

Projet Engi'Blocks

Rapport de deuxième soutenance

Groupe Buildingee

Antoine Adam
Alexandre Dugot
Mohamed Ghaffour
Nathan Labernardiere

30 avril 2021



Table des matières

1 Introduction

2 Avancements individuels sur les fonctionnalités

2.1	Personnages
2.2	Gestions des sauvegardes
2.3	Gestion du vaisseau et design.
2.4	Site Internet
2.5	Menu Principal.
2.6	Multijoueur.
2.7	Sons / effets sonores.
2.8	Interface carte.
2.9	Cinématiques
2.10	Missions et histoire.
2.11	Gestion / design planètes
2.12	Machines/inventions.

3 Structure du répertoire

4 Expériences personnelles

5 Avances et Retards

6 Prévisions pour la suite

7 Conclusion

1 Introduction



Le projet depuis la première soutenance a connu de nombreux changements qui rendent désormais le jeu de plus en plus jouable, bien que le fil de l'histoire du jeu ne soit pas encore présent. Actuellement le vaisseau lui-même paraît plus complet et accueillant et le développement du menu principal rend le tout plus présentable. Néanmoins, il manque encore un certain nombre de finitions pour que notre jeu paraisse plus qualitatif.

Il est d'ailleurs intéressant de préciser que diverses connaissances acquises lors des derniers TP de programmation nous ont quelque peu aidé pour certaines tâches, mais aussi pour optimiser notre code.

2 Avancements individuels sur les fonctionnalités

2.1 Personnages

Lors de la première soutenance, nous avons seulement abordé la partie développement du personnage qu'est l'astronaute et que le joueur contrôlera, c'est-à-dire ses déplacements et sa barre de vie. Cette fois-ci, il est question de gérer des IA qui sont en fait des monstres contre lesquels le joueur devra se défendre grâce à diverses attaques qui, elles aussi, ont été codées. Pour le moment ce ne sont que de simples IA, ces dernières peuvent seulement suivre le joueur, lui infliger des dégâts ou encore pouvoir juger à quel étage du vaisseau elles se trouvent et aviser quant au comportement à avoir pour atteindre et attaquer le joueur. Comme vous pouvez le voir ci-dessous nous avons implémenté trois types de monstres qui chacun ont leurs spécificités.



Le monstre en haut à gauche a pour comportement simple de toucher le joueur pour lui faire des dégâts et lorsque ce dernier tombe à 25% de sa vie totale, il se transforme en slime rouge qui aura une vitesse et des dégâts accrus. Pour cela il suffit juste de regarder en permanence la vie du monstre dont l'argument privé est situé dans la classe du monstre et au moment voulu changer l'image et l'animation du petit monstre. Pour le l'alien vert foncé en bas le comportement est quasiment le même au détail près qu'il peut sauter pour suivre le joueur lorsqu'il saute.

Pour pouvoir faire en sorte que le monstre suive le joueur nous avons besoin de connaître la position du joueur en permanence pour savoir si le monstre doit changer d'étage ou si ce dernier doit aller à gauche ou à droite. Pour cela voici un petit bout de code pour expliquer cela.

```
Y = GameObject.Find("perso_0");  
y1 = Y.transform.position.y;  
y2 = this.gameObject.transform.position.y;
```

Tout d'abord, la première ligne sert à partir du script d'un monstre de pouvoir obtenir la référence au joueur dont la référence d'objet est "perso_0" pour pouvoir modifier ou obtenir ses paramètres. La seconde partie permet à partir de la variable y qui représente le joueur de pouvoir obtenir la position en y du joueur. Quant à la dernière ligne, elle récupère la position en y du monstre auquel est attaché le script en question, d'où l'utilisation du "this".

C'est ainsi par ce principe simple que l'on récupère les positions et que par de multiples cas nous pouvons gérer les mouvements et les rendre un minimum logique pour éviter que le monstre ne fasse que suivre bêtement le joueur sans le faire efficacement.

Lors de la dernière soutenance nous avons déjà implémenté la barre de vie du joueur présente en haut à gauche mais surtout, la méthode "Take damage" auquel on donne un argument de type Int et qui permet d'enlever un certain nombre de points de vie au joueur. Il nous a donc suffi d'utiliser cette méthode lorsque le monstre est suffisamment proche du joueur pour lui infliger des dégâts.

En début de chaque aura des capacités de base (points de vie, attaque) mais ces dernières augmentent au fur et à mesure de la partie pour en augmenter la difficulté.

Pour finir j'aimerais évoquer quelques difficultés qu'il y a pu avoir dans le développement de cette partie. Premièrement, la fonction qui gère le fait que le monstre ne peut attaquer que toutes les deux secondes devait être écrite en utilisant la méthode "System.Threading.Thread.Sleep(); \$" , or il s'est avéré que cela ne marchait pas car cette dernière bloque l'ensemble du jeu pendant 2 secondes. Nous avons donc dû utiliser la méthode "Time.time" qui va donner l'heure universelle précise en seconde, il suffit ensuite par la suite de donner la valeur Time.time à une variable et pour finir on autorise le monstre à faire des dégâts seulement quand on aura atteint Time.time + 2.

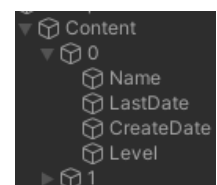
Et enfin la seconde et quant à elle liée à l'animation dont la fluidité est compliquée à gérer, d'ailleurs nous tenons à signaler qu'en l'état actuel des choses , les animations ne sont pas encore excellentes.

2.2 Gestion des sauvegardes

À la dernière soutenance le menu pause était créé et fonctionnel, il permet de sauvegarder, naviguer entre les menus, mais il était impossible d'utiliser une sauvegarde, pour cette soutenance un nouveau menu a été créé pour pouvoir sélectionner et charger une sauvegarde.



Ce menu n'utilise pas de module menu d'Unity dans le but d'être personnalisable le plus possible, Chaque élément (les blocs avec les noms des sauvegardes) est généré en C# en lisant le dossier de sauvegarde.



Le nom de sauvegarde est celui du fichier, la dernière modification ("Last:") est obtenue en récupérant la date de création du fichier parce qu'il est écrasé à chaque sauvegarde, le reste des informations est stocké dans la sauvegarde.

Le nom de la sauvegarde est protégé contre les caractères spéciaux qui ne peuvent pas être utilisés dans le nommage d'un fichier seule l'alphabet minuscule, majuscule et chiffre sont acceptés. La taille du nom est aussi limitée à 32 caractères.

2.3 Gestion du vaisseau et design

Cette fois-ci, nous nous sommes particulièrement focalisés non plus sur l'aspect extérieur du vaisseau, mais sur ce qu'il contient. Pour cela, une fois de plus nous avons utilisé Photoshop et nous avons dessiné des objets décoratifs et des objets utilisables (tels qu'un poste de commande, un fourneau...). De plus, certains de ces objets utilisables possèdent une animation, mais nous verrons ça plus en détail dans la partie **3.12 Machines / inventions**.

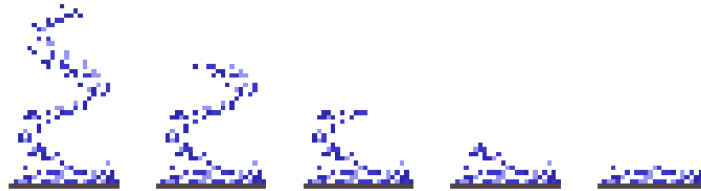
Ainsi, voici un exemple de résultat sur Photoshop une fois ces éléments placés (ce placement n'est pas forcément représentatif, il permet simplement d'avoir en un premier temps un aperçu de rendu) :



Nous sommes plutôt ravis du rendu que donnent ces objets et machines une fois placés car l'espace du vaisseau est enfin utilisé. Nous trouvons que le tout est bien plus immersif qu'auparavant et une fois en jeu le vaisseau paraît vivant.

Par ailleurs nous pouvons voir à la gauche du vaisseau un portail bleu. Ce portail permet de téléporter les joueurs à l'étage supérieur du vaisseau. Pour ce faire on regarde la position du

joueur en x et en y pour savoir si ce dernier est bien dessus et s'il est bien à l'étage du bas pour que ce dernier en appuyant sur la touche U ne se retrouve pas à voler s'il n'est pas sur le téléporteur. Il possède une animation en jeu permettant d'améliorer le réalisme en voici les différentes tuiles :



2.4 Site Web

Le site internet constitue indirectement la vitrine de notre projet, c'est donc une étape importante. Une fois les bases acquises en HTML et en CSS nous avons pris comme décision d'oublier Bootstrap et de coder entièrement notre site. Certes l'utilisation de Bootstrap aurait pu nous faire gagner beaucoup de temps mais nous pensons que le mérite a plus d'importance lorsque le site est codé entièrement par nos soins.

Une fois les connaissances acquises nous nous sommes penchés sur l'esthétique de notre site, nous avons décidé de placer 5 rubriques cliquables dans notre site qui sont : "Accueil", "Téléchargements", "Le jeu", "Images du jeu" et enfin "Qui sommes-nous ?". Toutes les rubriques sont déjà disponibles cependant la rubrique "Téléchargements" n'est pas encore finalisée. Chaque rubrique a pour but de mettre le futur joueur dans le contexte de notre jeu.

Voici ci-joint la page d'accueil du site :



Le site Engi'blocks a continué de représenter notre jeu sur la toile internet !

- Maintenu à jour assez régulièrement

Le plus important pour notre site web était qu'il reste à jour. Ainsi nous avons continué de poster des nouvelles de notre avancement sur la rubrique "Le Jeu". Les documents et exécutable de la 2e soutenance ont bien entendu été ajoutés à la partie téléchargement, en même temps que la publication de nouvelles images du jeu.

- Réorganisation

Le contenu du site étant de plus en plus conséquent, nous avons procédé à quelques réorganisations.

La partie Téléchargement a été réorganisée en deux parties :

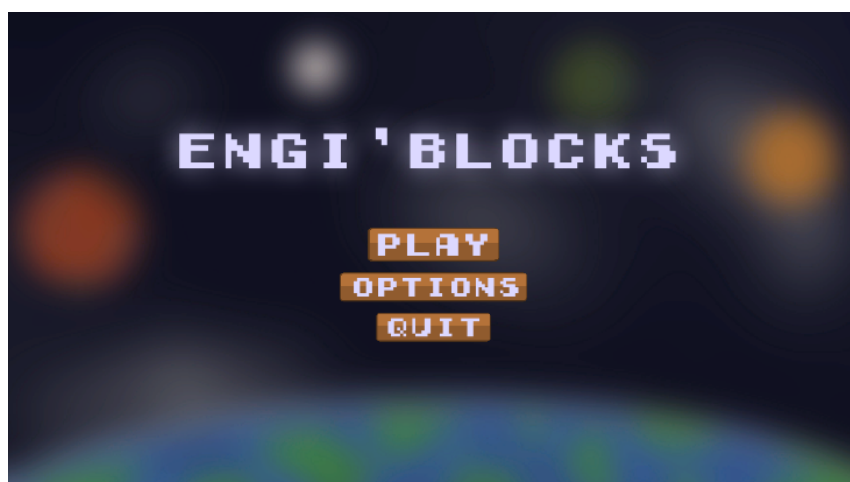
– Téléchargements du jeu : Cette partie permet de télécharger la dernière version publique de Engiblocks ;

– Soutenances : Cette dernière partie propose notre cahier des charges ainsi que tous les documents des deux soutenances précédentes.

Les autres parties ont été moins modifiées.

2.5 Menu principal

Pour cette soutenance, il nous était obligatoire de créer un menu principal pour pouvoir avancer dans nos tâches. Voici le rendu que nous sommes parvenus à obtenir :



Cela ne peut pas se voir sur la capture d'écran, mais nous avons également décidé de rendre le titre de notre jeu dynamique pour rendre le menu principal plus vivant (le titre avance et recule lentement).

Ainsi, ce menu propose trois options cliquables par le joueur que sont :

- PLAY : Cela permet pour le moment de renvoyer le joueur sur la scène principal (là où le jeu se déroule) avec le menu des sauvegardes (cf. **2.2 Gestion des sauvegardes**).

- OPTIONS : Ce bouton est accessible depuis le menu principal ainsi qu'en jeu. Une fois cliqué cela ouvre une fenêtre proposant différentes options aux joueurs. Pour des soucis d'esthétisme, le titre du jeu dynamique en fond est désactivé lorsque cette fenêtre est ouverte car sinon l'on verrait quelques lettres du titre dépasser. Ainsi, les joueurs peuvent choisir s'ils veulent jouer en fenêtré ou plein écran (le jeu sera toujours lancé automatiquement en plein écran) ou alors augmenter ou diminuer les sons et effets sonores.

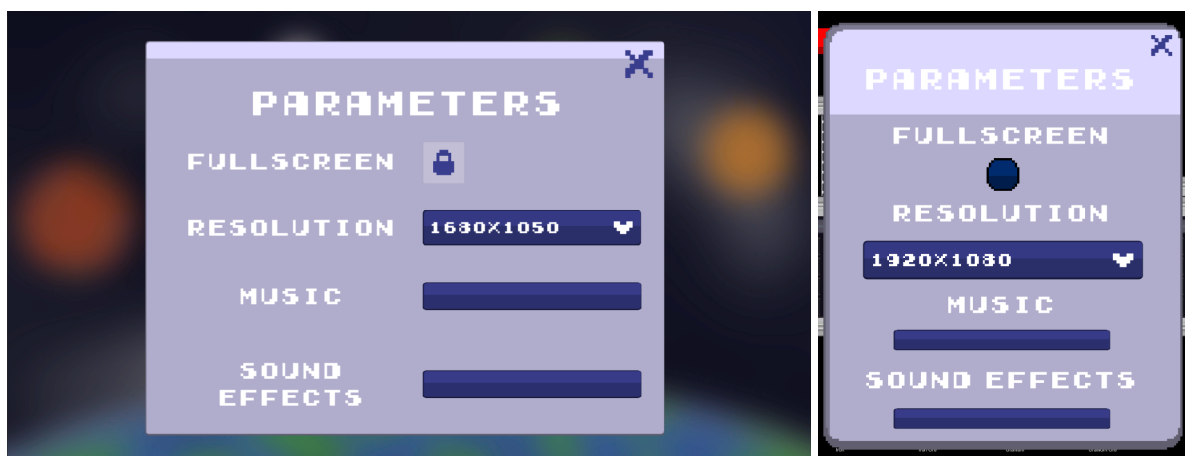
Il est également possible de choisir la résolution d'écran voulue. Cependant au départ nous avons rencontré un problème dans notre code car lorsque l'on lançait le jeu, les résolutions proposées étaient toutes en double. Pour corriger cela, nous avons cherché sur Internet et avons trouvé une ligne de code permettant de remédier à cet ennui :

```
resolutions = Screen.resolutions.Select(resolution => new Resolution { width = resolution.width, height = resolution.height }).Distinct().ToArray();
```

Cela permet de récupérer toutes les résolutions disponibles grâce au "Select". Ensuite, tous les éléments dupliqués sont supprimés grâce à l'expression "Distinct()".

Et enfin, les résolutions vont être stockées à l'aide de l'expression "ToArray()" dans notre tableau de résolutions "resolutions".

Voici ci-dessous à gauche le rendu depuis le menu principal ainsi qu'à droite le rendu en jeu :



- QUIT : ce bouton permet tout simplement à la personne de fermer l'application.

Pour rendre ce menu principal le plus agréable possible pour les joueurs, nous avons choisi d'utiliser la classe "PlayerPrefs". Cette classe va être utile dans la fenêtre où le joueur va effectuer ses réglages. Désormais les choix du joueur pour la résolution d'écran ainsi que la musique et les effets sonores seront toujours conservés entre deux scènes, ou alors à chaque fois qu'il lancera le jeu.

2.6 Multijoueur

Le multijoueur n'a pour l'instant pas été développé même si nous avons réfléchi à la façon dont nous aimerions qu'il soit fait. Nous comptons faire en sorte que deux joueurs puissent jouer ensemble mais dans un même vaisseau avec des paramètres qui seront modifiés en termes de difficulté. Malgré ce ne sont que de brèves idées, nous ne sommes pas encore sûr que l'implémentation du multijoueur se fera réellement.

2.7 Sons / effets sonores

Comme nous l'avons stipulé dans le précédent rapport de soutenance, l'entièreté des sons et effets sonores sont des sons libres de droit que l'on a pris sur Internet, notamment à l'adresse suivante : <https://soundimage.org/>. Tout d'abord, nous avons choisi d'utiliser un son ambiant de vaisseau spatial pour la musique de jeu. Étant donné qu'aucun de nous dans notre groupe n'a jamais réellement eu d'expérience dans le domaine de la musique, la boucle possède une légère coupure à chaque redémarrage du son. Nous essayerons donc d'améliorer ce problème non résolu d'ici la prochaine soutenance. Concernant les effets sonores, pour cette soutenance nous ne nous sommes pas réellement concentrés sur le fait de les implémenter mais plutôt sur le fait d'en trouver. En effet, cela n'a pas été simple de trouver plein d'effets sonores gratuits et libres de droits et correspondant à nos critères. Malgré cela, nous avons pu trouver différents effets sonores. Actuellement nous en avons pour les combats, les déplacements de joueur, les montres... Ainsi, pour le moment seuls les sons de déplacements du joueur sont implémentés.

À propos des futurs sons à mettre en jeu, il sera important de bien le faire car ils jouent un rôle très important dans l'immersion des joueurs.

Maintenant, concernant Unity, pour gérer tous ces sons nous utilisons un mixeur audio contenant un groupe avec l'ensemble des sons et effets sonores et à ce groupe y est rattaché deux groupes enfants ; un pour les sons et un pour les effets sonores. En effet, il était important de créer deux groupes enfants pour différencier les musiques des effets sonores car grâce à cela le joueur a désormais plus de choix pour les réglages du volume et cela est plus agréable. On peut retrouver ces réglages là dans le menu principal ainsi que dans le jeu (cf. captures d'écran de la partie **2.5 Menu principal**).

L'implémentation du son pour les déplacements a été simple à ajouter une fois la bonne technique trouvée. Pour cela, nous détectons si la touche du clavier flèche gauche ou flèche droite est utilisée est que le joueur est sur un sol avec le méthode "isGrounded".

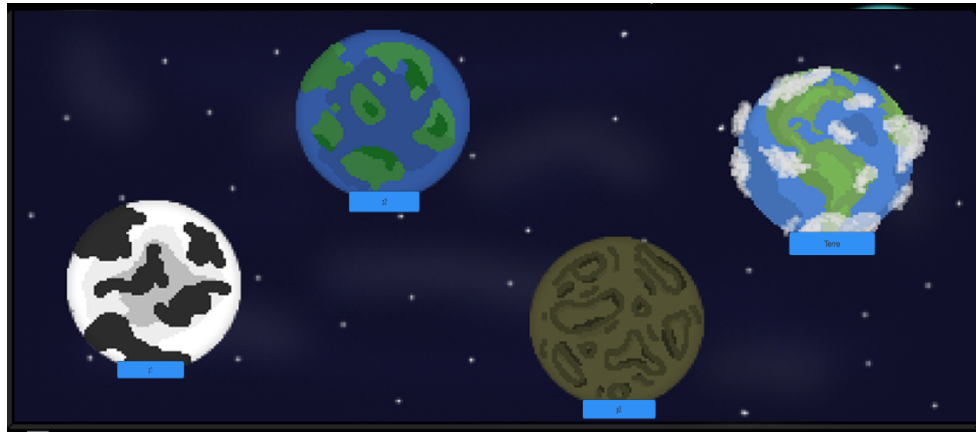
Également, nous avons utilisé un algorithme intéressant trouvé sur un forum pour ajuster plus correctement le volume du jeu.

```
audioMixer.SetFloat("Music", Mathf.Log10(volume) * 20);
```

Le fait d'utiliser la fonction log et de régler la barre de volume entre 0.0001 dB et 1 dB nous permet d'ajuster de manière plus précise le son. En effet, lorsque le joueur descendra la barre vers 0.0001 le son baissera beaucoup plus vite jusqu'à s'éteindre tandis que lorsque la barre s'approchera de 1 le son ne variera que peu.

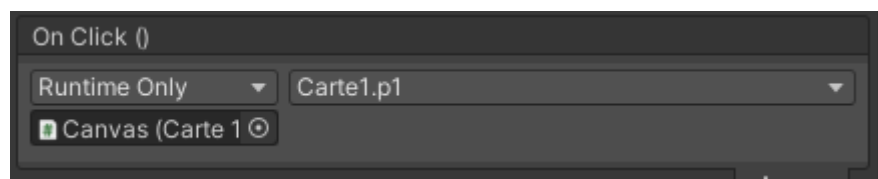
2.8 Interface carte

Dans le précédent rapport un poste de pilotage avait déjà été implémenté permettant lorsque nous sommes dans un certain rayon autour de celui-ci ainsi qu'en appuyant sur la touche M de pouvoir l'ouvrir. À ce moment-là, cette dernière n'était qu'un écran avec étoiles sans utilité. Mais désormais cette carte a enfin l'utilité prévue, lorsque l'on ouvre la carte, nous pouvons observer quatre planètes à l'écran ainsi qu'un bouton en dessous de chacune d'elle comme vous pouvez le voir ci-dessous.



Chacun de ces boutons permettent au joueur lorsque ce dernier en aura rempli les conditions de pouvoir passer au niveau suivant, conditions dont nous expliquerons les détails dans la partie **2.10** (Missions et histoire). Pour gérer ces 4 boutons nous avons utilisé un seul script avec à chaque une méthode reliée à chacun de ses boutons et qui dès que le bouton est activé par le joueur s'exécute.

Chaque bouton est ce que l'on appelle un GameObject sur unity, ces derniers ont donc besoin d'être reliés de la façon suivante.



Le Canvas(Carte 1) représente le code auquel on veut lier notre bouton et le Carte1.p1 représente la méthode "p1" que l'on veut exécuter lorsque l'on appuie sur le bouton en question. Pour finir lorsque le bouton est enclenché , nous activons l'apparition de 2 monstres qui à chaque nouvel appuie deviennent de plus de plus en plus fort.

Ci dessous vous trouverez un exemple de code exécuté lorsque l'on appuie sur le bouton tout à gauche de l'image présentée au-dessus.

```
public void p1()  
{  
    if (cp1)  
    {  
        monstre1.SetActive(true);  
        monstre2.SetActive(true);  
        cp1 = false;  
    }  
    else  
    {  
    }  
}
```

Ici, lorsque la condition liée au mission est vérifiée grâce à la variable booléenne “cp1”, nous activons la mise en marche des 2 monstres en utilisant la commande “GameObject.SetActive(true);” qui permet d’activer le GameObject en question.

Comme vous pouvez le remarquer, lorsque l’on rentre dans le “if” nous remettons de suite cp1 à l’état “false” pour que le joueur ne puisse relancer plusieurs fois le niveau et ainsi pouvoir profiter des divers avantages qu’offre la validation d’un niveau de façon illimitée.

Pour finir il y a également un “else” qui lui est vide pour le moment est qui à terme quand toutes les missions seront implémentées aura pour instruction si les conditions ne sont pas validées, d’envoyer un message au joueur lui disant les missions qu’il n’a pas achevées.

2.9 Cinématiques

Les cinématiques n’ont pas été une priorité pour cette soutenance. Nous avions pourtant prévu de commencer cette tâche malheureusement nous avons jugé plus raisonnable de finir correctement les autres tâches à réaliser.

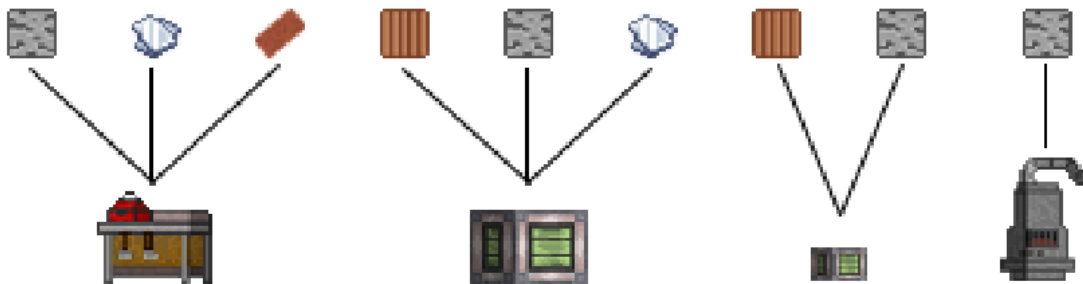
2.10 Missions et histoires

Tout au long du jeu, différentes quêtes seront disponibles. La quête principale du joueur sera de rentrer sur la planète Terre après s’être perdu dans l’espace, pour cela il remplira diverses tâches qui lui permettront de parvenir à son but. Les missions n’ont pas été implémentées sur Unity mais nous avons créé une liste des matériaux obtenables à partir de celles-ci :



pierre	bois	minerai de fer	fer	minerai de cuivre	cuivre	minerai d'uranium	uranium
--------	------	-------------------	-----	----------------------	--------	----------------------	---------

Nous avons également réfléchi à leur utilisation en fonction des objets que nous avons créé et voici ce que cela donnerait à l'aide d'un schéma :



Cela représente quelles ressources seront nécessaires afin de créer ces objets, cependant leur quantité n'a pas encore été décidée.

2.11 Gestion / Design planètes

Désormais, nous possédons trois fonds étoilés différents. En effet, nous avons réalisé un nouveau fond affiché en jeu pour cette soutenance, en voici un aperçu ci-dessous :



Le processus de création reste le même que pour les précédents : nous créons en arrière-plan, différentes planètes avec des étoiles puis une grande planète en bas d'écran (différente de

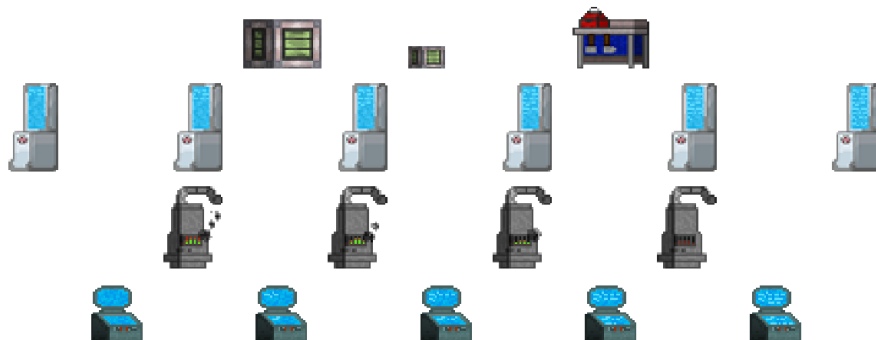
celles des deux autres fonds) qui indique la planète actuelle sur laquelle le joueur se situe. De plus, ce décor là a été réalisé à partir de Photoshop comme les deux précédents.

2.12 Machines / inventions

Maintenant, intéressons-nous à l'évolution de nos machines et inventions. Nous possédons désormais plusieurs dessins représentant au total six objets utilisables différents. En effet pour le moment nous avons :

- deux boîtes de tailles différentes permettant de stocker les objets du joueur
- un établi pour fabriquer des objets pour progresser dans l'aventure
- un téléporteur envoyant le joueur à l'étage supérieur lorsqu'il marche dessus
- un ordinateur de bord pour changer de planète (il y a plusieurs tuiles pour pouvoir faire bouger du texte à l'écran et améliorer l'expérience de jeu des joueurs)
- un fourneau qui permet aux joueurs de faire fondre les minerais qu'ils obtiennent, de plus cet objet est également animé; de la fumée en sort lorsqu'un minerai est en train de fondre
- un poste de commande qui servira aux joueurs de consulter les missions actuelles et futures à accomplir, de plus l'animation de l'écran est similaire à celle de l'ordinateur de bord

Voici une capture d'écran regroupant les inventions dessinées :

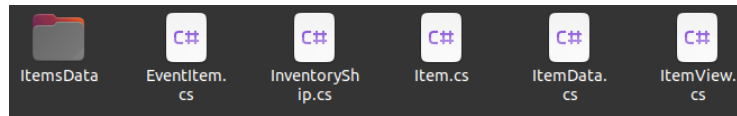


Il est important de noter que de futures machines seront créées d'ici à la soutenance finale. De plus, les joueurs auront la possibilité de choisir où placer certaines de ces machines. À présent, passons à la partie technique.

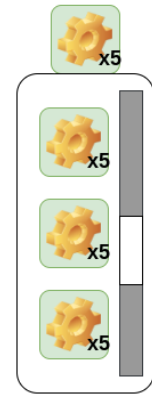
Lors de la dernière soutenance les items étaient mal implémentés, la partie construction de la vue n'était pas séparée de celle qui stock le type d'item et son nombre, ce qui rendait impossible la création de plusieurs vues pour un même item.

Les items sont à la base des machines, la première étape a été de séparer en trois la classe item, la vue qui est enregistré dans une liste de l'item pour pouvoir mettre à jour celle-ci

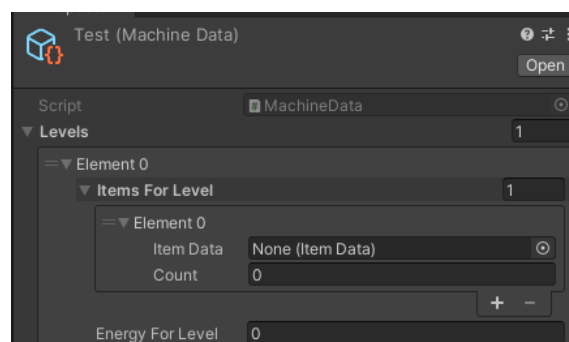
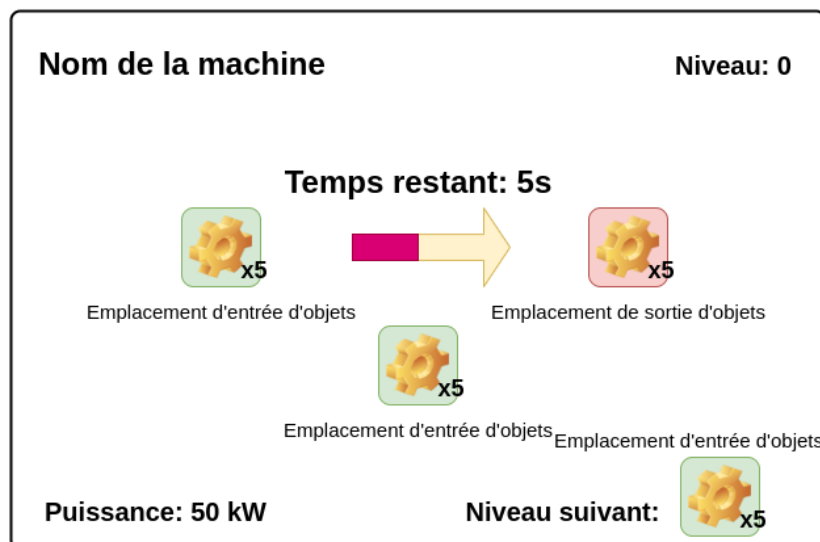
quand la quantité change, l'item qui ne fait que stocker le nombre et le type d'item et enfin l'inventaire du vaisseau qui stocke une liste d'item.



La seconde étape a été de créer la classe “Slot” qui permet d’afficher un item et d’interagir lorsqu’on clique dessus dans le panel des machines. Il y en a deux types, un pour l’entrée des items quand le joueur cliquera sur la case vide un menu déroulant avec la liste des items qu’il est possible de mettre dans la machine apparaîtra. Il pourra cliquer sur celui de son choix et par la suite le modifier s’il le veut. Le second type est la sortie, quand la machine aura terminé de faire son traitement l’item final apparaîtra dans la case et le joueur pourra cliquer dessus pour l’obtenir.

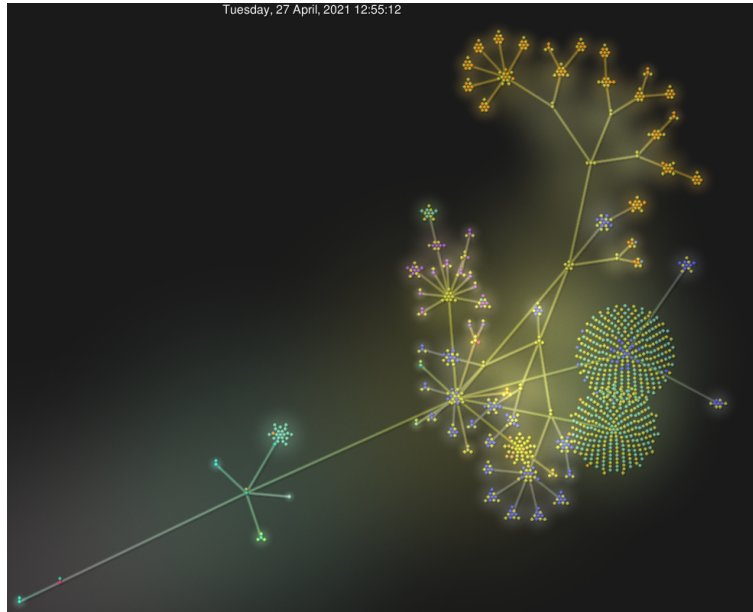


La troisième étape est la création de la classe “Machine” qui permet de regrouper la logique commune à la plupart des machines. Il est possible de créer des sous-classes pour changer ou désactiver certaines logiques. Un point important est la création d’un “ScriptableObject” pour rendre la création des machines plus rapide en séparant les constantes de la logique. Ainsi, tout ce qui concerne la vitesse, les puissances, les recettes en fonction du niveau de la machine et ce qu’il faut pour passer au niveau supérieur est décrit dans ce “ScriptableObject”. Malgré qu’il soit complexe, il est rapide à remplir via l’interface d’Unity.



3 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.



4 Expériences personnelles

Dans cette partie, nous délivrerons chacun nos ressentis personnels à propos de notre travail fourni.

Antoine Adam : À la première soutenance j'avais fait principalement des implémentations en C# sans rencontrer trop de problèmes. C'est à partir de la seconde soutenance que j'ai vu beaucoup de bugs arriver, j'ai perdu du temps à les corriger et il a même fallu des fois recréer certaines parties du code. Cependant au final le code a gagné en stabilité. Je comprends qu'il nous est imposé de faire des prévisions, mais je trouve que c'est très compliqué de les respecter si on veut faire un code propre. J'ai du mal à faire autrement, quand je fais quelque chose trop rapidement le code n'est pas adapté à de futures modifications, ça crée par la suite des bugs et l'obligation de refaire la partie qui pose problème. C'est donc ce qui m'a permis de prendre conscience des problématiques liées aux délais dans les projets.

Alexandre Dugot : Je dois dire qu'après la première soutenance, j'avais bien du mal à trouver mes marques sur la plateforme unity étant donné que je ne l'avais jamais utilisé pour des projets de cette envergure. Mais finalement, après être resté très accroché à des vidéos tutoriels ou encore des docs unity pour me permettre de coder efficacement je suis content car désormais j'ai réussi à prendre mes marques avec Unity mais aussi le C#. Aujourd'hui cela me permet donc de coder des choses bien plus intéressantes sans avoir à passer des heures sur internet à me demander comment cela peut se faire. Pour finir, lors de la précédente soutenance je n'avais guère eu des problèmes de bug sur mes programmes mais augmentant la difficulté de ces derniers je me suis rendu compte que les bugs deviennent de plus en plus

fréquents dans un programme prenant de l'ampleur et que par conséquent ceci occupe une grande partie du temps que l'on passe à créer le jeu.

Mohamed Ghaffour : Je dois avouer que lors de la préparation de cette deuxième soutenance j'ai rencontré des problèmes dont j'ignorais la connaissance. En effet j'ai compris qu'il était parfois préférable de se détacher de son idée de départ et de lire les documentations afin de progresser. Ce projet est pour moi une réelle opportunité de progresser, en effet je sens une réelle différence sur la compréhension et la manipulation de l'orienté objet en C#. J'ai également appris à manipuler l'HTML et le CSS, ce qui a été pour moi une grande découverte.

Au-delà de la programmation, ce projet m'a énormément aidé sur l'aspect communication et esprit d'équipe. En effet le groupe a été vraiment soudé et je pense que c'est l'une des règles d'or lors d'un travail de groupe.

Nathan Labernardiere : Durant la première soutenance, je n'avais pas vraiment eu l'occasion de travailler sur Unity. Cependant, pendant cette deuxième période j'ai pu approfondir et améliorer mes connaissances en programmation avec Unity. J'ai ainsi rencontré de nouveaux problèmes que je ne connaissais pas avant étant donné que les scripts sur lesquels j'ai pu travailler changent beaucoup de ce que nous faisons en TP de programmation. Malgré cela, j'ai quand même pu continuer à travailler sur les fonctionnalités qui me tenaient à cœur lors de la soutenance précédente. En effet, les tâches relatives aux graphismes de notre jeu suscitent mon intérêt et me passionnent donc. Ainsi, ce projet représente selon moi un très bon exercice afin de m'entraîner et d'apprendre comment réaliser un projet de grande envergure.

5 Avances et Retards

Voici un rappel des prévisions d'avancement pour les différentes soutenances :

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%

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

- La gestion des cinématiques
- La gestion des missions
- La gestion du multijoueur

Nous pensions à la base implémenter le début du multijoueur de notre projet avant la deuxième 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 avons réalisé les missions en pseudo-code. C'est-à-dire que toutes les conditions sont là, mais qu'il manque encore l'implémentation de ces dernières due à des problèmes. Notamment le fait que nous avons un

bug qui fait que la machine gérant les missions ne s'affiche pas ou encore que pour une raison inconnue nous n'arrivons pas à obtenir le nombre d'item dans le code de la machine. Et pour finir, les cinématiques n'étant qu'un plus dans le jeu, nous avons pensé que ces dernières devraient plutôt être implémentées à la fin lorsque le reste sera complètement opérationnel.

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

- Personnages
- Structure du vaisseau
- Interface carte

Voici maintenant les avancements à l'image de ce qu'ils sont actuellement :

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

6 Prévisions pour la suite

Dans cette partie nous allons vous présenter cette fois-ci ce que nous avons prévu pour la toute dernière soutenance. Bien-sûr ce ne sont que des prévisions, car nous ne savons pas encore quelles difficultés nous attendent par la suite.

Ci-dessous, nous allons vous présenter les prévisions pour chaque partie de notre projet en le découpant par tâche.

1) Personnage :

La partie liée aux personnages arrive quasiment à son terme, mais il manque encore des fonctionnalités importantes ainsi que certains détails pour rendre le jeu plus agréable pour celui qui y joue. Premièrement, les monstres peuvent déjà attaquer le joueur en lui faisant baisser sa barre de vie, mais le joueur lui ne peut guère exécuter cette tâche, cela sera donc l'objectif pour la prochaine fois. Pour en finir avec la partie attaque des personnages, nous allons également faire en sorte que le joueur puisse gagner de nouvelles armes au fur et à mesure de la partie pour que ce dernier puisse rivaliser avec des monstres de plus en plus forts.

Il faudra également empêcher le personnage de se déplacer lorsqu'un menu est ouvert. Pour finir avec cette première partie, nous comptons également ajouter des animations pour rendre le jeu plus joli, mais aussi améliorer les IA qui sont nos monstres pour qu'elles puissent attaquer le joueur de façon plus efficace, notamment dans les derniers niveaux pour rendre le jeu plus compétitif.

2) Sauvegarde :

Le menu de sauvegarde et leur gestion sont déjà terminés, mais il faut signaler qu'à chaque ajout de fonctionnalité par la suite, il faudra modifier le système de sauvegarde en sauvegardant les nouveaux attributs.

3) Gestion du vaisseau et design :

Le travail restant à faire dans cette partie est majoritairement relatif aux ajouts d'objets à l'intérieur du vaisseau. Nous ne voulons pas que le vaisseau paraisse vide et que le joueur se lasse vite de notre jeu. C'est pourquoi nous avons jusque-là ajouté plusieurs objets avec de petites animations et notre prochain objectif sera de continuer à faire cela. Nous prévoyons donc d'ajouter d'autres petits objets de décorations ou bien des objets avec une animation.

4) Site internet :

La prochaine soutenance aura surtout pour but de mettre les règles du jeu et un wiki sur le site pour qu'un joueur débutant puisse apprendre à jouer. Lorsque le jeu sera terminé il faudra implémenter la toute dernière chose qui est la mise en ligne du jeu pour que l'on puisse le télécharger.

5) Menu principal :

Le menu principal nous semble plutôt complet pour l'instant que ce soit sur l'aspect graphique ou bien par les options proposées. Cependant, certains aspects graphiques dans le menu peuvent être améliorés ainsi que proposer de nouvelles options telles que le choix de la langue pourrait être envisageable ou alors ajouter une option crédits notamment pour les sons sera nécessaire.

6) Multijoueur :

Comme précisé dans le précédent rapport, le multijoueur pour le moment n'a pour objectif d'être développé. En effet, nous préférons tout d'abord obtenir un jeu qui fonctionne avec un seul joueur avant de nous consacrer à cette partie. Nous ne sommes donc pas sûr que le multijoueur sera réellement implémenté à la fin du projet.

7) Son/Effets sonores :

Comme nous l'avons dit dans ce rapport, une partie des sons n'a pas encore été implémentée. Ainsi, les objectifs d'ici à la prochaine soutenance sont d'améliorer la qualité de certain dû à la manière dont ils sont joués en jeu et de finir d'implémenter tous les effets sonores restants. L'objectif final étant de rendre l'expérience la plus agréable et que les joueurs prennent du plaisir en jouant avec du son.

8) Interface carte :

Pour cette partie il ne reste que peu de choses à faire, mais il reste tout de même à ajouter les conditions liées aux missions à ajouter pour que le joueur ne puisse activer les nouveaux niveaux que lorsque c'est voulu.

Il faudra également faire en sorte que le joueur obtienne des récompenses lorsque le joueur passe un nouveau niveau.

Pour finir, il n'y a pour le moment que peu de niveaux disponibles alors nous pensons certainement en ajouter pour rendre le jeu plus long et plus riche dans son ensemble.

9) Cinématiques :

Dans cette partie, l'ensemble des cinématiques seront faites lors de la dernière partie du projet. Ces dernières consisteront en un ensemble d'images , de textes et de sons qui donneront au joueur les bases du scénario de notre jeu.

10) Missions et histoire :

Pour les missions, il n'y a pas grand-chose à dire à part que nous aimerions juste en faire un nombre plus conséquent, mais aussi des missions plus difficiles qui puissent tenir en haleine le joueur le plus longtemps possible. ceci sera possible lorsque les machines et inventions seront finalisées.

11) Gestion/Design planètes :

À l'heure actuelle, les joueurs peuvent choisir entre trois types de planètes différentes. Cela est déjà satisfaisant, mais nous aimerions pouvoir proposer encore une voire deux planètes différentes aux joueurs. Pour cela il faudra dessiner deux nouveaux fonds affichés en jeu et les implémenter dans Unity.

12) Machines / Inventions :

Pour ce qui est des machines, nous allons finir d'implémenter la classe mère des machines en leur ajoutant certains attributs comme le fait d'avoir une barre d'énergie par exemple. Certaines machines sont déjà implémentées, mais nous comptons en ajouter d'autres qui seront sur le même principe que les précédentes, mais qui ajoutent de nouvelles fonctionnalités intéressantes.

7 Conclusion

Premièrement, nous sommes heureux de nous dire que nous sommes passés d'un jeu encore très superficiel à un jeu quasiment jouable sur lequel le joueur peut déjà effectuer maintes actions. La dernière fois, nous avions des connaissances que nous pouvions considérer comme basiques sur la plateforme Unity. Néanmoins, désormais nous arrivons à développer bien plus aisément notre jeu en y implémentant des choses bien plus poussées et même plus optimisées en termes de performances.

Encore une fois, nous avons bien appris depuis la précédente soutenance. Cela nous sera certainement utile pour la suite, que ce soit pour des projets, des TP que nous allons réaliser dans le cadre de nos cours à EPITA, à l'étranger ou encore pour des projets personnels que nous pouvons avoir envie de réaliser.

Pour finir, comme dit précédemment le jeu commence à devenir jouable, mais il manque encore bien des détails à implémenter, mais aussi des liaisons à faire entre les diverses parties du projet. Malgré cela, nous sommes toujours très optimistes et avons hâte de voir l'aboutissement de notre jeu.