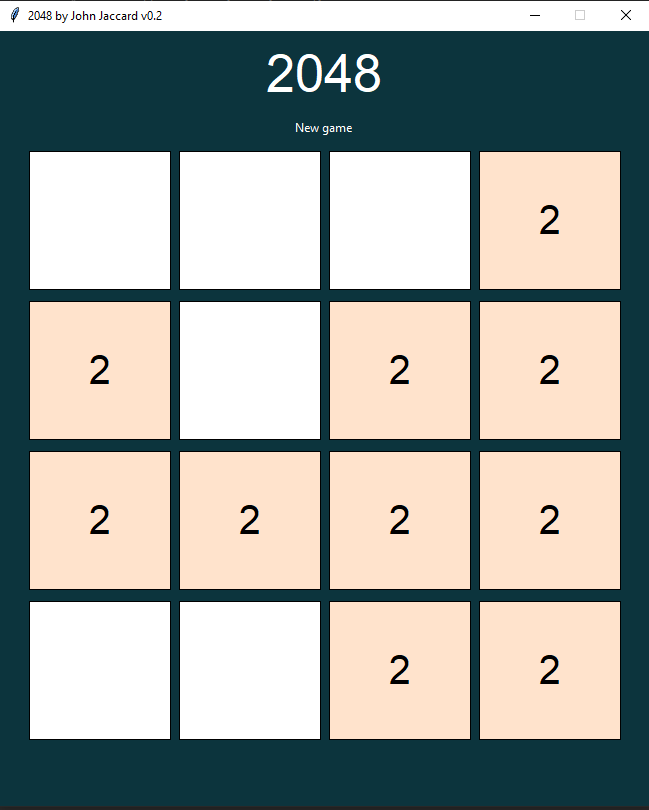


|  |
| --- |
| 2048 |

Illustration:



Jaccard, John

John.jaccard@cpnv.ch

SI-MI1b

03.02.23

Table des matières

[1 Introduction 3](#_Toc130551668)

[1.1 Cadre, description et motivation 3](#_Toc130551669)

[1.2 Objectifs 3](#_Toc130551670)

[1.3 Planification initiale 3](#_Toc130551671)

[2 Analyse 4](#_Toc130551672)

[2.1 Maquette 4](#_Toc130551673)

[2.2 Stories / tests d’acceptation 4](#_Toc130551674)

[2.3 Stratégie de test 4](#_Toc130551675)

[3 Implémentation 5](#_Toc130551676)

[3.1 Vue d’ensemble 5](#_Toc130551677)

[3.2 Choix techniques 5](#_Toc130551678)

[3.3 Points techniques spécifiques 6](#_Toc130551679)

[4 Tests 8](#_Toc130551680)

[4.1 Tests effectués 8](#_Toc130551681)

[4.2 Erreurs restantes 8](#_Toc130551682)

[5 Conclusions 8](#_Toc130551683)

[6 Annexes 9](#_Toc130551684)

[6.1 Sources – Bibliographie 9](#_Toc130551685)

[Journal de bord du projet 9](#_Toc130551686)

[6.2 Journal de travail du projet 9](#_Toc130551687)

# Introduction

## Cadre, description et motivation

Dans le cadre du cours MA-20 et 431 au CPNV,nous faisons un jeu Python,le 2048.

Le but est de créer le jeu 2048 en python en plusieurs sprints qui sont eux-même découpés en étapes et apprendre à utiliser des logiciels/sites d’organisations de projets comme Balsamiq ou Icescrum

## Objectifs

Créer le jeu 2048 en python et apprendre à utiliser des logiciels/sites d’organisations de projets comme Balsamiq ou Icescrum, le jeu utilise des touches directionelles et wasd afin de tasser des blocs dans une directions ce qui fera augmenter un score selon les blocs tassés et le but est d’arriver à avoir une tuile « 2048 » en tassant les blocs similaires

## Planification initiale

-Sprint 1 :

Création maquette +affichage en python [fin 10 février 2023]

-Sprint 2

Tassage de blocs et l’assigner à des touches [fin 10 mars 2023]

-Sprint 3 :

Apparition de tuiles, ajout de défaite et victoire +quelques petites fonctionnalitées supplémentaires [fin 24 mars 2023]

# Analyse

Jeu 2048 commencant avec une maquette ensuite par l’affichage de cases et finalement avoir un jeu fonctionnel bind avec des touches qui peut tasser avoir un score et faire apparaître un 2 à chaque mouvement

## Maquette

## 

## 

## Stories / tests d’acceptation

Les tests sont tous dans Icescrum

## Stratégie de test

Version 0.2 (tests sur icescrum pour plus de détails)

* Tasse 4 fonctionne en terminal
* Tasse 4 fonctionne dans toutes les directions et dans Tkinter

# Implémentation

## Vue d’ensemble

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

* Soit les touches directionnelles soit les touches « wasd » permettent à l’utilisateur de tasser et bouger les cases dans la direction souhaitée
* Un fichier texte «Gamesave» permet de sauvegarder la partie en cours et que si on quitte le jeu en le relancant, on soit toujours dans notre partie
* Un deuxième fichier texte «Score» qui permet comme le premier fichier texte de garder le score d’une partie en cours si on quitte le jeu
* Un dernier fichier texte «Highscore» permet d’enregistrer le meilleur score et de l’augmenter si le score est plus élever que le Highscore
* Quelques images mise dans le jeu servant de bouton comme le bouton undo ou les écrans de défaite et de victoire
* Un bouton caché qui permet d’ouvrir le menu «DEBUG » en cliquant dans le coin inférieur droit de la fenêtre

## Choix techniques

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

* Matériel : nous utilisons python 3.8 avec Pycharm Edu car pycharm et assez simple d’utilisation et aussi car nous l’utilisons depuis le début de notre formation
* Systèmes d'exploitation : Nous utilisons Windows 10
* Logiciels tiers : nous utilisons Word pour le document de projet, Balsamiq pour schématiser le projet et concevoir la colorimétrie, Icescrum pour le planifier et le découper avec des objectifs et github qui permet de sauvegarder notre travail en ligne

## Points techniques spécifiques

Application de tasse Cette section contient au minimum deux sous-sections qui décrivent chacune un élément technique précis, qui n’est pas évident et qui sert à comprendre le détail de fonctionnement du système.

Il peut s’agir de :

* Application de tasse 4 dans Tkinter :

En premier on récupère les touches du clavier avec un bind (lui-même assigné à une fonction) et on les assignes à une variable ce qui permet d’effectuer une certaine fonction selon la touche appuyée(avec une condition if et aussi utiliser différentes touches pour les directions comme wasd, WASD et les 4 flèches)

On prend dans un certain ordre pour gauche et bas abcd pour droite et haut dcba, le tout dans une boucle for de range(4) ce qui effectue la fonction 4 fois permettant de l’effectuer pour les 4 lignes ou colonnes

* Écran de fin :

Cette écran en une fonction qui va :

-On sauvegarde le score pour qu’il ne soit pas affectés par les opérations sur les copies de table suivantes :

-Créer deux copies de la table de base

La première va permettre de simplement avoir une sauvegarde de la table sans y toucher

Table de base

Table 2

Table1

La deuxième va être utilisé dans la suite de la fonction :

-On va ensuite utiliser notre fonction sur chaque ligne/colonne tasse4 dans chacune des directions possibles.

Quand la table n’a aucun changement (on voit ça en comparer la copie de table 1 à la copie de table 2) et si dans la ligne/colonne on ne détecte aucun mouvement, on ajoute 1 à une variable (initialiser à 0 au début de la fonction) et vu qu’on à 4 lignes/colonnes si en allant à droite rien ne change on ajoute 4.

(Ex pour la gauche)

Var+=0

Table1

Var+=4

?

Table 2

Table 2

Table1

Table 2

Table1

Table 2

Après l’avoir fait dans les quatre directions, la variable est égale à 16 cela veut dire que plus rien n’est faisable et on affiche donc l’image avec l’écran de défaite, en mettant un bouton afficher le jeu qui permet de voir la partie perdue.

-On remet le score avec la copie faite au début afin qu’il ne soit pas altéré par ces manipulations

# Tests

## Tests effectués

Tous les tests sont effectués dans Icescrum

## Erreurs restantes

S'il reste encore des erreurs :

* Aucune remarquée pour l’instant dans la version 0.2
* Juste certains décalages de sauvegardes selon quand on quitte le jeu

# Conclusions

Le projet était intéressant car beaucoup de choses avait été vues auparavant mais peu pratiquées, aussi certaines fonctions nécessitaient de bons moments de réflexions sur leurs fonctionnements et façons de les appliquer.

La méthode de planification était aussi nouvelle et intéressante à apprendre à l’utiliser avec Icescrum en la découper en sprints qui était découpés en storys,

De même pour le logiciel de schématisation qui était utile et donnée ensuite l’idées de l’objectif à atteindre en codant ensuite.

J’ai surtout personnellement apprécier la partie « bonus » ou j’ai eu l’occasion de développer pleins de petites fonctions et de découvrir beaucoup de choses avec python et plus spécialement Tkinter ;

La partie d’application de la fonction « Tasse4 » était très intéressantes.

En bref j’ai beaucoup l’aspect code relativement libre même si quelques exigences demandées.

# Annexes

## Sources – Bibliographie

Aide de Léo pour fonction tasse 4

Aide de Thibault pour fonction New Game

Scan dossier : https://www.tresfacile.net/liste-combobox-tkinter/#:~:text=Tkinter%20Combobox%20est%20une%20sorte,indiquant%20tous%20les%20choix%20possible.

## Journal de bord du projet

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

## Journal de travail du projet

|  |  |  |
| --- | --- | --- |
| **Date** | **Durée (h)** | **Travail effectué** |
| 01.02.23 | 1.5h | Création maquette et choix des couleurs sur Balsamiq |
| 03.02.23 | 1.5h | Découverte Icescrum et création structure du projet sur Icescrum |
| 03.02.23 | 2.25h | Création affichage python avec label et couleurs automatiques |
| 08.02.23 | 1.5h | Début fonction display |
| 10.02.23 | 1h | Finir la fonction display+New game |
| 15.02.23 | 1.5h | Tasse 4 en ligne de commandes |
| 17.02.23 | 5.25h | Commencer application tassage dans toutes les directions |
| 22.02.23 | 1.5h | Continuer fonction tassage pour l’adapter dans les autres direction |
| 01.03.23 | 1.5h | Appartition 2 dans le jeu + score |
| 03.03.23 | 5.25h | Compléter Icescrum et doc projet plus remettre une version 0.2 propre |
| 08.03.23 | 1.5h | Ecran de win(problème d’image encore |
| 10.03.23 | 5.25h | Ecran de win ok,ecran de loose fait aussi |
| 14.03.23 | 1.5h | Ajout debug mod(cheat permettant de spawn qqch) |
| 17.03.23 | 5.25h | Plusieurs nouvelles fonctionnalitées du menu débug (rainbow mod, random permettant de changer l’emplacement des tuiles aléatoirement et clear Highscore) |
| 22.03.23 | 1.5h | Absent |
| 24.03.23 | 3.75h | Ajout .wav player, complétion de :document projet, Icescrum et maquette, et rendu des documents |