

Projet Engi'Blocks

Rapport de première soutenance

Groupe Buildingee

Antoine Adam
Alexandre Dugot
Mohamed Ghaffour
Nathan Labernardiere

03 mars 2021



Table des matières

1 Introduction

2 Modifications du cahier des charges

3 Avancements individuels sur les fonctionnalités

- 3.1 Personnages
- 3.2 Gestions des sauvegardes
- 3.3 Gestion du vaisseau et design
- 3.4 Site Internet
- 3.5 Multijoueur
- 3.6 Effets sonores
- 3.7 Interface carte
- 3.8 Cinématiques
- 3.9 Missions et histoires
- 3.10 Gestion/ design planètes
- 3.11 Machines/inventions
- 3.12 Structure et design du vaisseau

4 Structure du répertoire

5 Expériences personnelles

6 Avances et Retards

7 Prévisions pour la suite

8 Conclusion

1 Introduction



Cette première période du projet avant la soutenance numéro une nous a permis de développer les fonctionnalités de base du jeu sans trop rentrer dans les détails, ces derniers étant nombreux et pas tous implémentables pour le moment.

Dans le cadre de ce projet nous avons tous les 4 appris à nous familiariser avec divers logiciels comme Unity pour créer le jeu et ses bases ainsi que GitHub qui nous permet en toute simplicité de pouvoir développer un projet de groupe.

Ce rapport de soutenance présentera les diverses fonctionnalités décrites dans notre cahier des charges, notre avancement sur ces dernières et le rôle de chacun des membres du groupe dans leur réalisation.

2 Modifications du cahier des charges

Le groupe n'a en rien changé en termes de membre, nous sommes toujours 4.

- Alexandre DUGOT (chef de projet)
- Antoine ADAM
- Mohamed GHAFfour
- Nathan LABERNARDIERE

La répartition des tâches a légèrement changé par rapport à ce que nous avions prévu de base. Les personnages étant initialement la tâche de Nathan, cette tâche a finalement changé et est maintenant sous la responsabilité d'Alexandre. La gestion des effets sonores a,

elle aussi, été inversée, passant d'Alexandre à Nathan. Ces changements ont eu lieu de part de la prise de conscience que certains étaient plus à l'aise dans d'autres tâches. Voici donc notre tableau final:

Tâches	Antoine	Nathan	Alexandre	Mohamed
Sauvegardes	x		o	
Missions / histoires			o	x
Structure du vaisseau	o	x		
Machines / inventions	x			o
Personnages		o	x	
Gestion / design planètes		x	o	
Site Internet		o		x
Effets sonores		x		o
Cinématiques	o			x
Interface carte	o		x	
Multijoueur	x		o	

Pour le moment il n'y a que les bases du jeu qui sont faites, donc pour le moment nous ne pouvons pas encore envisager tous les futurs problèmes qu'il pourrait y avoir. Par conséquent, tout autre changement non cité dans ce rapport pourrait apparaître seulement plus tard en fonction des difficultés rencontrées.

Dans la partie 6 (**Avances et retards**) du rapport, nous récapitulerons le tableau d'avancements qui a été modifié.

3 Avancements individuels sur les fonctionnalités

L'avancement de chaque fonctionnalité du projet sera détaillé ci-dessous dans les diverses sous-parties. Pour chaque sous-partie, tous les aspects du fonctionnement seront présentés rapidement avec quelques explications techniques. Les responsables et suppléants des tâches en question seront précisés à chaque fois.

Certaines tâches vont paraître courtes dans leur présentation et leurs explications, car certaines n'ont pas encore été implémentées. Bien sûr, celles-ci seront détaillées en conséquence pour les soutenances à venir.

3.1 Personnages

Nous avons choisi de dessiner un petit astronaute comme personnage. Le design de ce personnage est assez classique et peut ressembler à des design assez récurrents sur Internet. Pour animer notre personnage, il nous a fallu réaliser cinq positions différentes. Quatre de ces cinq positions correspondent au déplacement de l'astronaute sur le sol et le cinquième correspond à l'animation du saut. Ces animations sont plutôt basiques mais elles permettent quand même à notre protagoniste de se déplacer de façon fluide. Pour le moment, nous avons réalisé deux personnages, la seule différence entre eux réside dans leur colorimétrie.

Voici le rendu obtenu:



Vous voyez donc ici deux exemples d'animation qui ont été implémentés et qui sont exécutés en fonction de l'action du personnage. Pour l'animation, cela consiste à faire défiler une suite d'images très rapidement afin de pouvoir créer l'illusion d'un personnage qui court ou qui saute.

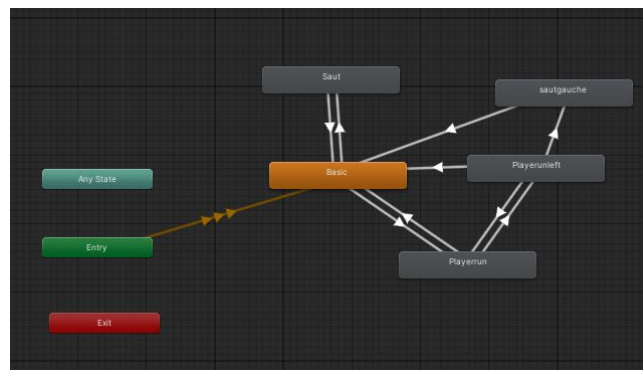
De base les animations ont été faites seulement pour se déplacer du côté droit, nous avons donc dû utiliser une fonction mirror qui permet de retourner les images pour que le personnage obtient un effet de mouvement vers la gauche.

Pour détecter si le joueur va à gauche ou à droite, nous avons utilisé une propriété de la librairie physics2D qui consiste au fait que lorsque le personnage se déplace à gauche la vitesse est négative et à droite positive. Il nous suffit donc juste de connaître le signe de la valeur de la vitesse.

Pour contrôler le joueur il y a pour le moment trois touches, dont les flèches droite et gauche qui permettent respectivement d'aller à gauche et à droite, il y a également la barre d'espace qui permet de sauter.

Pour faire en sorte que le joueur ne puisse pas faire de doubles sauts nous avons fait en sorte de détecter grâce aux hitbox du sol et du personnage si le personnage se trouve au sol ou non. Si le personnage n'est pas au sol, c'est qu'il est en train de sauter donc ce dernier ne peut donc plus sauter avant d'être revenu au sol. Pour cela la commande ci-dessous a été utilisé:

Physics2D.OverlapArea(groundCheckLeft.position,groundCheckRight.position);.



Ci-dessus vous pouvez voir la façon dont les animations du joueur sont gérées, le rectangle vert correspond simplement au fait d'entrer dans l'animation quand on lance le jeu, le rectangle orange lui correspond à l'état initial du personnage (quand il ne bouge pas).

Enfin, les rectangles gris correspondent chacun à une animation en particulier du joueur, ces dernières sont reliées à l'état initial avec des transitions qui consistent en un ensemble de conditions qui font que telle ou telle animation se déclenche.





Pour finir sur cette longue partie en lien avec les personnages, nous allons voir la barre de vie qui grâce à Unity a été assez simple à gérer. Cela consiste en la création d'un "canvas" sur lequel nous avons placé une image que l'on peut voir ci-dessus dont la barre rouge a une taille variable entre 0 et 100 qui correspond au nombre de points de vie. Nous allons donc faire varier cette valeur dans nos scripts en fonction des dégâts que reçoit le joueur, quand la barre est vide une animation de mort entre en scène.

3.2 Gestion des sauvegardes

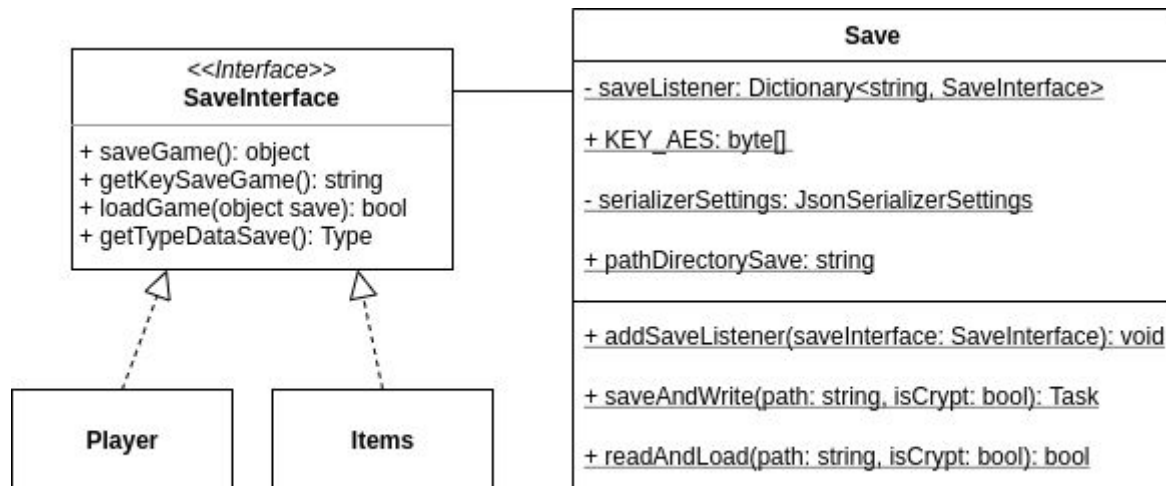
L'intérêt des sauvegardes dans ce jeu est de pouvoir à la fois partager une partie en cours et de pouvoir faire des pauses au milieu d'une partie sans perdre son avancement, elles peuvent aussi servir indirectement de retour en arrière.

Nous allons aussi beaucoup les utiliser pour le debug du jeu en provoquant des situations inattendues, tout en pouvant les recréer rapidement autant de fois que nécessaire pour les corriger.

Lors des recherches plusieurs possibilités de stockage des données se sont présentées, elles étaient déjà implémentées dans Unity mais inadaptées à des items et machines gérés de façon dynamique, Il était donc plus préférable de développer un système de stockage plus bas niveau pour avoir plus de maîtrise sur celles-ci.

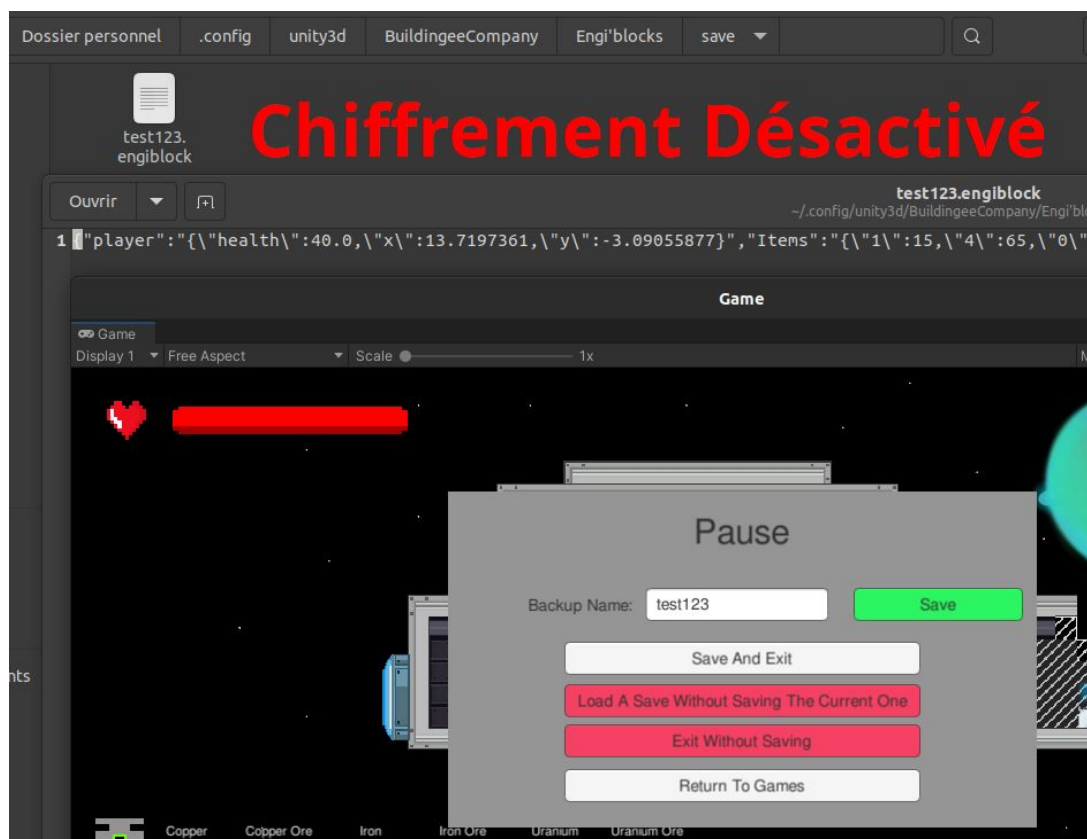
Le système de sauvegarde fonctionne en implémentant une interface au class stockent des données à enregistrer et d'ajouter leur objet dans un listener de la class Save, au moment de sauvegarder les méthodes implémentées des objets seront utilisées pour traiter de façon personnalisée les données puis elles renverront un objet qui sera sérialisé en json, cette chaîne de caractère en format json sera ensuite chiffrée en AES et en fin le résultat serait enregistré dans un fichier dans un répertoire donné par Unity (pour gérer le multi OS)

Le chargement fait exactement les étapes inverse, mais avec quelque particularité liée au parsing du json, Le problème est de savoir quel type de données renvoyer en sachant que c'est des types assez complexes (Dictionary<int, int> par exemple), la solution a été implémentée aux objets une méthode qui renvoie le type attendu et de leur renvoyer un objet qu'elles créeront.



(Représentation en UML partielle et simplifiée)

Le chiffrement des sauvegardes à uniquement pour but d'évité leur modification, l'utilisateur ne pourra pas facilement provoquer une situation inattendue (un json non conforme provoqueras une erreur gérée), les utilisateurs seront moins tentés de tricher simplement en modifiant un json, ils devront décompiler le jeu pour obtenir la clé AES hardcodé et ensuite déchiffrer, modifier et rechiffrer ce qui est beaucoup plus dur.



3.3 Gestion du vaisseau et design

La partie graphique de notre jeu a entièrement été réalisée à l'aide de Photoshop. Ainsi, aucune image n'a été prise d'Internet. Néanmoins, pour certains designs, nous avons repris l'aspect graphique notamment ceux de Starbound qui utilise la même perspective d'affichage.

Cette partie a été relativement longue, car dans un premier temps, nous voulions que le joueur ait une vision de haut, plongeante. Ainsi, nous avons commencé à réaliser des croquis de ce point de vue, or, après quelques réflexions, nous avons pris la décision que le joueur aurait une vision de profil sur le jeu. Il a fallu alors revoir nos assets graphiques. Plusieurs dessins au brouillon et essais sur Photoshop ont été nécessaire afin d'aboutir à ce résultat:



L'aspect global du vaisseau est relativement simple, pas de surfaces rondes ou lisses, car notre objectif est de rester dans un univers que l'on pourrait qualifier de pixelisé. Pour ce vaisseau réparti sur deux étages, nous avons choisi des couleurs plutôt métalliques pour la surface extérieure du vaisseau et des couleurs relativement sombres pour en représenter l'intérieur.

Pour rendre ce vaisseau plus vivant nous avons également réalisé un réacteur dynamique, c'est-à-dire que nous faisons défiler 3 images différentes du réacteur. Cela peut sembler négligeable cependant cet effet améliore le réalisme du jeu.

3.4 Site Internet

-Mohamed GHAFfour
-Nathan LABERNARDIERE

Le site internet constitue indirectement la vitrine de notre projet, c'est donc une étape importante. Pour le moment, ce dernier est assez simple.

Le site internet est hébergé grâce à GitHub Pages et est accessible publiquement à cette adresse : <https://engi-blocks.github.io/site/>.

Pour réaliser le site nous avons choisi de le faire nous-même dans son entièreté. Pour cela, nous utilisons Bootstrap avec pour langage l'HTML auxquelles nous avons appliqué un design grâce aux CSS. Le site réalisé est un site web statique et non dynamique, car nous avons jugé que faire un site web statique était le plus approprié à notre situation. La prise en main au départ était assez étrange, car personne dans notre groupe n'avait encore touché à ce type de programmation. Cependant, il y a une documentation assez large sur Internet, de plus, on peut retrouver de nombreux tutoriels que ce soit sur YouTube ou sur certains sites pour prendre en main Bootstrap.

Une fois les bases acquises nous avons commencé à nous pencher sur l'esthétique de notre site. Nous cherchions à faire quelque chose d'assez épuré et agréable à l'œil. Ainsi, après plusieurs tentatives et une bannière réalisée sur Photoshop nous sommes parvenus à ce résultat:



Ce thème sombre est immersif permettant de plonger la personne entrant sur notre site directement dans notre univers. De plus, nous avons décidé de placer 5 rubriques cliquables dans notre site que sont: “Accueil”, “Téléchargements”, “Règles du jeu”, “Nouveautés” et enfin “Qui sommes-nous?”. Elles n’ont pas encore été implémentées, c’est-à-dire que pour le moment elles nous redirigent uniquement vers une page blanche avec le menu du site. Nous avons déjà commencé à rédiger quelques notes afin de nourrir ce site. Des captures d’écran du jeu permettent d’illustrer la catégorie “Nouveautés” afin de montrer l’évolution du développement du projet, cette rubrique nous permettra également d’exposer les différents problèmes rencontrés au cours de ce dernier.

3.5 Multijoueur

Au départ, nous avons rédigé dans notre cahier des charges que nous allions commencer à gérer la partie multijoueur de notre jeu dès la première soutenance, or, nous avons dû changer nos objectifs. En effet, nous ne pouvons pas commencer cette tâche dès la première soutenance, ainsi, nous la commencerons pour la seconde soutenance.

3.6 Effets sonores

Cela peut paraître être une partie négligeable et effectivement ce n’est pas la plus importante, mais nous pensons qu’un jeu dénué de tout son est plutôt triste. C’est pour cela que quasiment toutes les actions ainsi que la cinématique de début de jeu seront accompagnées de divers effets sonores ainsi que de voix.

Mais cette tâche n’a toujours pas été commencée, car avions prévu de ne pas entamer cette partie lors de la première soutenance. Pour autant, nous avons quand même commencé à réfléchir où trouver nos sons et effets sonores (libres de droits) et rechercher ensuite comment nous pourrions les implémenter par la suite. En effet, nous ne pensons pas que nous allons réaliser ces sons par nos propres moyens, car personne dans notre groupe ne possède d’expérience dans la musique.

3.7 Interface carte

Le but de cette partie est d'obtenir une carte que l'on peut ouvrir depuis le poste de pilotage à l'avant. À partir de cette dernière, il y aura possibilité d'accéder aux autres systèmes stellaires qui seront déblocables au fur et à mesure en ayant accompli les missions.

Pour le moment tout n'est pas encore implémenté, pour l'instant le joueur peut accéder à la carte grâce à la touche "M" quand il est dans la "boxcollider2D" du poste de pilotage, cette touche permet d'obtenir une image au centre de l'écran représentant la carte. Pour quitter la carte ce dernier peut appuyer sur la touche "ESCAPE". Pour le moment il n'y a que cela qui est implémenté, car nous attendons d'avoir terminé les missions étant nécessaires aux conditions de la carte pour pouvoir commencer à y ajouter les divers systèmes. Pour afficher la carte à des moments voulus nous avons utilisé la commande **InteractUI.enabled = false;** où le interactUI correspond au tag du fichier image/texte à afficher et ensuite le enabled qui étant faux ou vrai affiche ou non la carte ou le texte.

Pour détecter lorsque le joueur appuie sur une touche en question nous avons utilisé la commande **Input.GetKeyDown(KeyCode.example)** qui permet de détecter quand le joueur appuie sur la touche en question et renvoie un booléen.

Voici donc un rapide aperçu ci-dessous:



3.8 Cinématiques

Nous n'avons encore rien effectué pour cette tâche, car nous comptons la commencer pour la prochaine soutenance. En ce qui concerne la façon de les réaliser, nous hésitons encore entre deux méthodes. Soit, nous enregistrerons une vidéo pour la faire défiler aux moments voulus soit, nous créerons une séquence d'images à partir d'Unity. Nous envisageons également d'enregistrer des voix off à écouter lors de ces cinématiques.

3.9 Missions et histoires

La partie Missions et Histoires constitue une étape importante du projet, en effet selon l'originalité et la qualité de ces dernières le joueur sera plus au moins en immersion.

Il nous a semblé évident que nous devions d'abord avoir en tête l'histoire du jeu, pour ensuite pouvoir enfin imaginer des missions qui donneront envie au joueur de progresser.

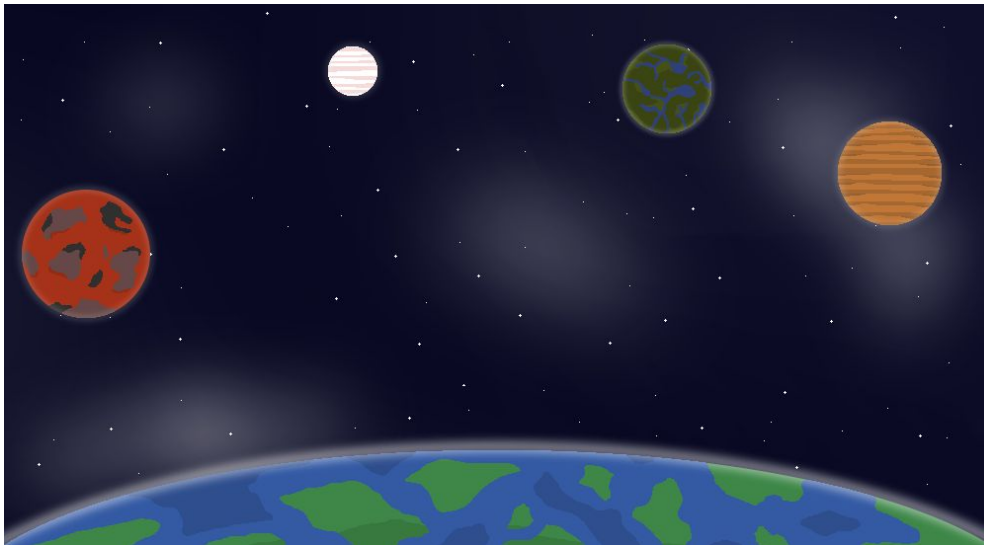
A la date de cette première soutenance de nombreuses missions ont vu le jour et l'histoire du jeu commence réellement à prendre forme. Cependant, pour implémenter ces missions, il est préférable d'attendre la version définitive des machines et de la carte, en effet ces dernières peuvent encore évoluer et donc donner place à d'autres missions ou à sens inverse mener à une incohérence.

Nous avons tout de même commencé à réfléchir à un schéma qui nous aidera à implémenter les missions. Si toutes les conditions ont été validées, le joueur pourra évoluer d'un niveau et atteindre une nouvelle planète.

3.10 Gestion/Design planètes

Cette partie consistait en la réalisation des différents décors autour du vaisseau lors de changements de planètes. Nous avons dessiné sur Photoshop un fond étoilé avec quelques planètes au loin et la planète sur laquelle nous nous situons au premier plan. Ces décors sont particulièrement longs à réaliser car le plan est grand. Ce qui est le plus compliqué dans ces tâches là est d'avoir beaucoup de créativité et cela peut prendre parfois du temps pour trouver la bonne inspiration.

Au bout de plusieurs tentatives, voici un exemple de décor obtenu:



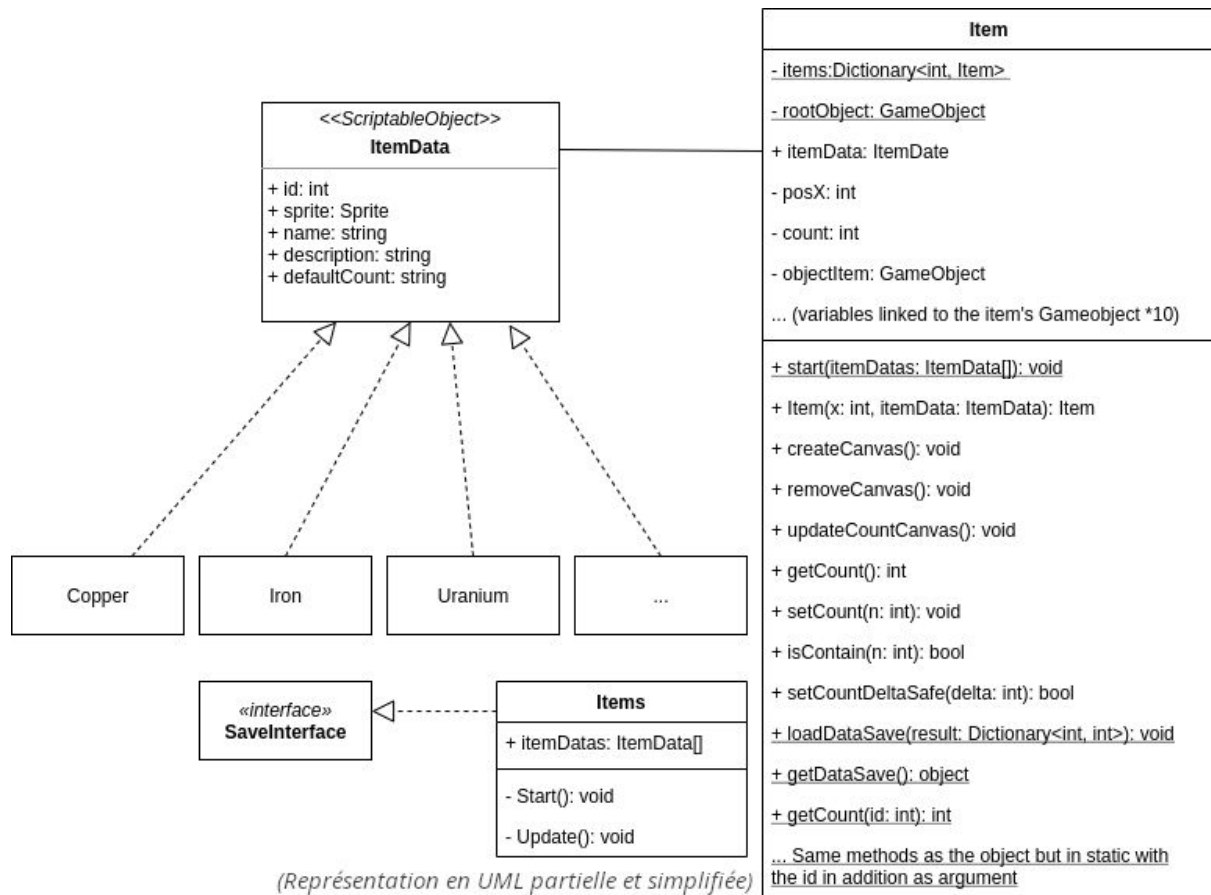
Nous sommes plutôt satisfaits du résultat obtenu, car il s'approche assez de nos attentes sur les décors. Nous trouvons que le tout est immersif, mais nous hésitons encore à ajouter quelques effets dynamiques dans ce décor pour améliorer l'immersivité du joueur ainsi que le réalisme. Par la suite, en association avec la partie carte, il faudra faire en sorte que le fond et la planète changent quand on change de système, pour le moment la planète ci-dessus n'est que la première planète rencontrée.

3.11 Machines / inventions

Cette partie a pour but d'ajouter de la stratégie et d'augmenter la durée de jeu, pour y parvenir le joueur devra récolter des minerais sur les planètes au fur et à mesure de son avancement, les traiter avec des machines d'enrichissement et de purification pour les utiliser en réparant, en améliorant ou en ajoutant des machines, ces machines sont vitales pour le joueur puisqu'elles lui permettent de recycler son air, se soigner, propulser son vaisseau et pleins d'autre chose.

Le premier aspect développé sont les items, Pour simplifier la suite et permettre des ajouts facile, les items sont gérés dynamiquement en utilisant les ScriptableObjects d'Unity, Ils créent de petit fichier contenant tous les paramètres différenciant entre des objets dont la logique est la même.

Au chargement du jeu tous les ScriptableObject d'items sont utilisés pour créer des objets item contenant en plus l'information de leur quantité, les méthodes de ces objets sont ensuite utilisées pour gérer l'interface de l'inventaire principal.



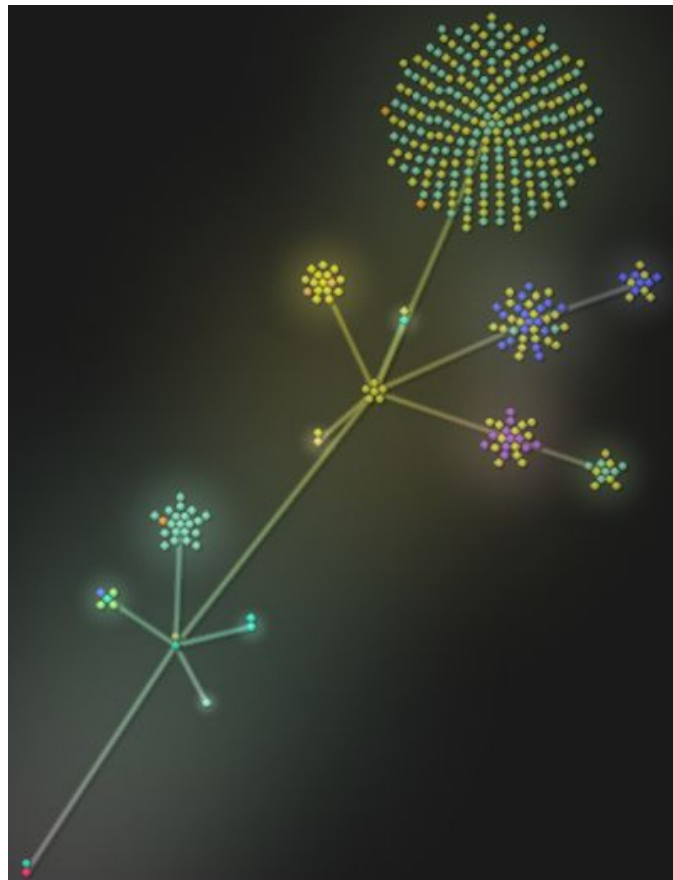
Il y a un problème qui serait résolue dans la soutenance suivante, la gestion graphique n'est pas séparé de la class item, comme il y aura sûrement un peu plus d'item dans le futur, l'inventaire principal affiché en permanence en bas de l'écran seras vite surchargé, pour régler ce problème, l'inventaire principal affichera uniquement les items importants, le reste seras visible via une machine (un coffre).



La jauge à gauche est l'indicateur de dioxygène de l'air du vaisseau, elle sera liée au fonctionnement du recycleur d'air.

Les machines ne sont actuellement pas développées, mais les recherches et la structure des classes sont réalisées, pour les machines les ScriptableObject sont pas viables, il y a beaucoup trop de logique différentes, il y aura donc une classe abstraite machine qui contiendra toute la logique et les paramètres identiques, les machines seront des classes filles, les machines seront gérées dynamiquement comme les items avec leur classe abstraite pour éviter une complexité et une redondance sur l'interface graphique.

4 Structure du répertoire



La structure du repository Git, c'est-à-dire du projet, est représentée par cet arbre ci-dessous. Cette visualisation a été générée à l'aide de l'outil Gource à partir de l'historique de modification Git. Chaque branche représente un dossier, chaque élément représente un fichier et chaque couleur représente un type de fichier.

5 Expériences personnelles

Ce projet qu'est ce jeu vidéo est d'une dimension bien supérieure en termes de travail à fournir que tout ce que nous avons pu entreprendre depuis le début d'année. Cela nous a donc montré à nous 4 que des jeux auxquels nous jouons sans trop savoir comment ils sont fait sont en fait de véritable projet long et parfois éprouvant mais malgré cela nous nous amusons

beaucoup dans cette expérience. Il y a plusieurs raisons à cela, premièrement nous sommes libres d'entreprendre ce que nous voulons tant que cela est codé dans le langage C#, deuxièmement, c'est un travail de groupe ce que nous trouvant motivant car à chaque fois que l'on va voir l'avancement du projet nous sommes impatient d'en voir les avancées et enfin cela nous permet d'apprendre à utiliser les applications GitHub et Unity qui nous seront sans doute utiles pour la suite de notre carrière. Pour en finir avec cette partie nous pensons qu'il est primordial de savoir gérer une situation de groupe ainsi qu'une présentation orale très tôt dans nos études, car à l'avenir cela sera sûrement la base de notre travail et deviendra une habitude à prendre.

6 Avances et retards

Par rapport aux prévisions présentées dans le cahier des charges, nous pouvons noter quelques retards dans les domaines suivants:

- La gestion du multijoueur
- La gestion des missions

Nous pensions à la base implémenter le début du multijoueur de notre projet avant la première soutenance, mais finalement nous nous sommes confortés dans l'idée qu'il valait mieux faire fonctionner le reste, c'est-à dire la partie solo avant d'étudier la question d'un potentiel mode multijoueur. Pour les missions, nous préférons commencer ces dernières lorsque le système de carte et les machines seront opérationnelles. Malgré cela les idées de bases sont déjà là, notamment pour les missions dont nous avons déjà une claire idée de la façon de les faire et surtout en quoi elles vont consister et ce qu'elles vont apporter dans le contexte du jeu.

Cependant, malgré certains retards, nous avons pris beaucoup d'avance dans certains domaines dont les suivants en particulier.

- Gestions des sauvegardes
- Gestion des personnages
- Le site internet

Le site internet ne devait être créé que pour la seconde soutenance en grande partie, mais finalement nous avons trouvé le temps de le faire, mais ce dernier n'est pas encore terminé. Pour ce qui est du personnage, l'implémentation de la barre de vie n'était pas prévue pour tout de suite, mais nous l'avons finalement fait ainsi qu'une barre d'oxygène qui quand elle est vide fait baisser la vie du personnage. Bien entendu cela n'est qu'une partie de ce qui devra être fait pour finaliser le jeu.

7 Prévisions pour la suite

Tout d'abord, avant d'expliquer nos prévisions pour la future soutenance, nous allons rappeler notre tableau de répartition des tâches ainsi que notre tableau d'avancement avec les modifications effectuées (en **vert** pour les augmentations et **rouge** pour les diminutions) depuis le cahier des charges:

Tâches	Soutenance n°1	Soutenance n°2	Soutenance n°3
Sauvegardes	75%	95%	100%
Missions / histoires	30%	70%	100%
Structure du vaisseau	50%	80%	100%
Machines / inventions	25%	75%	100%
Personnages	40%	75%	100%
Gestion / design planètes	20%	50%	100%
Site Internet	20%	90%	100%
Effets sonores	0%	40%	100%
Cinématiques	0%	50%	100%
Interface carte	40%	75%	100%
Multijoueur	0%	10%	100%

Nous avons donc changé nos objectifs d'avancement sur les tâches suivantes: gestion des sauvegardes, machines et inventions et multijoueur. En effet, réaliser un tableau d'avancement aussi complexe n'était pas une tâche simple étant donné que nous ne n'avions jamais réalisé de projet tel auparavant. Ainsi, nous nous sommes rendus compte du changement nécessaire de ces tâches qu'à partir du moment où nous avons suffisamment avancé dans notre projet.

En ce qui concerne désormais nos avancements pour la suite, voici une liste regroupant toutes les prévisions des tâches en expliquant brièvement ce qui sera effectué:

-sauvegardes:

Pour finir il sera possible de charger une sauvegarde depuis le menu pause du jeu ainsi qu'à partir du menu principal du jeu.

-missions / histoire:

Les missions, à terme, consisteront en une suite de conditions qui devront être remplies. Lorsqu'elles seront remplies ces dernières donneront accès aux nouvelles planètes, il faudra donc implémenter cela dans la carte pour savoir si les tâches sont validées. Pour finir elles pourront également faire gagner des ressources ou des avantages.

-structure du vaisseau:

Nous aimerions implémenter des vaisseaux marchands ou bien des vaisseaux ennemis pour cela il faudra réaliser de nouveaux design de vaisseaux qui seront assez semblables à celui que possède le joueur.

-machines / inventions:

Dans cette partie nous avons encore une barre d'énergie à implémenter ainsi que la possibilité de créer des machines et de les poser à partir des diverses ressources dont l'interface a déjà été implémentée.

-personnages:

Il y a encore quelques bugs d'animations avec le personnage qui doivent être réglés, notamment lorsque ce dernier saute et avance vers la gauche donc ce problème doit à terme être réglé. Pour le moment seul le personnage principal est implémenté, mais nous aimerions qu'à chaque changement de planète il y ait des monstres qui apparaissent pour mettre le joueur en difficulté. Ces monstres devront être des petites IA qui suivront le joueur et auront pour but de suivre le joueur pour lui faire des dégâts.

-gestion / design planètes:

Il faudra effectuer de nouveaux décors pour faire varier le plus possible l'environnement dans lequel le joueur évoluera. Cela permettra aussi de rendre le jeu un peu moins redondant et permettre au joueur de ne pas s'ennuyer.

-site Internet:

Pour le moment le site internet est purement visuel, mais à terme, il faudra que l'on puisse télécharger le jeu depuis celui-ci. Il y aura aussi un wiki avec les règles du jeu pour que toute nouvelle personne puisse y jouer s'il le veut. Il y aura pour finir quelques images/pub en rapport avec le jeu pour donner envie d'y jouer.

-effets sonores:

Pour le moment nous ne considérons pas cela comme important, mais il faudra sur chaque action y implémenter un bruit correspondant ainsi qu'une musique de fond. Cela étant plutôt rapide et facile à faire, nous le ferons donc à la toute fin.

-cinématiques:

Il faut que lorsque le joueur lance le jeu et le termine qu'il y ait une cinématique pour introduire l'histoire présente dans le jeu ainsi que pour conclure cette histoire de façon à ce que le joueur ait une belle impression de finalité.

-interface carte:

Pour le moment, comme pour le site internet cette dernière n'est que visuelle, le but est donc pour la prochaine fois de rajouter les boutons pour que le joueur puisse aller de planète en planète, mais aussi ajouter la partie des conditions, c'est-à-dire que l'on ne pourra cliquer que lorsque certaines conditions seront remplies.

-multijoueur:

Il faudra que plusieurs joueurs puissent rentrer dans un même vaisseau et faire en sorte qu'étant en multijoueurs, les éléments de gameplay devront être adapté (Ex: il devra y avoir plus de monstre en fonction du nombre de joueurs). Nous ne savons pas encore par contre combien il pourra y avoir de joueur en même temps.

8 Conclusion

En définitive, malgré un projet qui semble bien lancé, il nous reste bien des choses à implémenter. Mais au moins maintenant nous avons une base solide pour notre jeu, ce qui nous permet désormais de pouvoir implémenter de plus en plus d'éléments de gameplay. Les débuts étaient difficiles, car nous ne connaissions pas bien l'application Unity mais à force de pratique nous nous sommes améliorés et les connaissances acquises nous permettent à présent d'implémenter quasiment toutes les fonctionnalités voulues.

Ce projet est également un très bon moyen d'améliorer nos compétences dans l'utilisation du C# à travers une plateforme comme Unity. Et n'oublions pas l'outil Git qui est un outil puissant qui nous sera utile dans notre futur dans les entreprises ou autre projet en lien avec EPITA. Pour en finir nous sommes plutôt optimistes quant à l'avancée du jeu et nous trouvons que l'état actuel des choses est positif.