



Un projet réalisé par l'équipe : Time's Masters

Sa cheffe : Aïssa Sohane

*Ainsi que Danycan Dané, Royer Julia et DiSanto
Alexandre*





Sommaire



I-Introduction P.3

II-Légères modifications de responsabilités P.4

III-Description des tâches effectuées par personne

A/ Sohane et Danaé P.5

B/Alexandre P.12

C/Julia P.13

IV- Tâches à effectuer pour la prochaine fois P.15

V-Conclusion

A/Avis personnels P.17

B/Annexe P.19

Introduction :

Lors de la création du cahier des charge nous n'avions aucune idée de ce à quoi ressemblerait la charge de travail pour la création d'un jeu comme Time Quest 42.

En effet, un tel projet nécessite une équipe soudée, avec une bonne communication, et beaucoup de motivation.

Le travail d'équipe permet d'avancer plus facilement, car nos parties sont complémentaires, sans ça, nous ne pourrions pas livrer un projet à temps.

Une fois cette partie mise en place, les plannings ont été légèrement adaptés.

Notre organisation n'est probablement pas figée, car la route est semée d'embûches, nous y ferons face, en s'adaptant au maximum aux contraintes.

Ce premier rapport de soutenance retrace l'intégralité de notre travail ces dernières semaines. Il y contient, de légères modifications du cahier des charges.

Vous y trouverez également les difficultés que nous avons rencontrées, notre avancée dans le code, le début de la modélisation de notre plateau, ainsi que nos impressions personnelles.

Légères modification des responsabilités

Au moment de la création du cahier des charges, nous n'avions aucune idée du temps que prendrait chacune des tâches, et surtout du temps que chacun était prêt à investir dans ce projet.

En conséquence, voici les modifications que nous avons faites sur la répartition des tâches :

	Alexandre	Danaé	Julia	Sohane
Site web	○	x		
Graphismes	x	x	○	x
Menu			x	○
Sauvegarde	○			
Soundtrack	x	x	○	x
Fonctionnalités du jeu		○		○
Back end	x	○	x	○
Documents		○		○

(Rapports de soutenances, diapo, cahier des charges,...)

○ → Responsable

x → Suppléant

Danaé et Sohane sont désormais également responsable des fonctionnalités du jeu et du Back end.

Le tableau ci-dessous présente ce qui a été fait ces cinq dernières semaines.

On peut dire que nos tâches personnelles ont été accomplies, nous avons réussi à faire preuve de cohésion et d'entraide entre nous.

Nous avons en effet trouvé un rythme de travail qui nous convient.

Semaine	1	2	3	4	5
Julia	Comment utiliser Unity	Unity	Graphisme de la map	Graphisme de la map	Générer la carte
Alexandre	Comment créer un site web	Comment créer un site web	Créer le backend du site web	Continuer le backend du site web	Finir le backend du site web
Sohane	Carte du jeu	Carte du jeu	Constructeurs	Architecture des Fonction	Rapport de soutenance
Danaé	Carte du jeu	Carte du jeu	Constructeurs	Architecture des fonctions	Rapport de soutenance

Description des tâches accomplies

A) Sohane et Danaé

1-Conception du jeu

La première étape, une des plus importante a été la revue et la précision des règles, et du matériel requis pour un jeu de stratégie basé sur l'interaction avec son plateau.

Pour avoir une idée des mensurations de notre plateau, nous avons créé un tableau sur Excel de taille 42*42 cases (la dimension du plateau initial, celui dans le cahier des charges), mais nous nous sommes rendu compte, qu'avec 42 tours et un plateau d'une telle dimension, le jeu deviendrait compliqué, voire impossible, surtout si l'on veut ponctuer le jeu, avec des actions, des intelligences artificielles, et surtout si l'on veut faire en sorte que les deux équipes se croisent.

Après plusieurs tests, le plateau final aura une dimension 24*24 cases.

En ce qui concerne les différentes populations, nous n'avons rien changé, car nous avions déjà au préalable réfléchi à tout ça sur le cahier des charges.

Sur ce dernier n'était pas précisé les différentes cases et actions possibles sur le plateau.

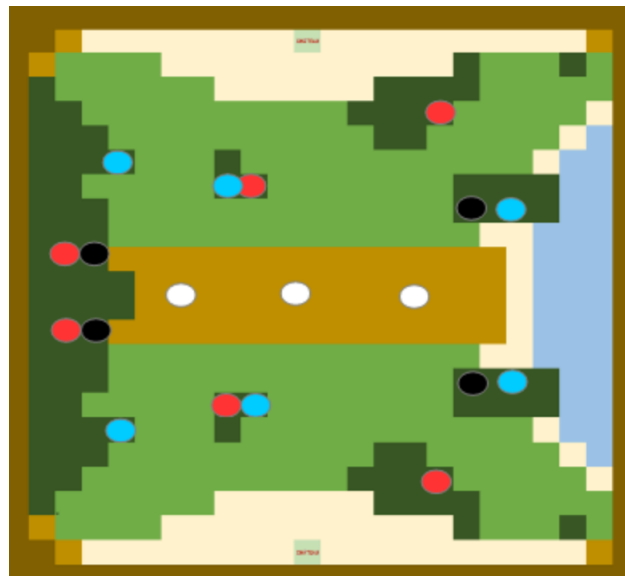
Cette partie nous a pris énormément de temps car il a été très dur de trouver un juste milieu, afin d'avoir un plateau assez rempli pour divertir, amuser, sans paraître lourd, en ne favorisant aucune équipe.

Voici donc les actions, événements définitifs de notre jeu :

- Dans la forêt, (rond rouge), des chevaux pourront être capturés, en échange d'énergie (de gemmes), dans le but de créer des cavaliers ou des rois, et donc d'avancer plus vite, d'avoir plus de vies, et pour le roi d'échanger un personnage de la troupe ennemie en allié.
- Des cannettes de coca (un peu de modernité) ainsi que des carottes apparaîtront de manière aléatoire sur la carte (à chaque tour), et pourront être consommées par les personnages qui passent par-là, (une bouteille de coca et une carotte seront générées par tour).
- Toujours dans la forêt, les joueurs pourront récolter du bois (rond bleu), en échange d'énergie, ce dernier leur servira pour pêcher, ou pour construire un bateau (leur permettant de traverser plus vite et d'éviter les montagnes (en marron clair), qui ralentissent les mouvements).

- ATTENTION !
Des troubadours et des ours rôdent près des forêts et de la montagne, ce sont nos intelligences artificielles, ils se rapprocheront des personnages qui s'approchent un peu trop à leur goût d'eux.
S'ils sont sur la même case, ils les attaquent, réduisant la jauge de vie du personnage attaqué. (Les ours sont représenté par les ronds noir),(les troubadours par des rond blanc).
- Si deux joueurs se croisent celui qui attaque récupère un quart de l'or de l'attaqué.

Cette carte n'est que le brouillon qui nous a servi à placer les case, brouillon qui sera modélisé sur Unity.



Au début de la partie les deux joueurs ont 5 pièces d'or, et 10 vies.

Rappelons :

Les différents Types de personnages :

Le soldat :

Le personnage qui se situe au bas de la classe d'armes, il ne peut se déplacer que d'une case autour de lui.

- Vie : 10
- Attaque : 5
- Prix : 3

Le cavalier :

Le personnage se situe au milieu de la classe d'armes, il peut se déplacer de 3 cases autour de lui, car il sait monter à cheval.

- Vie : 20
- Attaque : 10
- Prix : 9

Le roi :

Le roi se situe tout en haut de l'échelle sociale.

Il peut être à pied, alors, il se déplace comme un soldat.

- Vie : 20
- Attaque : 10

S'il est à cheval, il se déplace comme un cavalier.

- Vie : 40
- Attaque : 20

Si une troupe ennemie est à portée de lui, il peut, s'il en a les ressources, transformer une troupe ennemie en alliée.

Un joueur ne peut avoir qu'un seul roi à la fois !!

- Magie : 3 (il peut changer 3 personnages adverses)
- Prix : 40

Les I.A : (ils se déplaceront tout seul dès qu'un personnage sera dans leur périmètre)

- Le barbare :
Comme un soldat. Il se déplace si la troupe d'un des joueurs est à deux cases de lui.
- L'ours :
Comme un cavalier, si une troupe est à 5 cases de lui, il va vers elle.

Une fois ces modifications faites, s'est posé la question du code, comment coder ce jeu ?

2-Le début du code

Il nous est paru évident, grâce au TP Quidditch, de faire des classes, des constructeurs, en bref, de programmer en orienté objet.

Nous avons pris exemple sur ce fameux TP, pour la construction de nos classes (une des phases la plus importante, car il faut être précis).

Nous avons donc fait 5 classes (plus la classe print pour afficher notre jeu).
(Notre code se trouve dans les annexes)

La classe Characters:

Pour construire cette classe les TP en Programmation orienté objet nous ont été d'une aide précieuse. (Tel que le TP Space Invaders, dont vous trouverez la classe joueur en annexe) En effet, ils nous ont permis de maîtriser les termes d'héritage d'une classe, de constructeurs ou encore de classe mère.

Grâce à des règles de jeu précise, nous avons facilement trouvé les attributs de la classe, ici nous avons mis pour chaque joueurs les caractéristiques suivantes :

- Une position (ses coordonnées)
- Une équipe
- Une jauge de vie
- Un prix (pour pouvoir l'acheter)
- Un prix s'il est attaqué sur terre
- Sa perte de bois s'il est attaqué dans le lac
- Sa force d'attaque
- Le nombre de déplacement par tour
- La liste des deux équipes (Team1 et Team2)

Nous avons également commencé à écrire certaines fonctions, les plus générales :

- ValidPositionPersonnages (qui vérifie qu'un personnage est bien dans le terrain)
- AskMove qui va déplacer les personnages quand le joueur va cliquer sur les touches directionnelles.
- Une fonction CanAttack, qui va regarder si un joueur peut attaquer son adversaire.

Dans cette classe, nous avons fait des sous-classes héritières de Characters (avec les différents personnages), en mettant leurs attributs à jour avec les valeurs écrire dans la règle du jeu.

Dans ces 3 classes filles, (soldat, cavalier et roi), nous avons rédigé la fonction update, qui en tenant compte de validposition, va mettre à jour la position de chaque personnage.

Il y en a une dans chaque classe, car chaque personnage ne se déplace pas du même nombre de case.

La classe ia :

Elle se présente presque comme Characters, les attributs sont les suivants :

- Une position (coordonnées)
- Une variable attaque, qui va marquer la perte de gemme du joueur à l'issue de l'attaque de l'ia.
- Un temps d'attente après une attaque, afin de laisser s'enfuir le joueur.
- Et une jauge vie qui va descendre à chaque attaque.

La classe Map, avec les attributs suivants :

- La position du carreau (car nous avons décidé de coder notre jeu carreau par carreau).
- Un booléen pour savoir si le personnage est dans cette zone.

Sachant que nous avons créé les zones suivantes :

- Forêt
- Montagne
- Lac
- Désert
- Prairie
- Château
- Interdit (représente le contour de la carte où il est interdit d'aller).

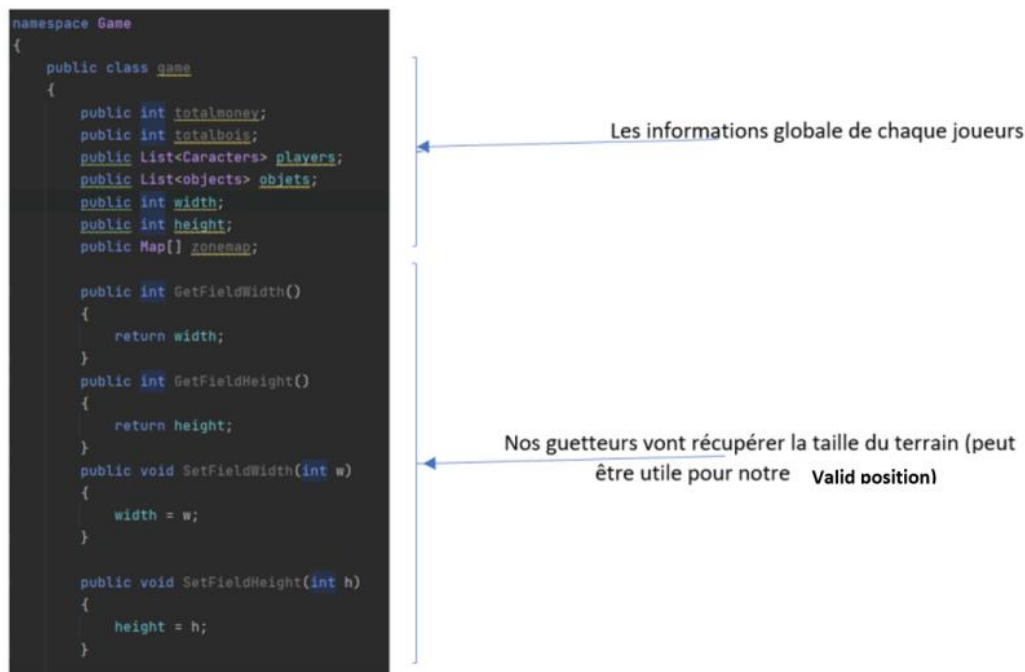
La classe objects présente tout ce qui va être présent sur la carte, et que les joueurs pourront utiliser, les attributs de la classe sont les suivants :

- Une position
- Un coût en or
- Un coût en vie
- Un gain en vie
- Un gain en or
- Un booléen pour savoir si le joueur souhaite réaliser l'action, s'il en a la possibilité.

Les classes héritières sont donc tout ce qui a été cité dans les règles de jeu.

Enfin, la classe games, va « rassembler » toutes les classes, les assembler et créer un lien entre elle.

Encore ici, la classe Game du TP Quidditch a été d'une aide indispensable (nous l'avons mis en annexe), voici notre classe :



Cette fonction va permettre d'ajouter chaque joueur à son équipe

Ajouter les cases d'actions, les objets à la carte

Constructeur de notre fonction

```
public void AddTeam(Caracters cara)
{
    foreach(var carac1:Caracters in cara.Team1)
        players.Add(carac1);
    foreach(var carac2:Caracters in cara.Team2)
        players.Add(carac2);
}

public void add_object(object obj)
{
    objets.Add(item:new Cheval(x:3, y:10));
    objets.Add(item:new Cheval(x:10, y:8));
    objets.Add(item:new Cheval(x:19, y:9));
    /*le reste en symétrie de ces coordonnées*/
}

public game(int w, int h)
{
    width = w;
    height = h;
    players = new List<Caracters>();
}
```

Vous pouvez retrouver l'intégralité de l'avancé de notre code en annexe.

Danaé ayant déjà suivi une formation sur OpenClassroom sur le HTML CSS, et étant suppléante du site web, c'est naturellement qu'elle a commencé à préparer l'architecture du site,

Les contraintes ont été de bien coordonner le CSS et l'HTML, afin que la présentation des pages donne envie de naviguer sur notre site, et d'attirer les futurs joueurs.

Elle a donc utilisé sublime text, un éditeur, pour coder le site, voici son code :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="test.css" />
    <title>Time Quest 42</title>
  </head>
  <body>
    <header>
      <div id="bande">
        <div id="pho"><a href="Time quest 42.html">
        </a></div>
        <p></p>
        <div id="t1"><a href="Contenu/Origine.html"><h2>Origine</h2></a></div>
        <p></p>
        <div id="t3"><a href="Contenu/jeu.html"><h2>Le jeu</h2></a></div>
        <p></p>
        <div id="mc"><a href="Contenu/eda.html"><h2>Etat de l'art</h2></a></div>
        <p></p>
        <div id="mc"><a href="Contenu/tuto.html"><h2>Tutoriel</h2></a></div>
        <p></p>
        <div id="rech"><a href="Contenu/ins.html"><h2>Installation</h2></a></div>
      </div>
    </header>
    <div id="texte">
      <h1><strong>Time Quest 42</strong></h1>
      <h4><em>by the <strong>Time's masters</strong></em></h4>
      <a href="logop.png"></a>
      <p>Bienvenue sur la page web de Time Quest 42 un projet de 4 étudiants en informatique.
      Nous présenterons ici divers points clés importants ! </p>
    </div>
  </body>
</html>
```

```
@font-face {
  font-family: 'pol';
  src: url('Enchanted Land.otf');
}

body
{
  background-image: url("fond.png");
  text-align: center;
  color: white;
  font-size: 32px;
  font-family: 'pol';
}

header
{
  font-size: 25px;
  border: 2px white solid;
}

a
{
  color:white;
}

a:visited
{
  color:white;
}

#bande
{
  display: flex;
  justify-content: space-between;
  margin:auto;
  color: white;
}

p
{
  border-right: 1px white solid;
}

#t1,#t2,#t3,#t4,#mc,#rech
{
  margin: auto;
}
```

Elle a organisé sa page avec « head » les informations générales : d'où vient son style, et quel est le nom de l'entête.

Et body le contenu de la page avec l'entête où elle a utilisé des Div :

Les Div sont des identifiants permettant de former des blocs, où leurs composantes linéaires ou/et blocs sont de bases superposés, mais qu'on peut aligner grâce à la flex box. C'est ce qu'elle fait pour l'entête et pour la partie « Le jeu ».

Elle a pu aussi choisir une police libre de droit allant sur tous les navigateurs sauf internet explorer.

B) Alexandre

Tout d'abord pour le site web, il est parti sur du HTML comme il en avait déjà fait un petit peu l'an dernier. Ayant cette idée en tête il a cherché de quelles manières il pouvait le publier sur internet, il a trouvé trois façons d'héberger une page HTML : via un hébergeur gratuit, via un hébergeur payant et se l'héberger soit même.

Les performances de l'hébergeur gratuit lui paraissaient faible surtout s'il faut télécharger le jeu via le site web et il a trouvé que l'hébergeur gratuit ne laissait pas assez de contrôle sur le site.

Il a donc pensé à héberger lui-même via un Raspberry pi mais son wifi au centre de Toulouse n'étant pas de bonne qualité, il a préféré un hébergeur payant qui laisse du contrôle sur le site avec de bonnes performances tout en étant plus qu'abordable.

Pour la page html il s'est replongé dans les tutos pour voir comment faire un site présentable notamment avec Openclassrooms
« <https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-votre-site-web-avec-html5-et-css3/1604192-decouvrez-le-fonctionnement-des-sites-web> » .

Il y a appris comment utiliser le CSS pour largement mieux présenter/organiser le site.

Voici donc le résultat de notre site :



(Dans les annexes vous trouverez les différentes pages du site)

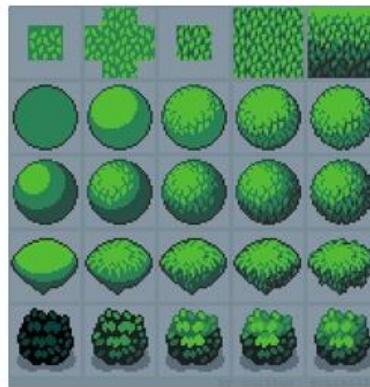
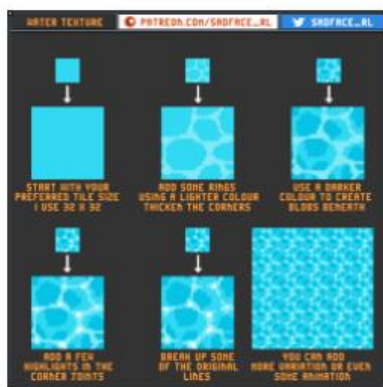
C)Julia

Afin de réaliser notre jeu, elle a dû se renseigner sur « Unity », un moteur de jeu. Pour cela, elle est allée regarder des vidéos sur la plateforme YouTube. Elle s'est notamment bien appuyée sur les tutos de la chaîne « TUTO UNITY FR ». Plus particulièrement sur sa série montrant comment créer un jeu vidéo en 2D sur « Unity ».

Par ailleurs, pour implémenter la carte de notre jeu, elle a réalisé une partie de l'aspect graphique. Plus particulièrement les zones qui auront une spécialité et rapporterons différents matériaux au joueur lorsque celui-ci interagis avec. Afin de rendre notre jeu plus agréable et que la partie graphique ne soit pas un frein pour nous, nous avons choisi de créer un jeu en 2D, style « Pixel Art ». Le pixel art est un style de dessin, dont l'intérêt est dans le rendu d'une image réalisée en pixel. Pour cela, elle a donc créé une « TileSheet ou TileSet » qui présente une partie ou toutes les textures d'un jeu vidéo. Elle s'est notamment inspirée de jeux reconnus dans ce domaine tel que « Dead Cells », ou bien « Pokémon » ou encore « Super Mario Bros ». Mais elle s'est également inspirée des recherches qu'elle a pu mener sur internet pour avoir des idées de style ou bien de dessin pour notre jeu.



Ces images l'ont aidé à réaliser la prairie verte de notre carte.



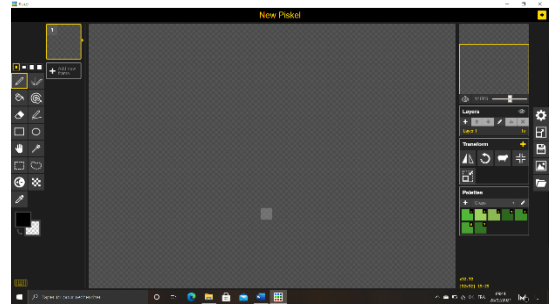
Pour créer ces fameuses textures, elle s'est d'abord appuyée sur une application nommée « Piskel ». Celle-ci propose gratuitement de créer ses propres dessins pixelisés.

Dans un deuxième temps elle a utilisé le logiciel libre de droit nommé « Gimp ». Il permet quant à lui de faire toutes sorte d'art pictural



numériquement. Celui-ci lui a été utile pour la construction même de la feuille de texture. Elle a tout simplement assemblé les précédemment dessins qu'elle avait réalisés à l'aide du logiciel « Piskel ». Et dans un dernier temps elle a créé la carte de notre jeu à l'aide du logiciel « Unity ».

« Piskel »



« Gimp »

Voici le plateau (même s'il n'est pas totalement terminé), calqué sur le brouillon que Sohane m'a envoyé :



Programme pour la prochaine Soutenance

Semaine	7	8	9
Julia	Graphismes persos	Générer les persos	Graphismes bâtiments
Alexandre	Système de sauvegarde	Système de sauvegarde	Système de sauvegarde
Sohane	Continuer le back end du jeu	Continuer le backend du jeu	Continuer le back end du jeu
Danaé	Comment implémenter souris	Implémenter souris	Continuer le back end du jeu

10	11	12	13
Continuer le back end du jeu	Créer les animations	Rattraper retard	Soutenance
Continuer le back end du jeu	Interaction avec le plateau	Rattraper retard	Soutenance
Continuer le back end du jeu	Rapport de soutenance	Rattraper retard	Soutenance
Continuer le back end du jeu	Rapport de soutenance	Rattraper retard	Soutenance

Les objectifs pour la prochaine soutenance, sont :

- Pour Julia qui a pris un léger retard sur la modélisation du plateau, de le terminer et d'avoir créé les personnages, bâtiments, afin de pouvoir tester notre jeu.
- Sohane et Danaé ont un peu d'avance sur leur programme, en effet, nous avons fini les constructeurs, déjà commencé les fonctions ainsi que la classe game en partie, et nous avons fini la conception du jeu.

Pour la prochaine soutenance, notre objectif est d'avoir terminé la partie code de notre jeu, du moins les fonctions indispensables, les plus importante, et commencer le menu du jeu :

Nous aimerions avoir fini la partie technique du jeu.

- L'objectif d'Alexandre est d'avoir trouvé comment configurer le système de sauvegarde.

Conclusion :

A) Impressions Personnelles

1-Sohane

Étant cheffe de groupe, ma première difficulté a été d'organiser nos agendas, de trouver un point d'entente, de motiver tout le monde, en bref d'attendre à connaître les habitudes de travail de chacun.

Il est vrai que nous avons eu du mal à avancer ensemble au début, et que nous n'avons pas les mêmes attentes de ce projet, que personnellement j'attendais beaucoup. Même si nous avons rencontré pas mal de branches qui nous ont barré la route, nous avons vite appris à prendre sur soi.

Le point positif que je peux en tirer personnellement, c'est que plus les semaines passent et plus il me tarde d'avancer dans ce projet, et de porter mon groupe, car dans le fond nous avons tous la même idée en tête :

Arriver à livrer Time Quest 42 dans les temps.

Je suis plutôt optimiste pour la suite de notre projet, et prête à affronter la difficulté, car le jeu en vaut vraiment la chandelle.

Personnellement, je n'avais qu'une hâte :
Commencer la partie code du jeu.

J'ai trouvé que l'ensemble des TP de programmation en orienté objet, m'ont apporté un bagage assez solide pour envisager cet exercice sereinement.

J'ai également découvert OpenClassroom, Unity et le langage html.

Au-delà du projet, ces connaissances nourrissent ma culture informatique, qui était moindre à mon entrée à l'EPITA, n'ayant jamais codé auparavant.

2-Danaé

Le projet je le trouve franchement assez excitant car je pense que c'est une bonne idée, ce qui est dur ce n'est pas forcément sa réalisation mais la gestion du travail de groupe et la communication quand tout le monde n'est pas aussi motivé.

Je pensais que ce serait très enrichissant et épanouissant mais quand on doit sans cesse essayer de motiver les autres en même temps que soit ainsi qu'attendre continuellement leur travail pour tenter de réaliser un travail de groupe ou chacun a besoins des travaux des autres.

Après j'ai appris à connaître des personnes que je n'aurais jamais connu autrement et ça c'est vraiment bien, d'autant plus que j'ai appris le css, html, php et javascript grâce à des idées de projets perso que m'a donné celui-ci.

Cela reste assez engageant pour la suite et montre toutes les perspectives que nous pouvons avoir, autant socialement que professionnellement

3-Alexandre

J'ai aimé chercher comment marchait le CSS et découvrir l'inconnu. Il n'y a rien qui m'a déçu, de difficile.

4-Julia

J'ai trouvé que la prise en main du logiciel « Unity » était plutôt simple et agréable. J'ai apprécié en apprendre plus sur celui-ci et j'ai très envie de l'utiliser davantage que cela soit pour notre jeu ou pour des projets personnels. Sur l'aspect graphique, je n'avais jamais expérimenté le dessin fait de pixel et je trouve cela plutôt simple, rapide et plaisant. Cela donne un rendu plutôt correct et sans trop de travail sur les dessins. J'ai très rapidement compris le fonctionnement de « Piskel ». Le logiciel « Gimp » quant à lui, est très complet et demande plus de recherches et d'apprentissages à son sujet. Cependant l'ayant déjà utilisé dans un but personnel, son fonctionnement ne m'était pas étranger. J'ai pu avoir quelques difficultés sur la création de textures, car il fallait rendre une image plutôt agréable avec une dimension de 32 sur 32. De plus, la fabrication de la carte sur « Unity » ne s'est pas fait sans accros. Sur les vidéos que j'avais regardé, pour créer cette base, ils ne mentionnaient pas explicitement une modification à faire sur la « TileSheet » avant de la placer sur l'interface de jeu. J'ai donc passé un certain temps à trouver mon erreur.

B) Annexe

En annexe n°1, l'intégralité de notre code. P.1 de l'annexe

En annexe n°2, la classe Game du TP Quidditch. P. 10 de l'annexe

En annexe n°3, la classe joueur de Space Invaders. P.12 de l'annexe

En annexe n°4, les autres pages du site web P.13 de l'annexe