

# RAPPORT DE PROJET IHM



CAPILLIEZ Maxance  
SEDDAR Naïm  
PINHAL Nicolas  
DUHEM Alexandre  
WARLOUZET Cloé

Année 2017-2018

---



---

## TABLES DES MATIÈRES

<b>Introduction</b> .....	Page 2
Règles du jeu.....	Page 3
Outils informatiques.....	Page 4
Diagramme de Gantt.....	Page 5
Diagramme de classe.....	Page 6
Diagramme des Cas Utilisations .....	Page 7
L'arborescence.....	Page 8
Charte graphique.....	Page 9
Code et démonstration.....	Page 10
<b>Bilan</b> .....	Page 13



---

## INTRODUCTION

Après des heures d'acharnement, de travail et de réflexion, nous avons le plaisir de vous présenter notre projet: le [Space 2048](#) !

### De quoi s'agit-il?

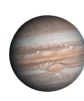
Il s'agit d'un jeu disponible sur ordinateur. Le Space 2048 est une version revisitée du 2048 original que l'on peut observer ci-dessous.

2	2	8	4
		2	32
4	8	32	64
1024	512	256	128

Comme demandé, nous avons changé le gameplay et le graphisme selon notre imagination, et le moins que l'on puisse dire, c'est que nous en avons beaucoup.

Notre jeu s'inspire du 2048 mais aussi du Space Invaders afin d'inventer un jeu totalement nouveau. Laissez-nous vous expliquer les règles du jeu.

---



---

## Les règles du jeu

Les règles du jeu sont nouvelles, mais restent cependant relativement simples et à la portée de tous.

Il s'agit là de réussir, grâce aux touches flèches, à bouger la fusée sur le plan horizontal de façon à intercepter une planète ayant le même chiffre que celui de la fusée. Ainsi, selon les règles du 2048, votre fusée doublera son chiffre. Le but étant de doubler son score jusqu'à atteindre 2048.

Cependant, si cette dernière était la seule règle, le jeu serait beaucoup trop simple!

Voici deux contraintes auxquelles vous ne pourrez échapper:

- Si vous touchez une planète ayant un chiffre **plus petit** que la fusée, celle-ci prendra le chiffre de la planète.

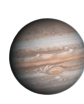
Par exemple, si votre fusée est à 128 et que vous touchez un 8, votre fusée prendra la valeur 8.

- Si vous touchez une planète ayant un chiffre **plus grand** que la fusée, celle-ci sera divisée par deux.

Par exemple, si votre fusée est à 4 et que vous touchez un 8, votre fusée prendra la valeur 2.

En revanche, si vous êtes à 2 et que vous touchez autre chose qu'un 2, vous aurez perdu.

Le Space 2048 comporte **3 niveaux** de difficulté étant différenciés par la rapidité de mouvement des planètes.



---

## Les outils informatiques

Nous avons choisi de développer notre jeu avec le langage de programmation **Python**. Plus particulièrement nous avons utilisé la librairie intitulée **pygame**.

Qu'est-ce que pygame ?



Pygame est créé en 2000 par **Pete Shinnars** lorsqu'il découvre Python et SDL.

SDL ou Simple Directmedia Library est une bibliothèque libre multi-plateformes permettant la gestion du multimédia dans la programmation. La SDL est écrite en C, mais est utilisable avec un grand nombre de langages, comme le C/C++, Pascal, ou encore ... Python !

Elle sert notamment à gérer :

- l'affichage vidéo 2D
- la gestion de l'audio
- la gestion des périphériques de commandes (clavier, souris...)

Pygame est la fusion de Python et SDL avec quelques ajouts ce qui est parfait pour la création de jeu vidéo en 2D.

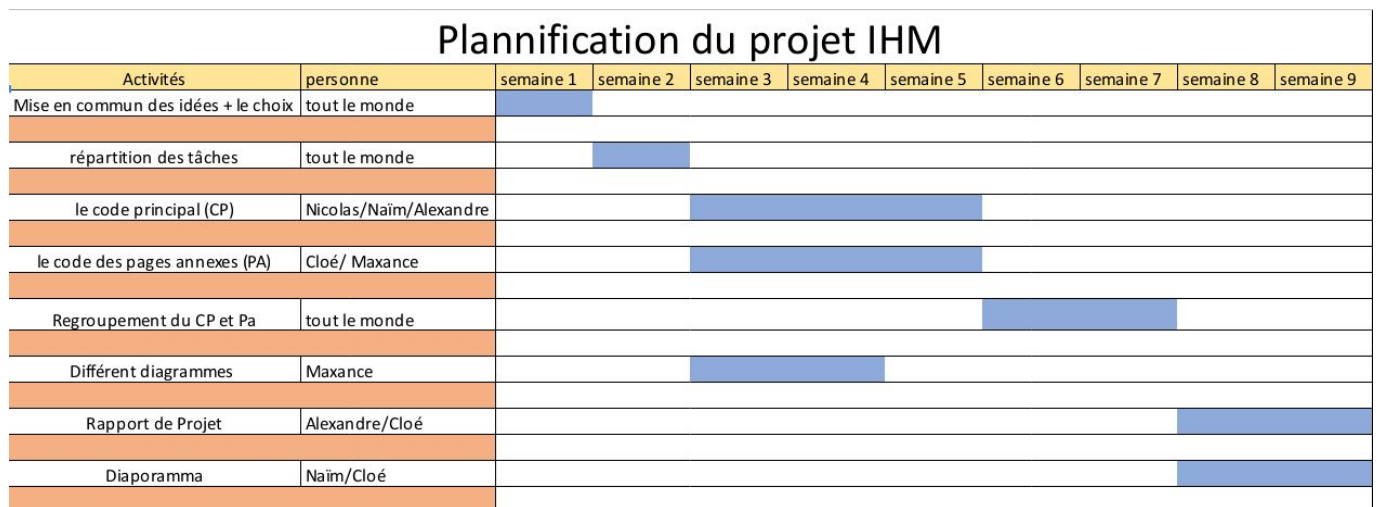
Voici une liste d'avantages de Pygame pour le développement de jeu vidéo :

- Compatible avec Python 3
- Portable sur différents systèmes d'exploitation
- Ajouts de fonction spécifique aux jeux-vidéos
- Gestion de fonctionnalités propre au JVs comme les sprites

Pour la gestion du projet nous avons choisi Github qui est un outil collaboratif. Il possède un système de partage de fichier qui s'actualise lors de chaque dépôt. De plus il possède une interface console et graphique.



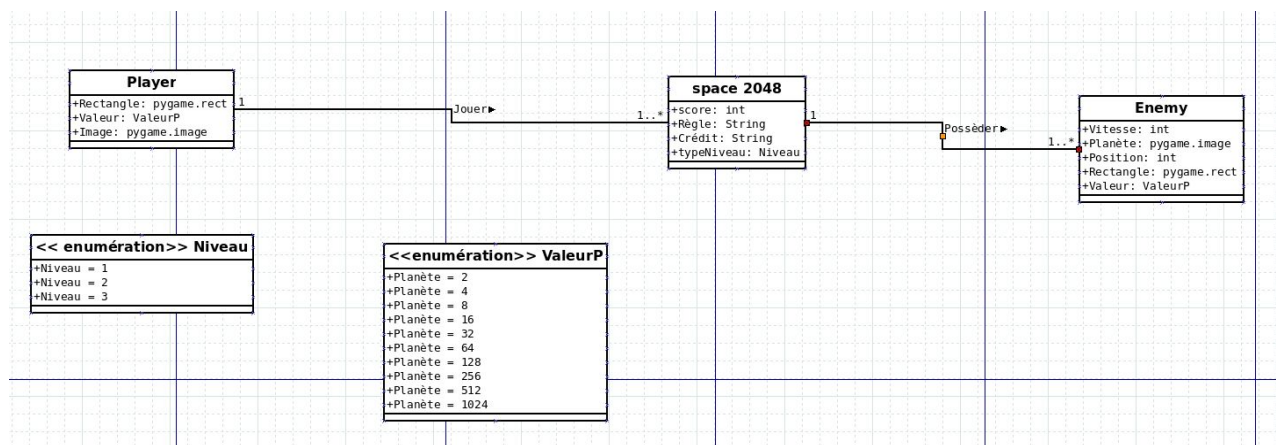
## Diagramme de Gantt



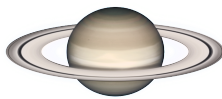
Ce diagramme de Gantt représente notre planning, les différentes tâches que l'on a réalisées durant ce projet. On peut voir que le travail a bien été réparti dans l'ensemble du groupe. Chacun s'est appliqué dans une tâche qu'il aimait et avec laquelle il était à l'aise. Ainsi, tout le monde travaille dans la joie et la bonne humeur et donc par conséquent, être plus efficace.



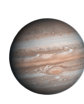
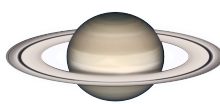
## Diagramme de Classe



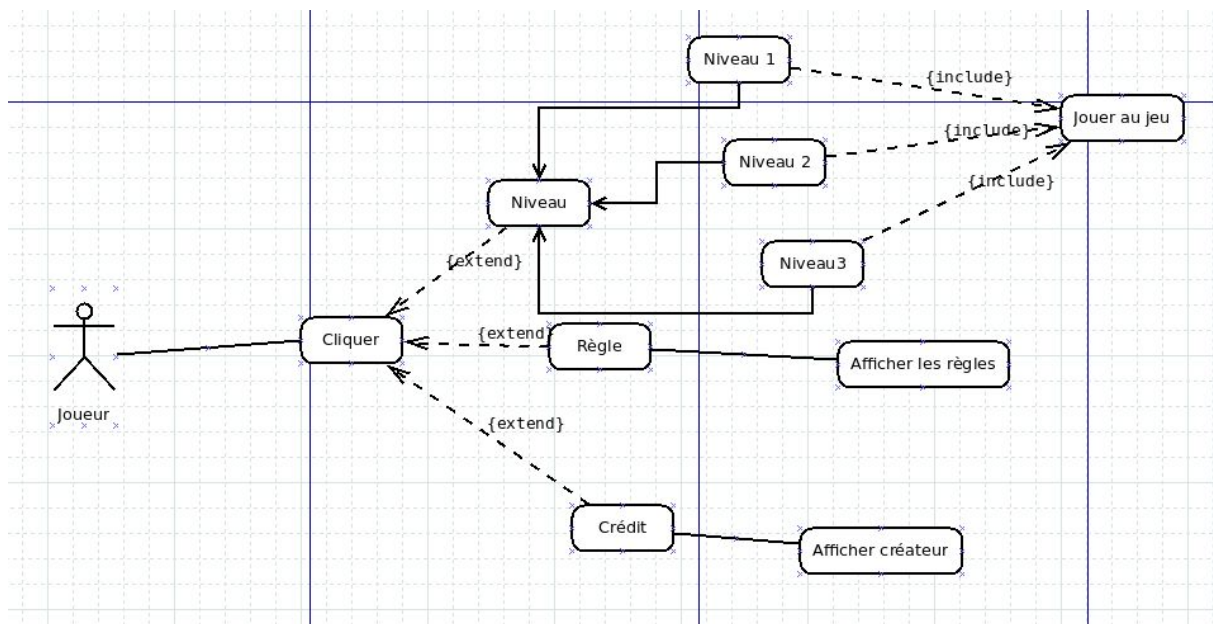
Ce diagramme de classe représente le jeu. Nous avons 2 classes, une classe joueur et une classe space 2048. Un joueur peut jouer 1 ou plusieurs fois au space 2048. Nous avons plusieurs niveaux pour le jeu donc on a mis une énumération pour les différentes sortes de niveau.







## Diagramme des cas d'utilisation

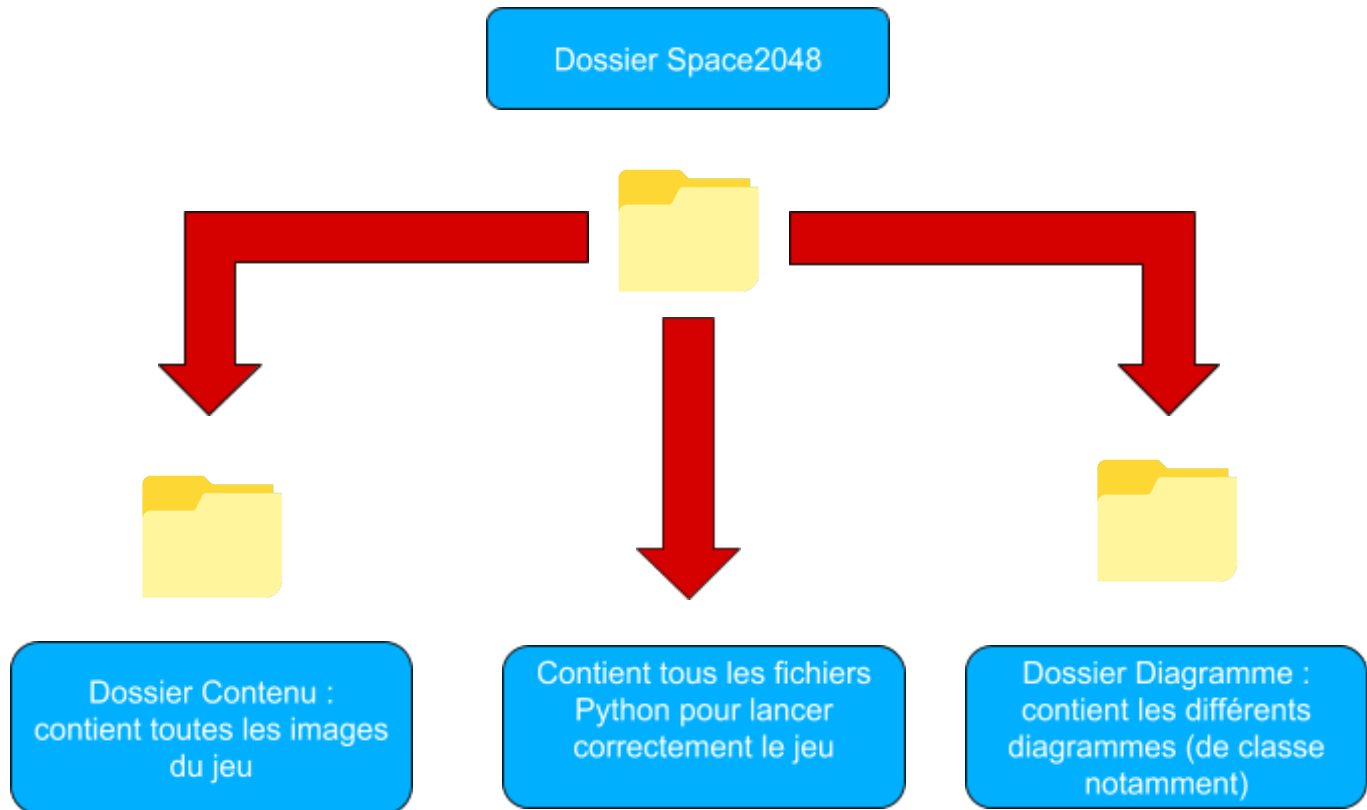


Le diagramme ci-dessus est un diagramme des cas d'utilisation du jeu. Un joueur peut soit cliquer sur les différents niveaux du jeu , soit aller voir les règles ou aller voir les créateurs du jeu. Si le joueur choisit de cliquer sur l'un des niveaux alors le jeu commence avec le niveau choisi.



---

## L'arborescence





---

## La charte graphique

Pour ce qui est de la charte graphique, nous sommes évidemment restés dans le thème de l'espace. Nous avons décidé de choisir ensemble une image de fond pour la fenêtre du jeu, nous servant de point de départ pour la charte graphique. Suite à cela, nous avons donc choisi de garder des tons de bleu, orange et jaune pour nos documents relatifs au jeu et pour le jeu, ainsi que des planètes et une fusée.





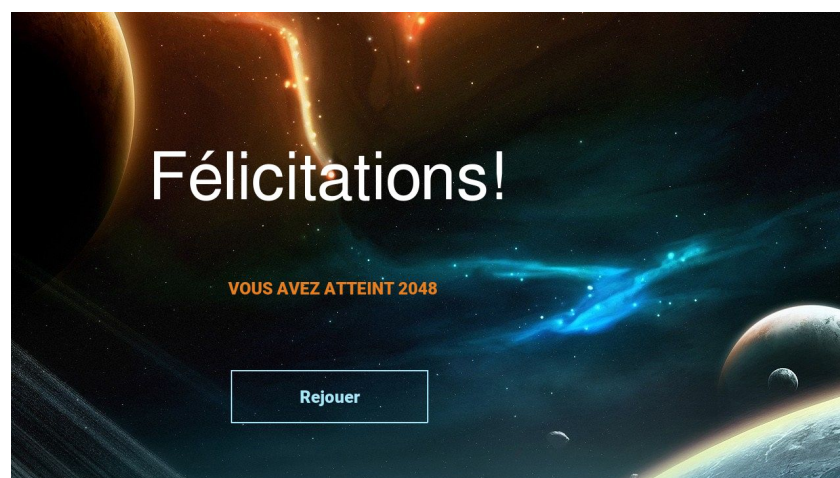
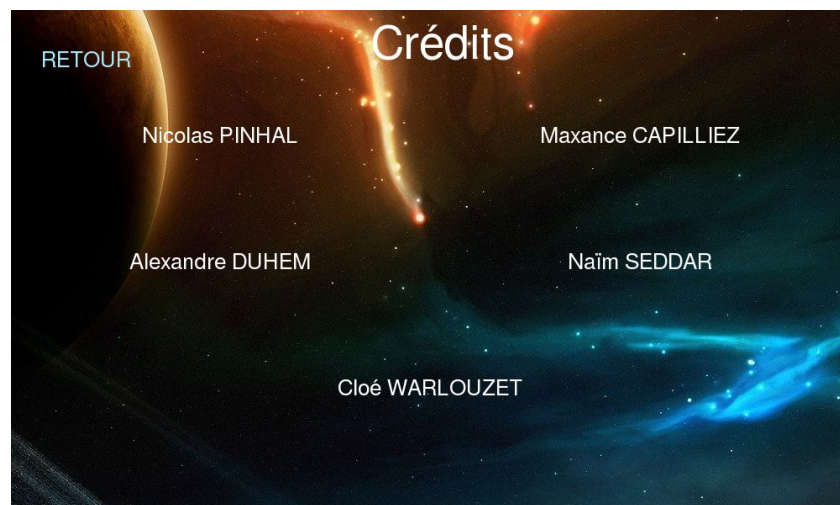
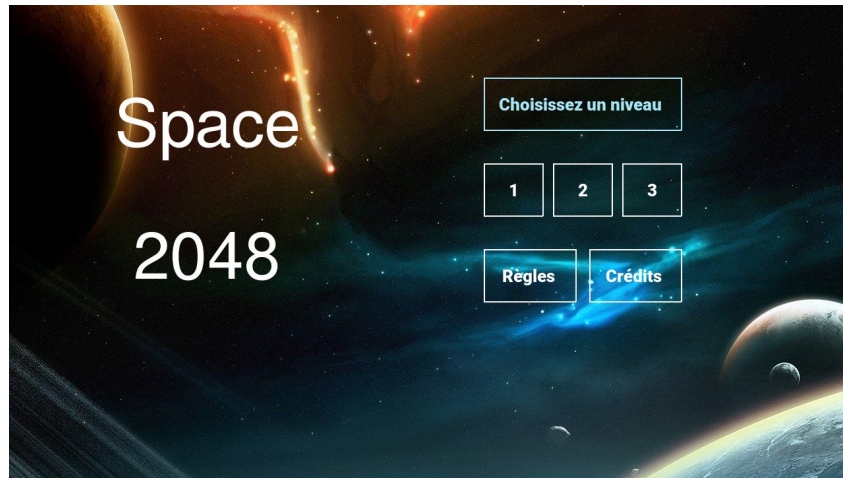
---

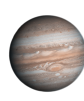
## Code et extraits du jeu

```
def checkColisions(joueur, score):
    j = len(en)
    for i in range(j):
        if en[i].getY() > 720:
            en[i].remove()
        if en[i].getX() <= joueur.getX() + 40 and en[i].getX() >= joueur.getX() - 40:
            if en[i].getY() <= joueur.getY() + 40 and en[i].getY() >= joueur.getY() - 40 and en[i].getvalue() == joueur.getValue():
                if joueur.getValue() == 1024 and en[i].getvalue() == 1024:
                    return "Won"
                else:
                    joueur.setValue(en[i].getvalue()*2)
                    print("valeur du joueur :", joueur.getValue())
                    joueur.setImage()
                    en[i].remove()
                    score = joueur.getValue()
                    return score
            if en[i].getY() <= joueur.getY() + 40 and en[i].getY() >= joueur.getY() - 40:
                if joueur.getValue() == 2 and en[i].getvalue() != 2:
                    return "Game over"

                elif en[i].getvalue() < joueur.getValue():
                    joueur.setValue(en[i].getvalue())
                    joueur.setImage()
                    en[i].remove()
                    score = joueur.getValue()
                    return score
                elif en[i].getvalue() > joueur.getValue():
                    joueur.setValue(joueur.getValue()//2)
                    joueur.setImage()
                    en[i].remove()
                    score = joueur.getValue()
                    return score
    return score
```







---

## BILAN

Notre bilan quant à ce projet est dans l'ensemble plutôt positif. Chacun a travaillé à sa façon sans pour autant laisser les autres faire le travail complet et chacun a pris plaisir à réaliser ses tâches.

Ce projet a permis à certains d'entre nous de découvrir Pygame et à d'autres d'approfondir leurs connaissances là dessus.

La gestion du temps du projet ainsi que du partage de données s'est très bien déroulée grâce à Github et à l'investissement de chaque membre du groupe et à l'organisation.

Pour les points négatifs, nous pensons notamment au fait que nous n'avions pas énormément d'outils (notamment en termes de langage) pour créer ce genre de jeu librement. Néanmoins, de nombreux sites permettent d'apprendre à se servir de langages différents en peu de temps.

Ce bilan reste, dans l'ensemble, très bon et nous pensons ce projet plutôt réussi pour notre part.