

# **Projet Engi'Blocks**

Rapport de deuxième soutenance

Groupe Engistar

Baptiste Schaudel  
Anouar Benouda

9 décembre 2021

# Table des matières

## 1 Introduction

## 2 Avancements individuels sur les fonctionnalités

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

## 3 Avances et Retards

## 4 Prévisions pour la suite

## 5 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 (fini) rend le tout plus présentable. Néanmoins, il manque encore un certain nombre de finitions pour que notre jeu paraisse plus qualitatif.

## **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 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.

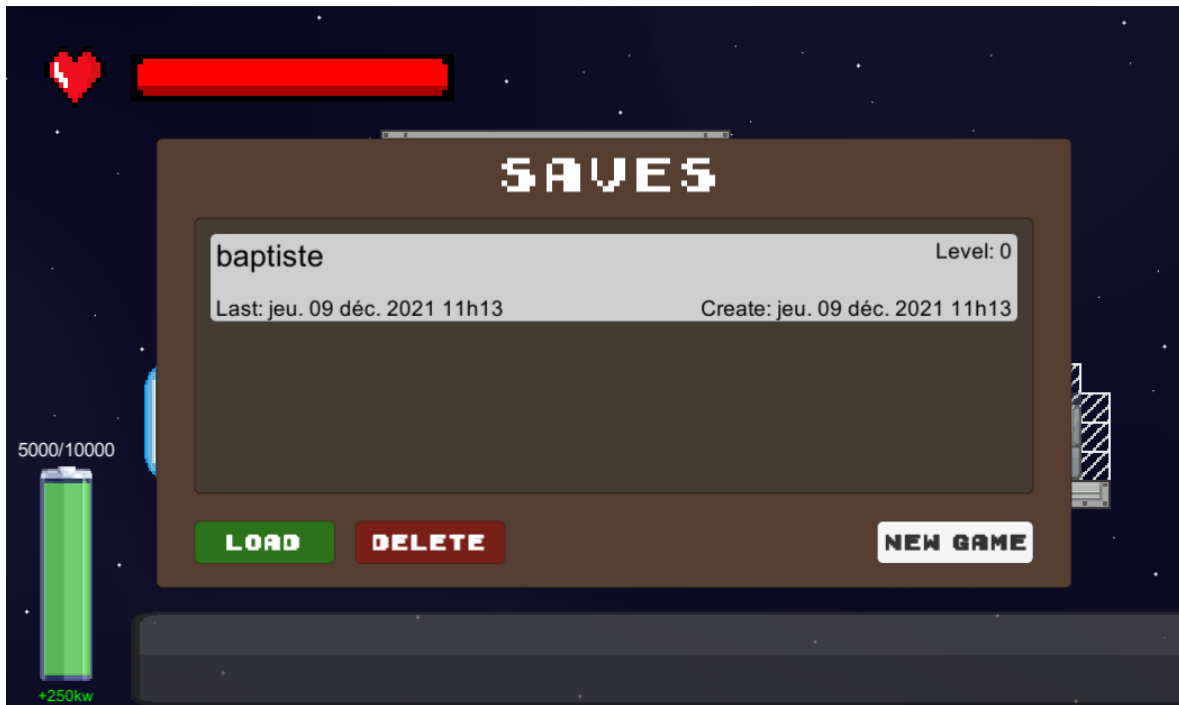
En début de chaque aura des capacités de base (points de vie, attaque) mais ces dernières augmentent au fur et à mesure de la partie pour en augmenter la difficulté.

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.

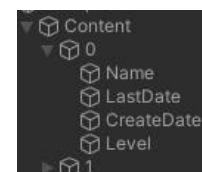
Et enfin la seconde est quant à elle liée à l'animation dont la fluidité est compliquée à gérer, d'ailleurs nous tenons à signaler qu'en l'état actuel des choses , les animations ne sont pas encore excellentes.

## 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 Menu principal

La dernière fois on vous avait présenté le menu principal qui était optionnel que pour quelques options (play et quit) cependant pour cette soutenance on l'a rendu opérationnel pour le bouton option.

- 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.

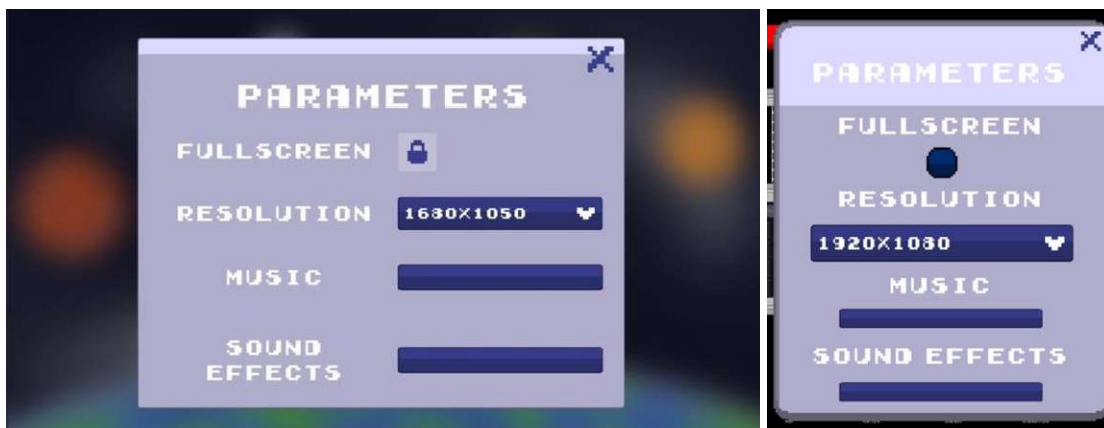
Il est également possible de choisir la résolution d'écran voulue. Cependant au départ nous avons rencontré un problème dans notre code car lorsque l'on lançait le jeu, les résolutions proposées étaient toutes en double. Pour corriger cela, nous avons cherché sur Internet et avons trouvé une ligne de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



---

## **2.4 Multijoueur**

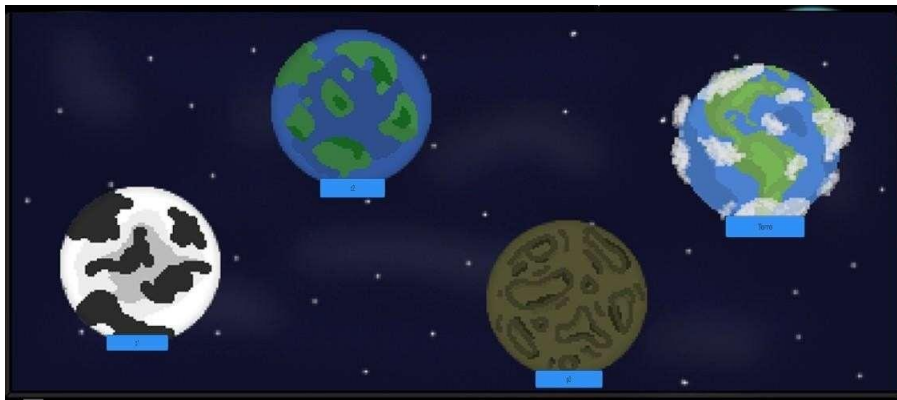
Le multijoueur n'a pour l'instant pas été développé même si nous avons réfléchi à la façon dont nous aimerions qu'il soit fait. Nous comptons faire en sorte que deux joueurs puissent jouer ensemble mais dans un même vaisseau avec des paramètres qui seront modifiés en termes de difficulté. Malgré ce ne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

## **2.5 Sons / effets sonores**

Comme nous l'avions 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/>. Donc nous avons terminé les effets sonores puisqu'ils sont opérationnels

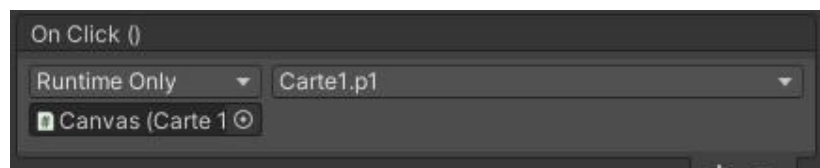
## 2.6 Interface carte

Dans le précédent rapport un poste de pilotage avait déjà été implémenté. À ce moment- là, cette dernière n'était qu'un écran avec étoiles sans utilité. Mais désormais une carte est opérationnel en appuyant sur la touche M, 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 toute 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 achevé.

## 2.7 Cinématiques

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

## 2.8 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 implémenté sur Unity mais nous avons créé une liste des matériaux obtenables à partir de celles-ci :

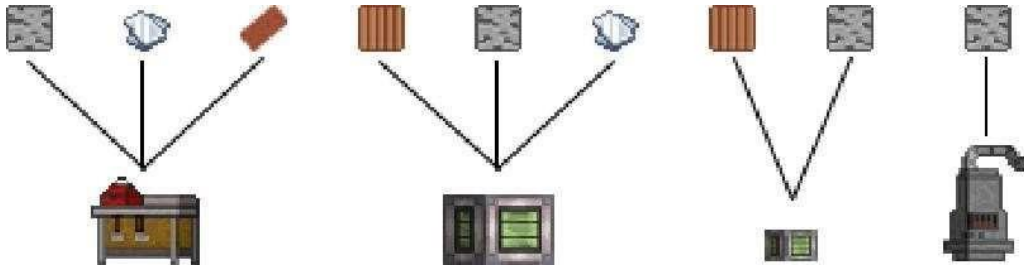


---

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.

## 2.9 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.

## 2.10 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 :

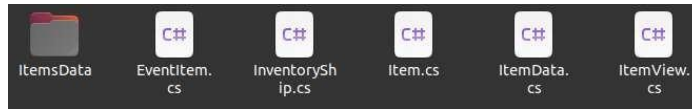


Il est important de noter que de futures machines seront créées d'ici à la soutenance finale. De plus, les joueurs auront la possibilité de choisir où placer certaines de ces machines. À présent, passons à la partie technique.

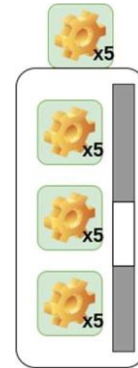
Lors de la dernière soutenance les items étaient mal implémentés, la partie construction de la vue n'était pas séparée de celle qui stock le type d'item et son nombre, ce qui rendait impossible la création de plusieurs vues pour un même item.

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.

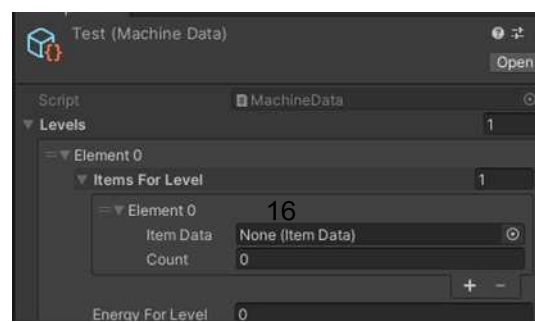
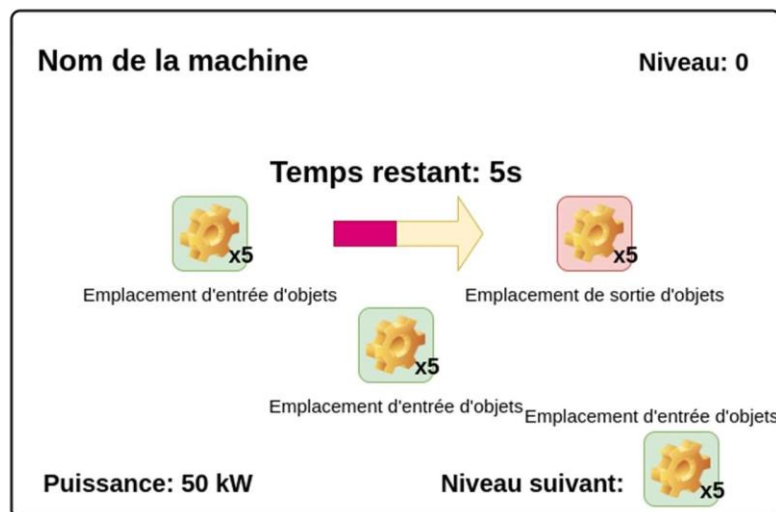


La seconde étape a été de créer la classe "Slot" qui permet d'afficher un item et d'interagir lorsqu'on clique dessus dans le panel des machines. Il y en a deux types, un pour l'entrée des items quand le joueur cliquera sur la case vide un menu déroulant avec la liste des items qu'il est possible de 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 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	0%	0%	100%
Effets sonores	0%	100%	100%
Cinématiques	0%	50%	100%
Interface carte	40%	75%	100%
Multijoueur	0%	0%	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 du site internet
- 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
- Effet sonore

---

## 4 Prévisions pour la suite

Dans cette partie nous allons vous présenter cette fois-ci ce que nous avons prévu pour la toute dernière soutenance.

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.

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 pour but de réaliser le site internet.

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 :

La partie effet sonore a été terminée.

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 et de textes 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.

## **5 Conclusion**

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. 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.