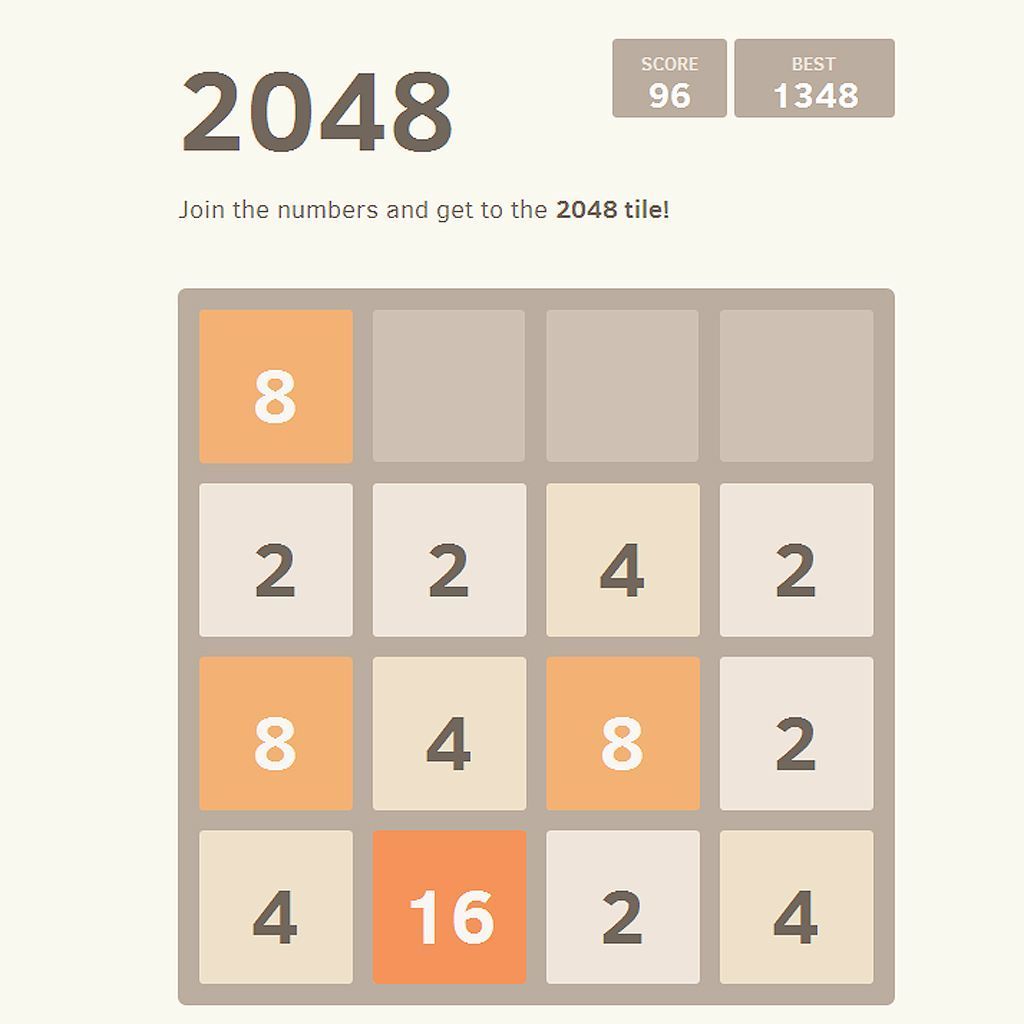


|  |
| --- |
| **2048** |



Fonseca Diogo Christophe

Christophe.FONSECA-DIOGO@cpnv.ch



Table des matières

SI-C1a

23.02.2023

[1 Introduction 3](#_Toc131667004)

[1.1 Cadre, description et motivation 3](#_Toc131667005)

[1.2 Objectifs 3](#_Toc131667006)

[1.3 Planification initiale 3](#_Toc131667007)

[2 Analyse 3](#_Toc131667008)

[2.1 Maquettes 4](#_Toc131667009)

[2.2 Stories / tests d’acceptation 4](#_Toc131667010)

[2.3 Stratégie de test 4](#_Toc131667011)

[3 Implémentation 4](#_Toc131667012)

[3.1 Vue d’ensemble 5](#_Toc131667013)

[3.2 Choix techniques 5](#_Toc131667014)

[3.3 Points techniques spécifiques 5](#_Toc131667015)

[3.4 Pseudo-code 6](#_Toc131667016)

[4 Tests 6](#_Toc131667017)

[4.1 Tests effectués 6](#_Toc131667018)

[4.2 Erreurs restantes 6](#_Toc131667019)

[5 Conclusions 7](#_Toc131667020)

[6 Annexes 7](#_Toc131667021)

[6.1 Sources – Bibliographie 7](#_Toc131667022)

[6.2 Voici les personnes qui m’ont aidé dans le projet 8](#_Toc131667023)

[6.3 Journal de travail 8](#_Toc131667024)

# Introduction

## Cadre, description et motivation

Le projet 2048 a été imaginé comme projet par le CPNV pour le module ICT-431.

Ce choix a été établit car nous n'allons plus utiliser le langage de programmation C mais le Python. Le jeu 2048 est un jeu qui se joue avec une grille de 4x4, le but du jeu est de tasser les tuiles semblables pour pouvoir augmenter le numéro de la tuile jusqu’à atteindre le nombre 2048 pour gagner.

Ce projet pourra lourdement solliciter mes compétences en programmation Python.

## Objectifs

Objectifs généraux

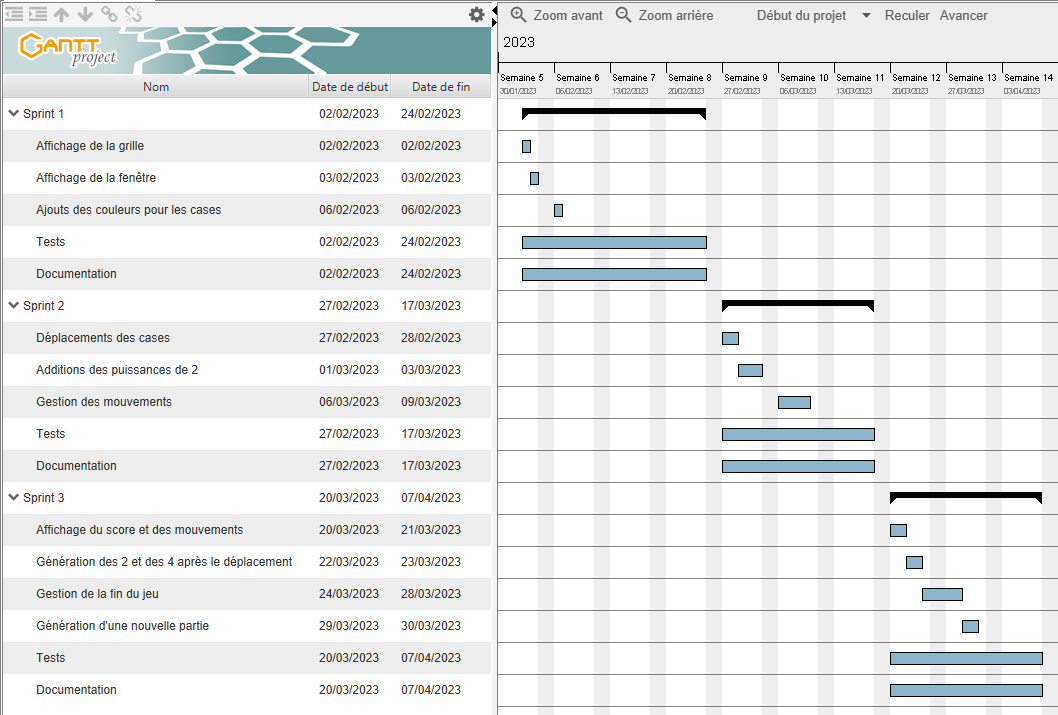
* Approfondir mes connaissances en Python.
* Savoir gérer un projet.
* Etre capable de faire une documentation détaillée.

Objectifs SMART

* Pouvoir avoir une grille de jeu 4x4.
* Afficher correctement les puissances de deux sur la grille du jeu.
* Que chaque nombre aient une couleur de fond différente dans la grille.
* A chaque clic sur la flèche directionnelle je veux pouvoir ajouter les nombres adjacents qui ont la même puissance de deux.
* Que le jeu prenne fin du moment que le nombre atteigne 2048.

## Planification initiale

Voici la planification globale du projet sur Gantt Project :



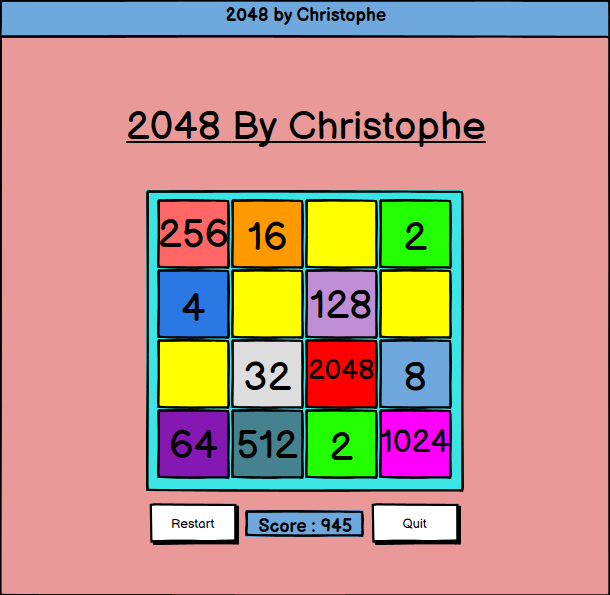
Pour une planification plus détaillée cela a été fait dans IceScrum :

https://icescrum.cpnv.ch/p/2048CHRIST/#/project

# Analyse

Ce paragraphe présente l’analyse avec les maquettes, des stories qui ont été créés et la stratégie de test.

## Maquettes



## Stories / tests d’acceptation

Les stories se trouvent sur Icescrum : <https://icescrum.cpnv.ch/p/2048CHRIST/#/backlog/all>

## Stratégie de test

Pour le développement du jeu 2048, le jeu sera programmé en python et sera installé sur la même machine où j’écrirai le code.

Je préparerai

* Le code de la grille du jeu 2048 sur PyCharmEdu.
* La fonction de tassage sur un autre fichier python dans PyCharmEdu pour pouvoir tester les différents cas qui seront possible sur le jeu 2048.

Sur la machine où je développe le jeu 2048 je ferai une série de tests unitaires pour voir si ma fonction n’a pas de problèmes, d’intégrations pour vérifier que si j’ajoute la fonction à la grille le code ne subisse pas une erreur, fonctionnels pour vérifier que le système fait ce qu’on attend de lui, et systèmes pour voir si le code fonctionne après avoir ajouté toutes les fonctionnalités dans le jeu.

Les tests seront faits par moi et certains de mes collègues de classe. Les tests de mes collègues seront faits sur d’autres machines Windows 10 pour que l’on puisse vérifier si le jeu fonctionne bien et s’adapte aux différentes configurations.

Les tests seront faits sur des machines Windows.

# Implémentation

## Vue d’ensemble

Le système réagit dès que l'utilisateur clique sur les flèches du clavier ou les touches WASD. Il réagira aussi lorsque l’utilisateur cliquera sur le bouton pour refaire une partie.

## Choix techniques

Pour ce projet, le matériel utilisé est le pc fixe fourni par le CPNV.

En terme de sauvegarde, nous avons un accès sur un serveur distant de fichiers pour que l'on puisse travailler en dehors du CPNV (Perso).

La machine est sous le système d'exploitation Windows 10.

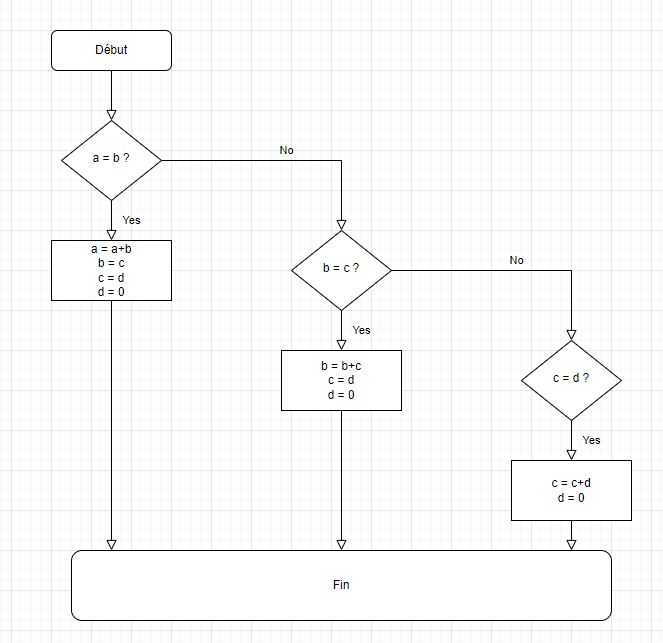
Les logiciels utilisés sont les suivant Pycharm Edu 2022.2, Visual Studio Code.

Pour le code, nous avons utilisé Python avec la version 3.8. Python est un langage de programmation interprété, multiparadigme et multiplateformes.

Nous avons utilisé aussi un module en Python qui se nomme Tkinter, Tkinter est principalement utilisé pour la conception d'interface graphique.

## Points techniques spécifiques

Voici le diagramme de flux qui compose la base de la fonction de tassage, cependant le programme a été codé avec un tableau



## ****Pseudo-code****

Voici un pseudocode pour savoir si on a gagné ou perdu le jeu 2048 :

                Fonction tablestate() :

                               Création d’une liste vide

                               Pour i qui va de 0 à 4 :

                                               Si une case = 2048 :

                                                               Afficher le messagebox : Vous avez gagné

                                                               Si le joueur répond oui pour rejouer :

                                                                               Fonction reset\_game()

                                                                               Fonction obj\_refresh()

                                                               Sinon :

                                                                               Quitter le jeu

                                               Pour j qui va de 0 à 4 :

                                                               S’il y a des cases = 0 :

                                                                            Ajouter dans la liste vide soit 2, soit 4

                               Si la longueur de la liste vide = 0 :

                                               Fonction test de possibilité de mouvement

                                               S’il y a possiblement un mouvement :

                                                               Alors continuer le jeu

                                               Sinon :

                                                               Afficher le messagebox : Vous avez perdu

                                                               Si le joueur répond oui pour rejouer :

                                                                               Fonction reset\_game()

                                                                               Fonction obj\_refresh()

                                                               Sinon :

                                                                               Quitter le jeu

# Tests

## Tests effectués

<https://icescrum.cpnv.ch/p/2048CHRIST/#/project>

## Erreurs restantes

* Sprint 1 : Aucune Erreur
* Sprint 2 : Aucune Erreur
* Sprint 3 : Dans le code nous avons rencontré une erreur seulement dans certains cas spéciaux pour la fonction de tassage (mix). Un des cas concerné est le suivant :

Si nous avons dans la grille [2 ;2 ;2 ;4] et que nous faisons un mouvement vers la droite le tassage ne se fait pas comme il le faudrait, la grille passe en [0 ;4 ;2 ;4] tandis qu’elle devrait passer en [0 ;2 ;4 ;4]. La solution serait de changer le sens de tassage de le faire passer de gauche à droite en droite à gauche.

# Conclusions

* Pour finir, je pense que les objectifs demandés par le chef de projet sont tous accomplis. Le but était tout simplement que l’on refasse le célèbre jeu 2048 avec Tkinter en 1 trimestre. Ce qui à été fait, le jeu 2048 que j’ai produit remplit l’ensemble des fonctionnalités du cahier des charges. Le score augmente à chaque addition de cases adjacentes et j’ai ajouté les mouvements pour que le joueur ait cette info en plus.
* Grâce à ce projet je sais à partir de maintenant mieux coder en Python et ai enfin compris les boucles for. Ce projet à augmenter considérablement mes compétences en programmation, j’ai pu mettre à rude épreuve ce que je savais faire en Python.
* Le point négatif de ce projet était tout simplement de coder avec les normes de codage et de ne pas oublier de remplir le journal de travail.
* Certaines difficultés sont apparues telle que la fonction qui compte les mouvements et la fonction mix. Pour la suite de ce projet je pense que j’ajouterai le highscore pour que l’utilisateur ait un côté compétitif. Je pense aussi ajouter une fonction cachée, qui permet à l’aide d’une combinaison de touches de générer 2 cases avec les valeurs 1024, le joueur pourra ainsi terminer sa partie plus vite. Je pourrais aussi créer moi-même une messagebox de toute pièce avec Tkinter.

# Annexes

## Sources – Bibliographie

* **W3shcool :** <https://www.w3schools.com/python/default.asp>

Nous avons utilisé l’aide de W3school pour les boucles For, les dictionnaires et les listes.

* **Tkinter 8.5 reference : a GUI for Python :** <https://anzeljg.github.io/rin2/book2/2405/docs/tkinter/key-names.html>

Nous avons utilisé l’aide de Tkinter 8.5 reference : a GUI for Python pour le nom des keybinds.

* **Documentation Random de Python :** <https://docs.python.org/3/library/random.html>

Nous avons utilisé l’aide de la documentation du module random

pour l’affichage des 2 et des 4 puis ajouter le pourcentage de chance

d’apparition des 4.

* **Documentation MessageBox de Python :** <https://docs.python.org/3/library/tkinter.messagebox.html>

Nous avons utilisé l’aide de la documentation du module MessageBox

pour afficher si le joueur gagne ou perds pendant le jeu.

* **Documentation Tkinter Graphical Interface :**

<https://docs.python.org/3/library/tkinter.html#module-tkinter>

Nous avons utiliser l’aide de la documentation du module Tkinter

pour faire une interface graphique époustouflante.

* **GeeksForGeeks :**

<https://www.geeksforgeeks.org/understanding-for-loop-in-python/>

Nous avons utilisé l’aide de GeeksForGeeks pour mieux comprendre

comment maitriser les boucles for.

* **Github :**

[**https://github.com/**](https://github.com/)

Nous avons utilisé l’aide de Github pour des recherches concernant

L’ajout d’une commande à un bouton.

## Voici les personnes qui m’ont aidé dans le projet

* Avec PERIYASAMY-GHANESHAN Judah-Abishek Nous nous sommes aidé pour le code et surtout la documentation.
* Brad-Dylan Munoz m’a aidé au code lorsque j’avais des erreurs.
* Mme Frederique ANDOLFATTO m’a aidé pour le nombre de mouvements.

## Journal de travail

