael	Gestion Obstacle
TITOUAN	Claude	Interface
LEO	Nathanael	Réseau

	Première soutenance	Deuxième soutenance	Troisième soutenance
Site internet	50%	100%	100%
Editeur 2D/graphiste	20%	50%	100%
Animateur	20%	50%	100%
Ingé son	0%	10%	100%
Gestion perso	40%	60%	100%
Interface	40%	60%	100%
Réseau	0%	20%	100%

Edition 2D et graphismes:

Pour construire nos niveaux, nous avons dessiné du pixel art pour obtenir nos tilemaps, ce qui nous permettra aisément d'alterner entre les blocs comme on le souhaite.

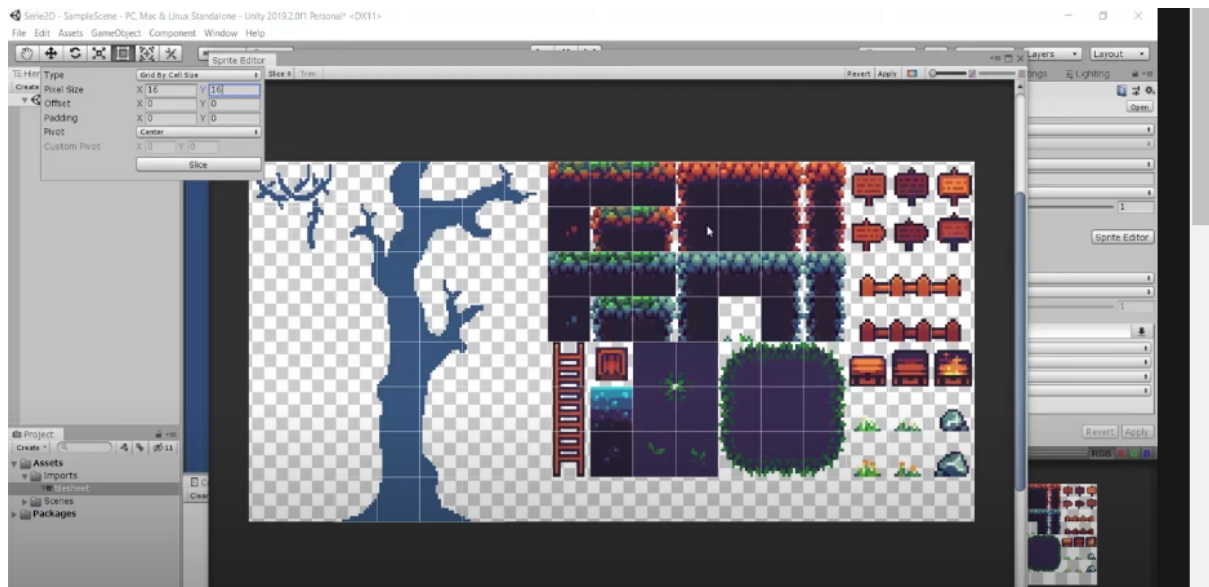
La première tilemap, ci-dessous, est celle du premier niveau. Ce premier niveau basé sur la Grèce Antique se passe dans l'espace, on y retrouve donc des escaliers, des piliers ainsi que des trophées. Le ciel est un fond étoilé.



Voici la première Tilemap du jeu pour le premier niveau ainsi que le fond étoilé faisant office de ciel.

Les logiciels tels que GIMP nous auraient permis d'atteindre le même résultat mais nous avons finalement décidé d'utiliser "Aseprite". Ce logiciel nous a permis de gérer plusieurs blocs à la fois, d'avoir une vue plus large de comment notre travail serait représenté dans un jeu. Nous avons appris à maîtriser ses bases grâce à de multiples tutoriels.

Avec Unity nous avons pu expérimenter les prémices de la conception d'un niveau grâce au outil mis à notre disposition. En important la tilemap dans le logiciel, on peut alors la diviser en plusieurs cellules de la taille des tiles, ce qui nous permettra de les disposer dans l'ordre et les places que nous souhaitons dans une des scènes que nous créons dans Unity.



Concernant l'avancement de la partie Edition 2D/Graphisme, la tilemap pour le premier niveau est déjà faite, nous sommes donc en avance par rapport au pourcentage fourni dans le cahier des charges, cependant il faudra accélérer la charge de travail par la suite pour atteindre les pourcentages que nous avons établi dans le cahier des charges. Pour la prochaine fois, on devra finir la structure du premier niveau qui fera office de tutoriel, permettant au joueur de maîtriser les bases du jeu ainsi que ses mécaniques.

Gestion du personnage:

Pour mettre en place les mouvements de notre personnage, nous avons premièrement choisi sur Internet un sprite sur lequel nous appliquerons les différents mouvements que nous souhaitons réaliser. Nous remplacerons ce sprite de test par le nôtre pour la soutenance intermédiaire.

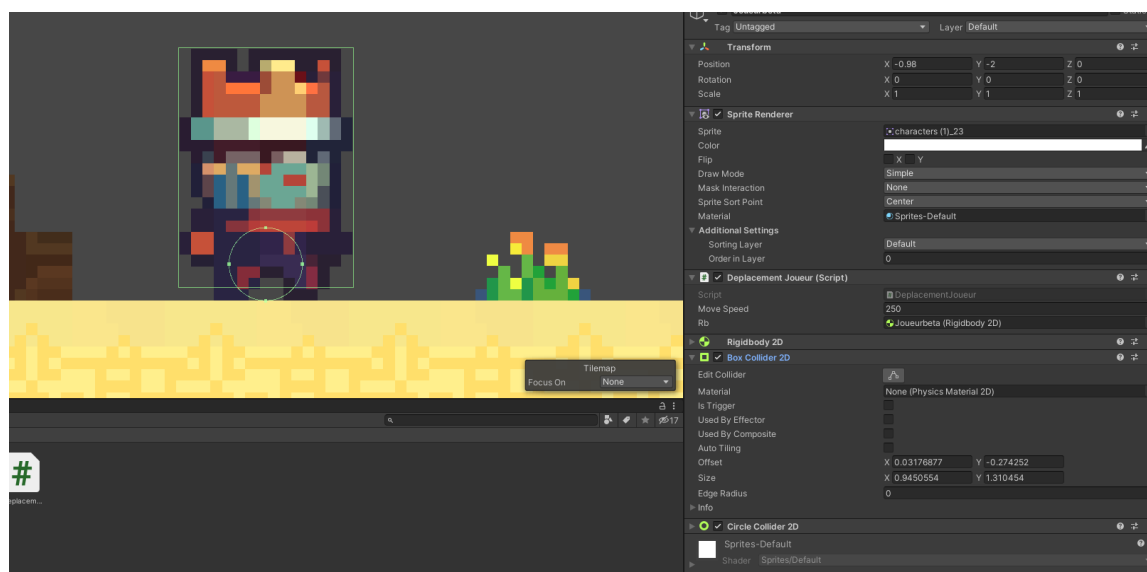
Nous avons utilisé Unity que nous avons lié à Visual Studio pour pouvoir programmer les déplacements du personnage.

Pour cette première soutenance, nous nous sommes penchés sur les mouvements de déplacements horizontaux et de sauts du personnage.

Pour le mouvement de déplacement horizontal, nous avons choisi d'effectuer celui-ci avec l'aide d'un rigidbody 2D pour que notre personnage soit affecté par la gravité et cela va nous permettre de lui ajouter une force pour le déplacer dans une direction voulu.

Nous avons donc en premier ajouté le rigidbody 2D au personnage tout en gelant la rotation sur l'axe Z ce qui va permettre au personnage de ne pas tomber vers l'avant ou l'arrière.

Puis nous avons ajouté un box collader 2D représentant la boîte de collision de notre personnage, pour qu'il puisse entrer en collision avec les sprites du niveau qui eux aussi possèdent un collader 2D spécifique aux tilemap. Nous avons vite remarqué que le personnage restait bloqué entre son collader et celui des sprites du niveau. C'est donc pour cela que nous avons aussi ajouté un circle collader entre la pointe des pieds de notre personnage.



Ici le box collider est le carré vert autour du personnage et le circle collider le cercle vert.

Ensuite nous avons commencé à programmer son déplacement horizontal à l'aide de Visual studio. Voir image ci-dessous.

```
public class DeplacementJoueur : MonoBehaviour
{
    public float moveSpeed; //vitesse de déplacement
    public Rigidbody2D rb; //élément faisant référence au rigidbody de notre personnage
    private Vector3 velocity = Vector3.zero; //vitesse

    // Message Unity | 0 références
    void FixedUpdate() //fixed update car on utilise de la physique
    {
        float horizontalMouvement = Input.GetAxis("Horizontal") * moveSpeed * Time.deltaTime; //calculer la vitesse du personnage sur
        //l'axe horizontal, le delta time permet de récupérer la vitesse au fil du temps

        MovePlayer(horizontalMouvement); //pour envoyer le mouvement au rigidbody pour déplacer le personnage
    }

    // 1 référence
    void MovePlayer (float _horizontalMouvement) //pour calculer quand le personnage doit faire son prochain mouvement
    {
        Vector3 targetVelocity = new Vector2(_horizontalMouvement, rb.velocity.y);
        rb.velocity = Vector3.SmoothDamp(rb.velocity, targetVelocity, ref velocity, .05f); //smoothdamp pour faire un mouvement plus fluide
    }
}
```

Nous avons créé deux méthodes: une méthode *FixedUpdate* pour calculer la vitesse du personnage sur l'axe horizontal et pour ensuite l'envoyer au rigidbody2D à l'aide de la deuxième méthode: *MovePlayer* qui va permettre de savoir quand le personnage va faire son prochain mouvement.

Notre personnage se déplace à présent de gauche à droite ou de droite à gauche sur l'axe horizontal.

Nous avons continué avec le système de saut qui est venu s'implémenter à nos méthodes créées précédemment.

Nous avons premièrement utilisé les booléens *isGrounded* et *isJumping* pour savoir si le perso veut sauter et si il est bien au sol et si toutes ces conditions étaient remplies pour que nous lui donnons une force qu'on choisit nous même dans l'inspector pour définir sa puissance de saut.

Pour se faire nous avons créé un *GroundCheckLeft* et un *GroundCheckRight*, ces deux checks vont permettre de savoir si le personnage est au sol et ainsi éviter les doubles sauts. Nous avons créé dans le code, une boîte de collisions renvoyant un booléen entre ces deux checks. (voir image ci-dessous)



```
using UnityEngine;

[Script Unity | 0 références]
public class DeplacementJoueur : MonoBehaviour
{
    public float moveSpeed; //vitesse de déplacement
    public float jumpForce; //force de saut

    public bool isJumping;
    public bool isGrounded; //pour vérifier si le perso est au sol

    public Transform groundCheckLeft; //pour faire reference aux gdl et gdr créé dans l'inspector
    public Transform groundCheckRight;

    public Rigidbody2D rb; //élément faisant référence au rigidbody de notre personnage
    private Vector3 velocity = Vector3.zero; //vitesse sur axe x,y,z car meme en 2D vitesse sur ces trois axes dans unity

    [Message Unity | 0 références]
    void FixedUpdate() //fixed update car on utilise de la physique
    {
        isGrounded = Physics2D.OverlapArea(groundCheckLeft.position, groundCheckRight.position); //va creer une boite de collision entre gcp, gcr et si collision renvoie //true sinon false

        float horizontalMouvement = Input.GetAxis("Horizontal") * moveSpeed * Time.deltaTime; //calculer la vitesse du personnage sur l'axe horizontal, //le delta time permet de récupérer la vitesse au fil du temps

        if (Input.GetButtonDown("Jump") && isGrounded) // pour savoir si le joueur veut sauter si il appuie sur la barre espace et si il est au sol
        {
            isJumping = true; // le perso veut sauter
        }

        MovePlayer(horizontalMouvement); //pour envoyer le mouvement au rigidbody pour deplacer le personnage
    }

    void MovePlayer(float _horizontalMouvement) //pour calculer quand le personnage doit faire son prochain mouvement
    {
        Vector3 targetVelocity = new Vector2(_horizontalMouvement, rb.velocity.y); //l'endroit vers lequel on va se déplacer sera basé sur _horizontalMouvement (axe x) et //sur rb.velocity.y (axe y) 2 axes car utilisation Vecteur2

        rb.velocity = Vector3.SmoothDamp(rb.velocity, targetVelocity, ref velocity, .05f); //smoothdamp entre vitesse actuel du rb et cible vers laquelle on veut //déplacer le perso pour faire un déplacement + lisse et vector 3 car vitesse sur les trois axes, 0.05 est l'échelle de temps sur laquelle le persona va //effectuer son déplacement.

        if(isJumping == true) //si il a appuyer sur la barre espace on ajoute une force a notre joueur sur l'axe vertical
        {
            rb.AddForce(new Vector2(0f, jumpForce));
            isJumping = false; //pour préciser qu'on est plus en train de sauter
        }
    }
}
```

Nous avons donc finalement réussi à faire marcher notre personnage et à le faire sauter. Nous avons donc atteint les objectifs prévus pour cette première soutenance même si l'animation de saut reste à améliorer.

Pour la prochaine soutenance nous ajouterons les mécaniques de combat contre des futurs monstres. De plus, nous avons remarqué qu'il fallait appuyer plusieurs fois sur la barre espace pour que le personnage saute donc nous devons donc corriger cela pour la prochaine soutenance.

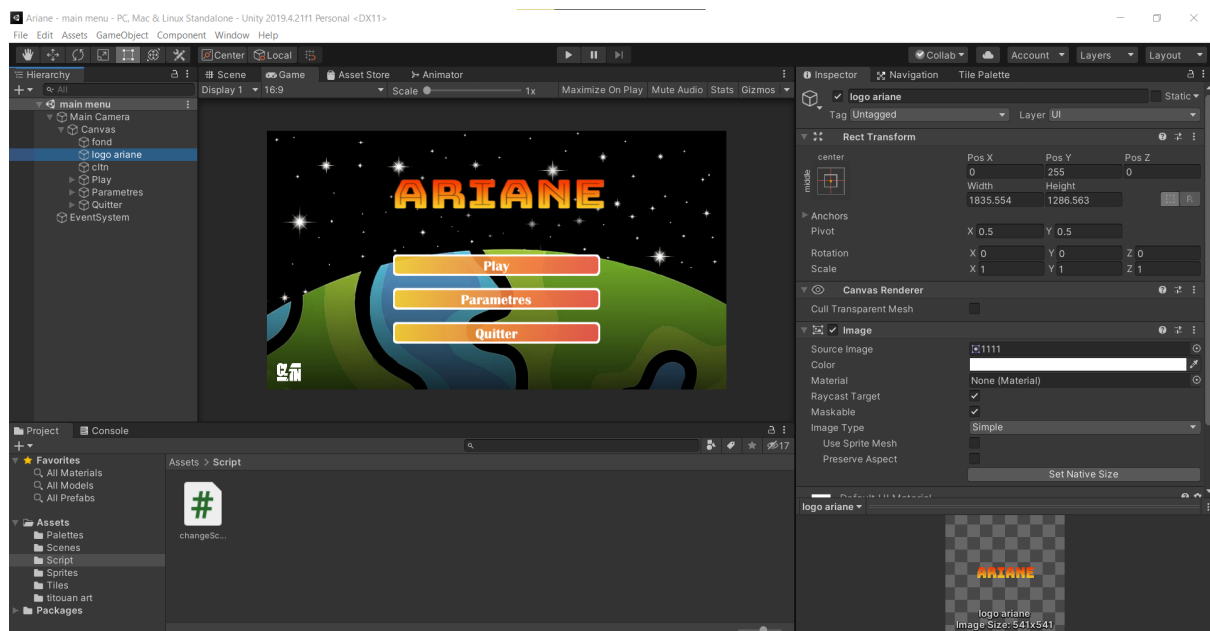
Interface :

Concernant l'interface, on a voulu recréer un thème spatial dans tout le jeu pour être accordé avec le thème du jeu qui parle d'Ariane, l'héroïne de notre histoire qui porte le nom des fusées de l'agence spatiale européenne. Pour cela, nous avons créé deux croquis sur Photofiltre pour savoir vers où se diriger et nous faciliter la création des menus dans Unity.

La première partie est le menu principal : il s'ouvre en premier quand on lance le jeu. Il s'agit donc d'une partie importante du projet, elle est composée d'un fond qui représente une planète quelconque avec l'espace derrière elle. Ensuite en haut, on retrouve le logo "Ariane". Ensuite nous avons trois boutons, le premier est le bouton "Play", il permet d'accéder au menu du choix de couleurs du personnage pour pouvoir ensuite démarrer le jeu, le deuxième est le bouton "Paramètres", il permet de nous emmener au menu paramètres où se trouveront les différents réglages du jeu, pour finir, le dernier bouton "Quitter" est le bouton de sortie de jeu. Enfin nous avons rajouté le logo de notre groupe en bas à droite.



Maquette du menu principal

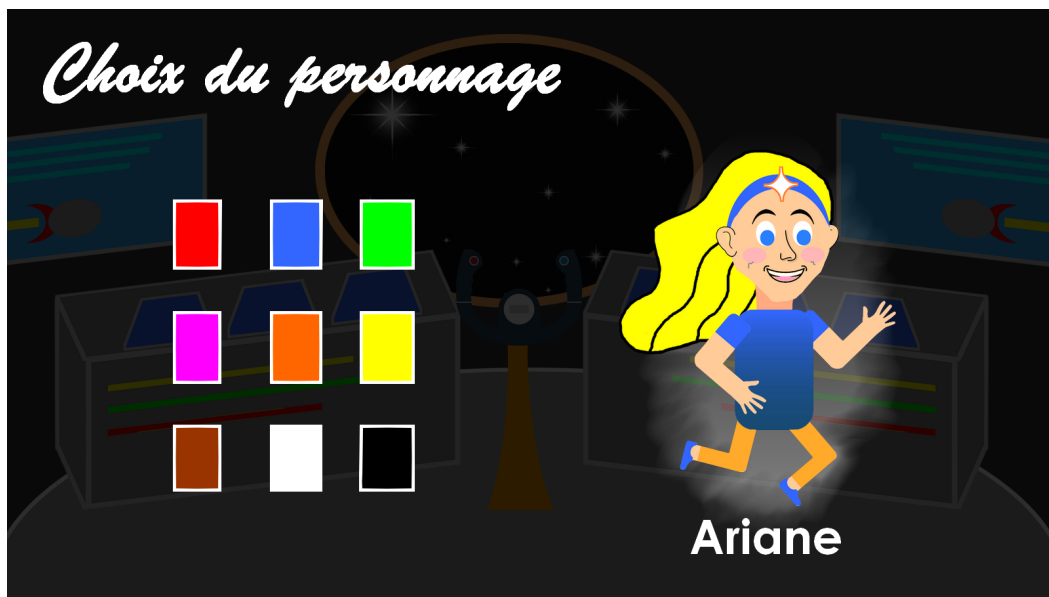


Version Unity du menu principal

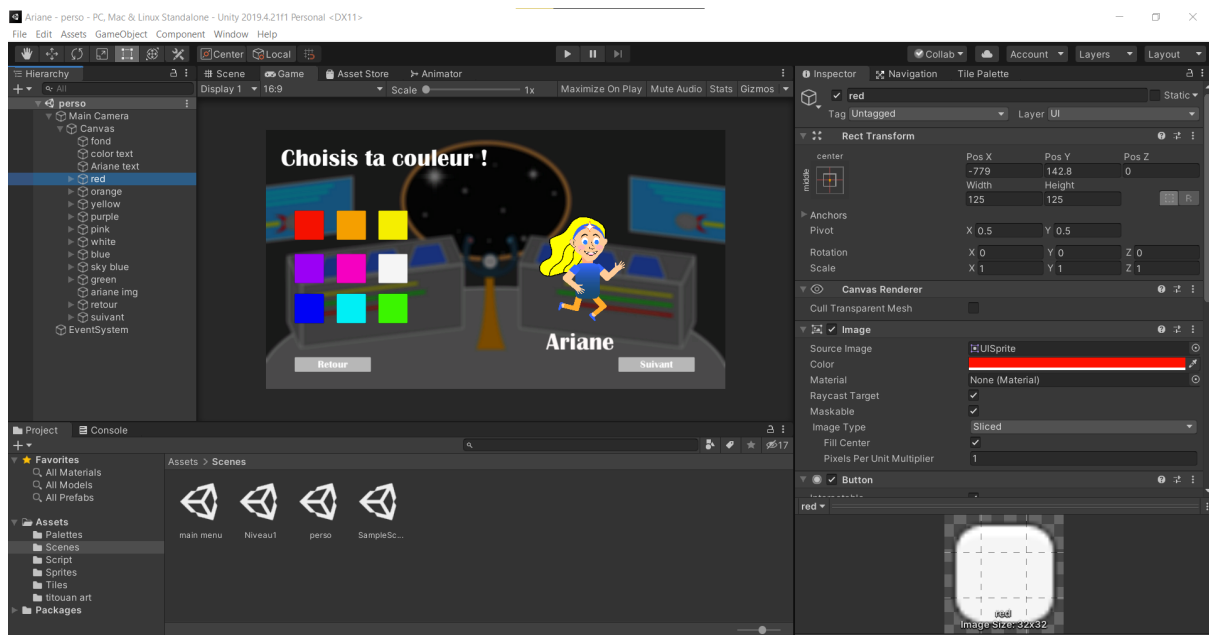
La deuxième partie est le menu de choix de couleurs du personnage, il est composé d'un fond représentant l'intérieur d'une fusée recréé sur Photofiltre, il est flouté et assombri par principe de visibilité du menu.

Ensuite en haut à gauche se trouve le nom du menu "Choisis ta couleur !", puis en dessous se trouve les neuf choix de couleur qui sont le rouge, le orange, le jaune, le violet, le rose, le blanc, le bleu, le bleu ciel et le vert, elles sont représentées sous forme de boutons qui par la suite pourront définir au personnage sa couleur. À la droite du menu se trouve la représentation d'Ariane avec sa couleur attribuée.

Enfin nous avons ajouté un bouton "Retour" pour revenir au menu principal et un bouton "Suivant" pour lancer le jeu. Ce menu n'est pas définitif, nous ajouterons des commandes liées aux parties en réseaux lorsque nous commencerons à développer cette partie.

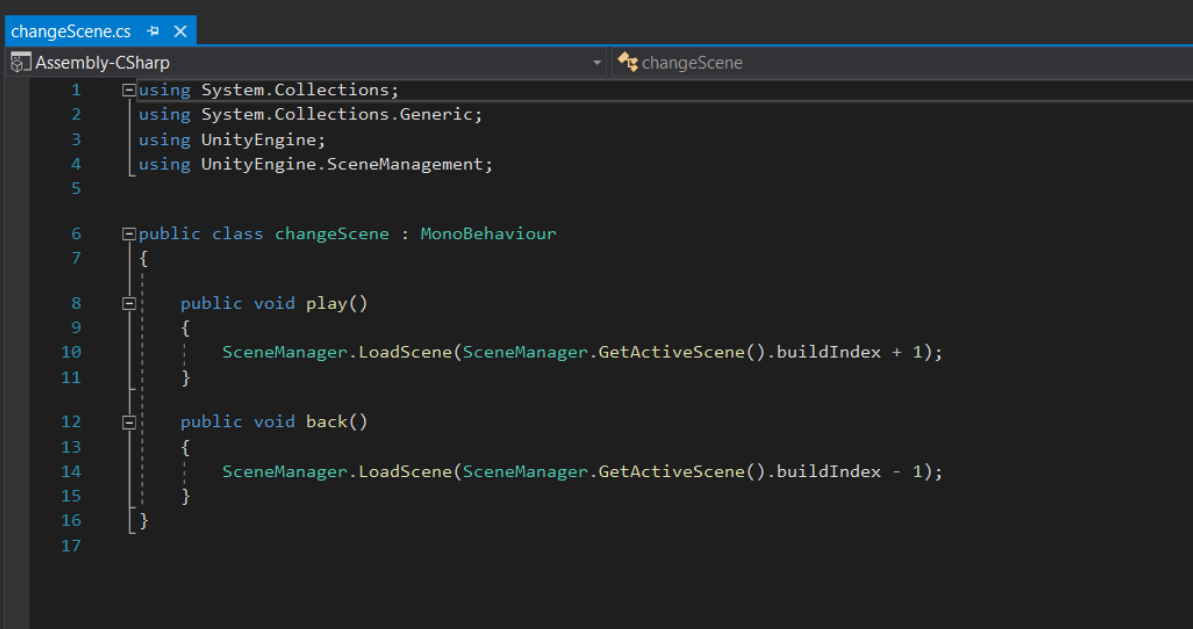


Maquette du menu “Choix de couleur”



Menu actuel du choix de couleur sur Unity

Pour l'instant nous avons inclus très peu de code dans l'interface du fait que le jeu n'est pas encore fonctionnel et qu'il nous manque le sprite final d'Ariane. Donc pour le moment les seuls codes présents sont pour changer entre les différents menus grâce aux boutons "Play", "Suivant" et "retour".



```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class changeScene : MonoBehaviour
7  {
8      public void play()
9      {
10         SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
11     }
12     public void back()
13     {
14         SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex - 1);
15     }
16 }
17
```

Animateur:

Concernant cette partie Animateur, nous avons décidé de la séparer en deux sous- parties: la première partie comprenant le scénario et la deuxième les animations du personnage et des monstres.

Nous avons prévu un avancement de 20% pour cette partie.

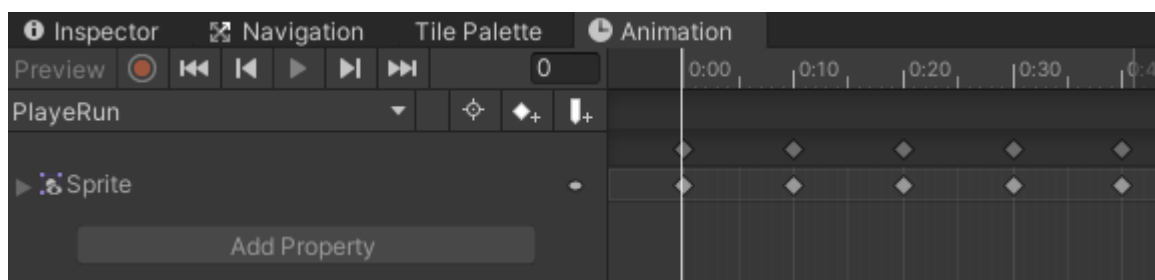
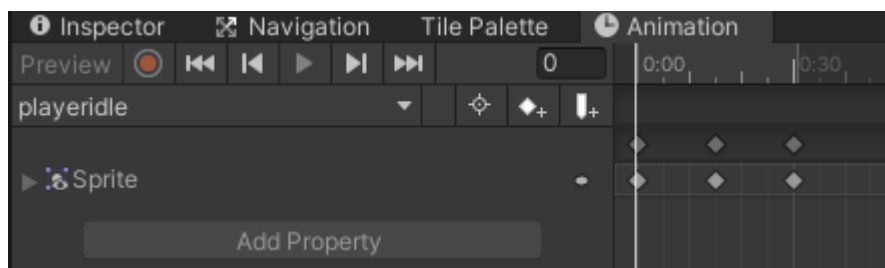
Nous n'avons pas commencé les différents scénarios, nous avons pensé qu'il était mieux de commencer par les animations des différentes créatures car beaucoup plus difficiles à mettre en place qu'un scénario qui lui demande juste une réflexion de groupe.

Comme dit précédemment nous avons utilisé Unity lié à Visual Studio pour mettre en place les différentes animations.

Nous avons fait les animations avec le précédent sprite utilisé (cf Gestion du personnage), comme dit dans la partie précédente nous remplacerons ce sprite par le nôtre pour la prochaine soutenance.

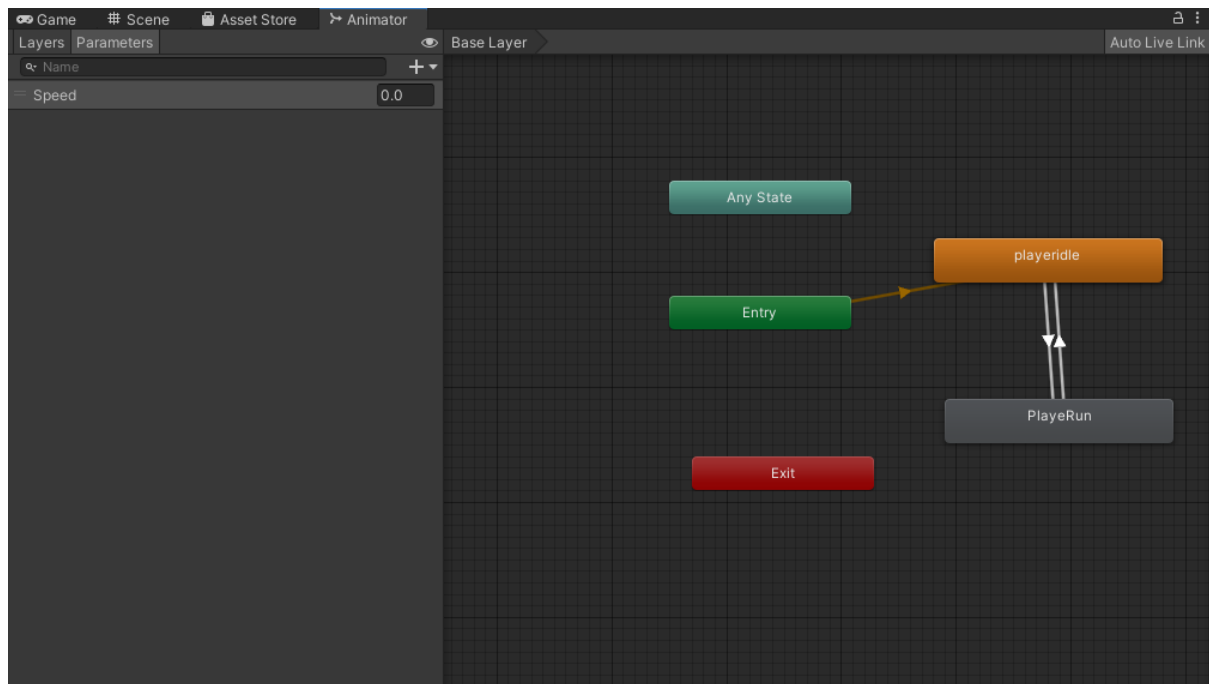
Nous nous sommes penchés sur l'animation de respiration (quand le personnage ne fait rien, quand il est à l'arrêt) et sur l'animation de déplacement du personnage.

Nous avons superposé les frames du sprites dédié au déplacement et à la respiration avec quelques secondes d'intervalles pour atteindre l'animation voulu.



Sur Unity nous avons utilisé l'outil l'Animator proposé qui permet de bien mettre en place les différentes animations et de les relier entre elles en y implantant des conditions ou non.

Nous avons ajouté un paramètre de vitesse pour savoir quand déclencher l'animation de marche.



Enfin nous avons implanté l'animation dans le script qui fait bouger le joueur en nous servant du paramètre speed créer auparavant.

```
public Animator animator; //animation  
  
float characterVelocity = Mathf.Abs(rb.velocity.x); //pour toujours renvoyer une vitesse positive  
animator.SetFloat("Speed", characterVelocity); //recupérer la vitesse horizontale
```

En somme, pour les animations nous pensons que les objectifs fixés ont été respectés pour cette première soutenance.

Nous prévoyons pour la prochaine soutenance de se pencher sur le scénario et d'ajouter de nouvelles animations sur le personnage principal comme les animations de combat, les animations des monstres et les animations de pertes de vie..

Site Web du projet:

Afin de créer le site web de notre projet, nous avons utilisé l'éditeur de texte Sublime Text et nous avons dû avoir recours à différents tutoriels pour apprendre à coder en HTML et en CSS. Ces tutoriels nous auront été très utiles et bénéfiques pour créer le site.



Logo de Sublime Text

Nous avons pu faire grâce au HTML la structure de la page d'accueil du site et des autres pages: nous avons pu écrire du texte et les titres; ajouter des images et faire la mise en page.

```
<html>
  <head>
    <meta charset="utf-8">
    <header>
      <title>Ariane</title></header>
      <link rel="stylesheet" href="Custom.css"> <!--lien vers le fichier .css pour la décoration du site-->
    </head>
    <body>
      <font color="white"><!--modifier la couleur de la police d'écriture-->
      <div id="block"><!--permettre de créer un bloc de contenu-->
        <h1>ARIANE</h1><!--nom du titre important-->
        <ul>
          <li><a href="Presentation du projet.html">Présentation du projet</a></li><!--lien vers les différentes pages-->
          <li><a href="Ressources.html">Ressources</a></li>
          <li><a href="Archives.html">Archives</a></li>
          <li><a href="Téléchargement du jeu.html">Téléchargement du jeu</a></li>
        </ul>
        <!--insertion d'une image accompagnée des dimensions-->
        <p> Nous vous souhaitons la bienvenue sur le site web de notre premier jeu intitulé Ariane.
        Ce site possède tout d'abord les différentes informations concernant notre projet (la presentation des membres, la chronologie du projet, un tutoriel et
        un synopsis du jeu et les éventuels problèmes rencontrés) .Ensuite nous aurons les ressources utilisées tels que les sons, les logiciels et bibliothèques.
        De plus des archives contenant les rapports des différentes soutenances seront aussi disponibles sur le site. Enfin le lien pour le téléchargement du jeu
        sera également présent.
        Nous vous souhaitons une nouvelle fois la bienvenue sur notre site et excellente navigation !</p><!--zone de texte-->
```

Capture du code en HTML

De plus, nous avons créé une page en CSS afin de customiser notre site et le rendre plus agréable à regarder: nous avons pu modifier la couleur de la police d'écriture, la typographie et la taille d'écriture; nous avons ajouté un fond accompagné d'une petite animation.

```
h1
{
    text-align:center;/*alignement du titre*/
    text-decoration: underline;/*titre souligné*/
}

body
{
    background-image:url("Images/photofunky.gif");/*fond'écran*/
    background-size:cover;/*couverture du fond*/
    font-size:1.2em;/*taille de la police d'écriture*/
    font-family:Verdana;/*typographie de la police d'écriture*/
}

#block
{
    width: 900px;/*taille du bloc de contenu*/
    margin:0 auto;/*positionnement du bloc de contenu au milieu de la page*/
    background-image:url("Images/photofunky.gif");/*fond d'écran du bloc de contenu*/
    background-size:cover;
    text-align:center;/*alignement du texte*/
    align-content:center;/*centrer le texte et les images*/
}

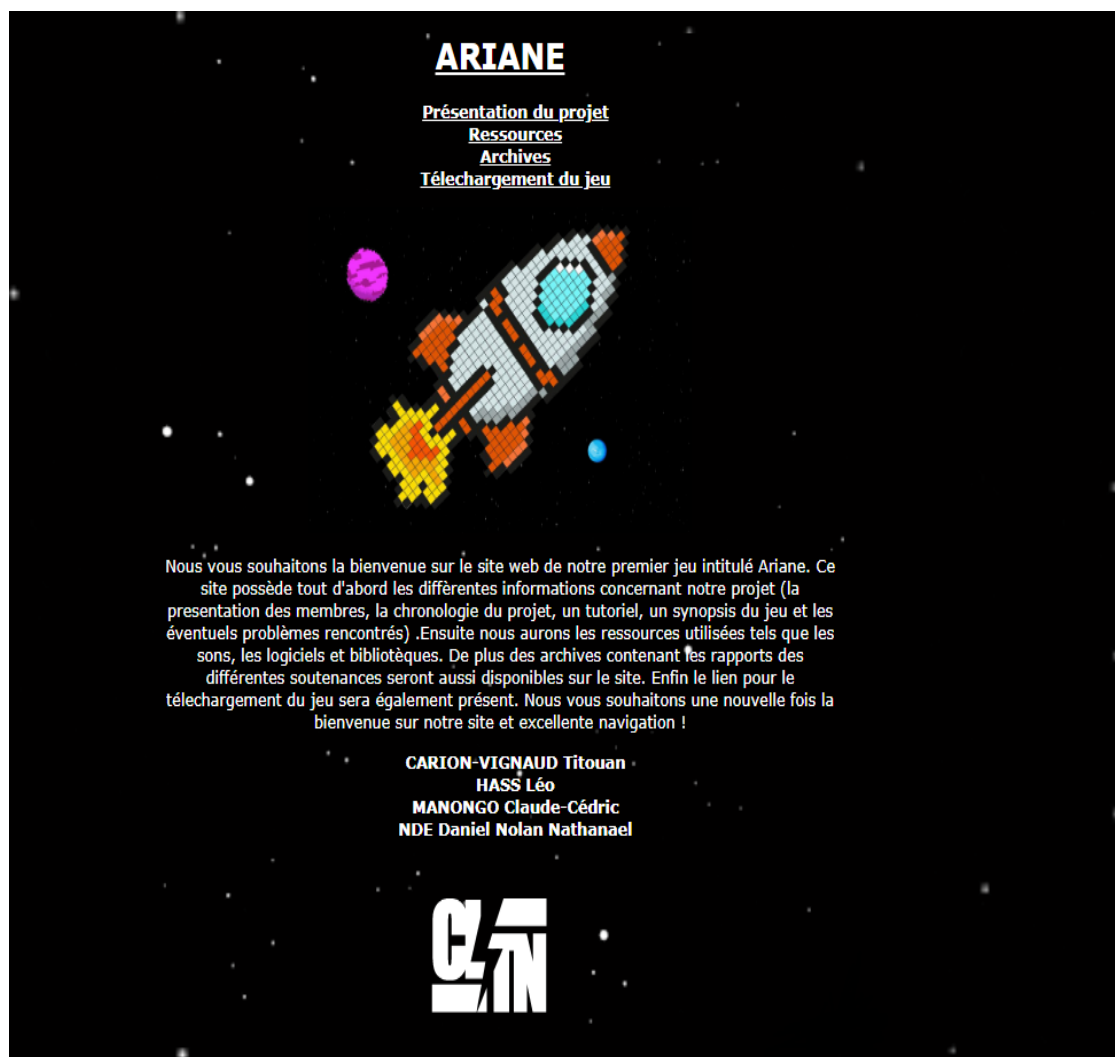
ul li
{
    list-style-type:none;/*permet d'enlever les points faits lors d'une liste*/
}

a:visited
{
    color:white;/*couleur des liens*/
}

a:hover
{
    text-decoration:underline;/*liens soulignés*/
    color:grey;/*couleur du lien lorsque la souris passe dessus*/
}
```

Capture du code en CSS

Grâce à cela, nous avons pu progressivement commencer à créer notre site. Nous avons pu bien avancer et faire ce que nous avons prévu dans le cahier des charges. La structure du site est déjà faite: les différentes pages, qui vont nous servir à ajouter du contenu à notre site, sont faites ainsi que le fond et les titres. De plus, nous pouvons déjà commencer à naviguer sur le site dans les différentes pages. Néanmoins comme nous ne sommes qu'au début, le contenu n'est pas encore présent mais il le sera pour les prochaines soutenances.



Voici la page d'accueil de notre site web

Notre site web sera composé de 4 parties accessibles à partir de la page d'accueil:

1) Présentation du projet:

Nous aurons premièrement les informations sur les membres du groupe, la chronologie de réalisation, les éventuels problèmes rencontrés ainsi que les solutions envisagées pour répondre à ceux-ci.



Capture de la page sur la présentation du projet

2) Ressources:

Ensuite, nous montrerons les sons utilisés, les logiciels, images et toutes les autres ressources qui nous auront aidés à mener à bien ce projet.

3) Archives:

De plus, nous mettons à disposition les différents rapports de soutenances et le rapport final du projet sur cette page.

4) Téléchargement du jeu:

Pour finir, nous aurons sur cette page le lien pour pouvoir télécharger le jeu.

Nous tenons à ajouter que nous avons hébergé notre site sur Github. L'avantage de l'hébergement sur Github est la gratuité de l'opération puisque nous ne payerons rien pour l'hébergement du site. De plus, il nous permet de faire des mises à jour régulièrement suivant l'avancement du projet et d'avoir un lien pour accéder facilement au site web sans devoir télécharger les fichiers HTML et CSS composant le site à chaque fois.

Ingénieurs son et réseau:

Concernant le son et le réseau nous avons décidé de ne pas s'y pencher pour cette première soutenance, nous avons préféré nous concentrer plus sur les mécaniques du jeu.

Au niveau du son, nous prévoyons pour la prochaine soutenance d'ajouter du son aux différentes mécaniques du personnage.

Nous pourrions par exemple ajouter des sons couplés aux animations de combat, ou encore aux animations de saut.

Nous pouvons aussi ajouter une musique d'ambiance pour le jeu en général ou encore du son qui se déclenche lors de l'utilisation de boutons ou lors de la réussite d'un niveau.

Au niveau du réseau nous comptons implémenter un mode multijoueur à 4 joueurs maximum.

Les différentes personnes pourront jouer ensemble à distance, ils pourront au début sélectionner une version d'Ariane qui leur convient et ensuite les 4 joueurs devront passer les différents niveaux tout en faisant attention à leur barre de vie.

La difficulté des niveaux sera donc augmentée comme par exemple avec la réduction du nombre de vies ou encore l'ajout de monstres.

3-CONCLUSION:

Au cours de cette première étape dans notre projet nous avons pris conscience des difficultés à créer un jeu vidéo. Pour y pallier nous avons mobilisé les compétences du groupe et chacun des membres du groupe a dû s'adapter aux différents aspects techniques du projet comme par exemple se servir de git pour travailler ensemble ou encore héberger le site.

Il a été aussi difficile de réunir tous les membres du groupe compte tenu des emplois du temps de chacun et de la crise sanitaire liée à la *COVID19* qui n'a pas arrangé les choses.

Toutefois les objectifs prévus pour cette soutenance ont été globalement atteints pour cette première soutenance. Nous sommes pour l'instant satisfait du premier aspect de notre jeu *Ariane* et les obstacles rencontrés ont permis d'améliorer la cohésion de groupe.

Nous envisageons pour les futures soutenances d'améliorer notre gestion du temps et nous essayerons d'atteindre les objectifs fixés pour les soutenances.

Nous serions très heureux de voir le résultat final de notre première création informatique en groupe et nous espérons que celle-ci marchera comme nous l'aurions voulu.



