

|  |
| --- |
| **2048** |

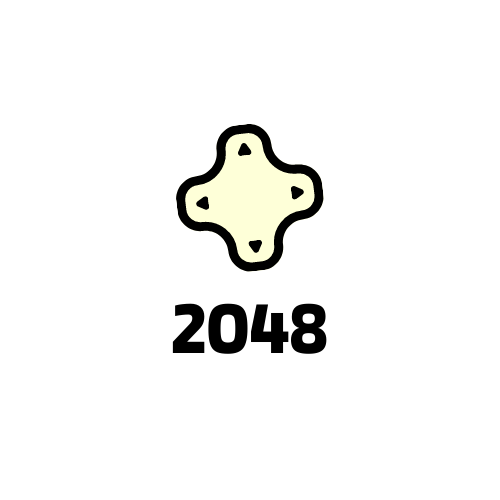




Table des matières

Nussbaum Théo

Theo.NUSSBAUM@cpnv.ch

SI-MIC1b

01.02.2023

[1 Introduction 3](#_Toc128733816)

[1.1 Cadre, description et motivation 3](#_Toc128733817)

[1.2 Objectifs 3](#_Toc128733818)

[1.3 Planification initiale 3](#_Toc128733819)

[2 Analyse 3](#_Toc128733820)

[2.1 Maquettes 3](#_Toc128733821)

[2.2 Stories / tests d’acceptation 4](#_Toc128733822)

[2.3 Stratégie de test 4](#_Toc128733823)

[3 Implémentation 4](#_Toc128733824)

[3.1 Vue d’ensemble 4](#_Toc128733825)

[3.2 Choix techniques 4](#_Toc128733826)

[3.3 Points techniques spécifiques 4](#_Toc128733827)

[4 Tests 5](#_Toc128733828)

[4.1 Tests effectués 5](#_Toc128733829)

[4.2 Erreurs restantes 5](#_Toc128733830)

[5 Conclusions 5](#_Toc128733831)

[6 Annexes 5](#_Toc128733832)

[6.1 Sources – Bibliographie 5](#_Toc128733833)

[6.2 Journal de bord du projet 5](#_Toc128733834)

[6.3 Journal de travail du projet 5](#_Toc128733835)

# Introduction

## Cadre, description et motivation

2048 est un jeu vidéo de type puzzle ou l’on doit faire en sorte qu’une case sois égale a 2048 en les fessent fusionner, je fais ça car c’est un projet de module et car je veux m’améliorer en python.

## Objectifs

Je veux que l’application soit fluide et belle visuellement et aussi sans bug

## Planification initiale

Sprint 1 : Maquette plus début du code

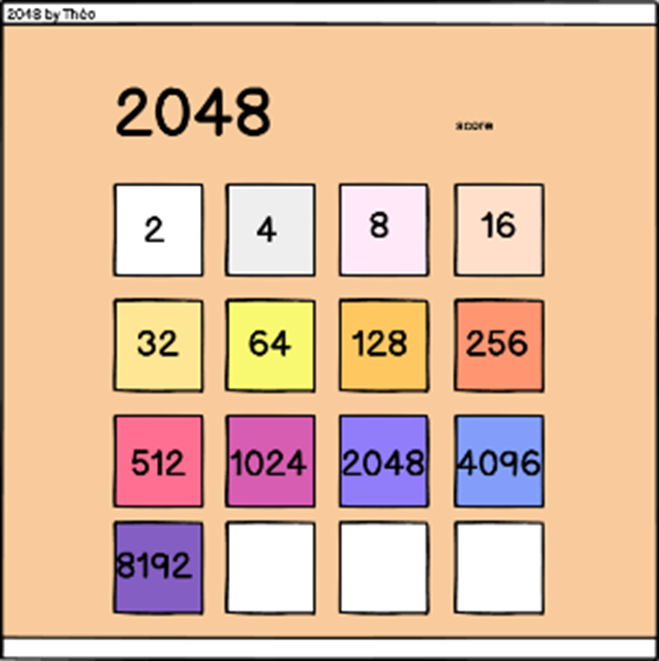
Sprint 2 : Code en version 0.2

Sprint 3 : Code en version 0.3

# Analyse

Le 2048 et un jeu ou l’ont peux bouger des tuiles avec soit w,a,s,d sois les flèches directionnelles, le jeu doit faire apparaître des tuiles de 2 ou de 4 au début du jeu, mon jeu doit être identique que l’image ici même

## Maquettes



## Stories / tests d’acceptation

Les stories et les tests d’acceptation seront définis dans icescrum :

<https://icescrum.cpnv.ch/p/MA2023NUSS/#/project>

## Stratégie de test

Comment j’ai tester mon jeu :

* Mettre 2,2,4,8,16,32,… puis voir si tous les tassement marche

# Implémentation

## Vue d’ensemble

Cette section décrit comment le système à réaliser interagit avec son entourage, en termes :

* D’utilisateur(s) humain(s)
* D’utilisateur(s) logiciel(s) (clients d’une API, par exemple)
* De réseau
* De ressources externes

## Choix techniques

Les divers choix qui ont été faits pour la réalisation du mandat, en termes de :

* Matériel
* Systèmes d'exploitation
* Logiciels tiers (utilitaires, frameworks, navigateurs cible,…)

Pour chaque élément cité, on donnera une justification du choix et on fera la distinction entre ce qui concerne le travail de réalisation et ce qui concerne l’utilisation en production

## Points techniques spécifiques

L’appelé de tasse\_4 a été compliquer a comprendre pour moi je vais donc l’expliquer :

La fonction tasse\_4 tasse tout vers la gauche faut donc lui envoie les nombres dans un ordre persils puis les remettre dans le sens que l’on veut.

Comme ce ci :

[number[line][3], number[line][2], number[line][1], number[line][0]] = tasse\_4(number[line][3], number[line][2], number[line][1], number[line][0])

Ensuit il faut rappeler votre affichage pour moi c’est un fonction nommée display() donc on l’appelle de cette manière :

display()

Puis reste plus qu’à lui assignée une ou plusieurs touche pour pouvoir tasser dans diffèrent direction.

Comme cela :

#attraper les touches  
fen.bind("<Key>",lambda event:move(event))  
def move(event):  
 touche = event.keysym  
 if touche=="w" or touche == "W" or touche =="Up":  
 tasse\_up(event)  
 if touche=="a" or touche == "A" or touche =="Left":  
 tasse\_left(event)  
 if touche=="s" or touche == "S" or touche =="Down":  
 tasse\_down(event)  
 if touche=="d" or touche == "D" or touche =="Right":  
 tasse\_right(event)

# Tests

## Tests effectués

Les tests sont fait dans IceScrum.

## Erreurs restantes

Pas d’erreur connu

# Conclusions

Développez en tous cas les points suivants:

* Objectifs atteints / non-atteints
* Comparaison entre ce qui avait prévu et ce qui s’est passé, en termes de planning et (éventuellement) de budget
* Points positifs / négatifs
* Difficultés particulières
* Suites possibles pour le projet (évolutions & améliorations)

# Annexes

## Sources – Bibliographie

Je n’ai pas utilisé de site mais pas mal d’aide de mes collège (Rayan cardamon, John Jacard)

## Journal de bord du projet

|  |  |
| --- | --- |
| **Date** | **Evénement** |
|  |  |

## Journal de travail du projet

|  |  |  |
| --- | --- | --- |
| **Date** | **Durée (h)** | **Travail effectué** |
| 01.02.2023 | 1h30 | Introduction du module plus faire la maquette |
| 03.02.2023 | 5h25 | Absent (rattraper) |
| 08.02.2023 | 1h30 | Code python |
| 10.02.2023 | 5h25 | Finir le code, remplir le word plus rendre le travail du sprint 1 |
| 22.02.2023 | 1h30 | Correction de tasse\_4 |
| 24.02.2023 | 5h25 | Terminer la correction puis le bon fonctionnement de tasse\_4 |
| 01.03.2023 | 1h30 | Ajoute de tasse\_4 dans mon code + l’appelé de tasse\_4 |
| 03.03.2023 | 5h25 | Finir le code en V0.2, remplir le word plus rendre le travail du sprint 2 |