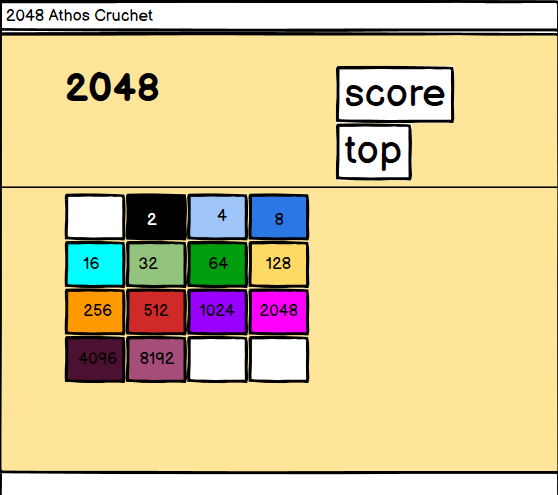


|  |
| --- |
| Projet 2048 |



Cruchet, Athos

[Athos.cruchet@cpn](mailto:Athos.cruchet@cpn)v.ch



SI-CIB

11/02/232

Table des matières

[1 Introduction 3](#_Toc130561511)

[1.1 Cadre, description et motivation 3](#_Toc130561512)

[1.2 Objectifs 3](#_Toc130561513)

[1.3 Planification initiale 3](#_Toc130561514)

[2 Analyse 3](#_Toc130561515)

[2.1 .Maquettes 3](#_Toc130561516)

[2.2 Stories / tests d’acceptation 3](#_Toc130561517)

[2.3 Stratégie de test 4](#_Toc130561518)

[3 Implémentation 4](#_Toc130561519)

[3.1 Vue d’ensemble 4](#_Toc130561520)

[3.2 Choix techniques 4](#_Toc130561521)

[3.3 Points techniques spécifiques 4](#_Toc130561522)

[L’assignation des touches « a, w, d, s» 4](#_Toc130561523)

[4 Tests 6](#_Toc130561524)

[4.1 Tests effectués 6](#_Toc130561525)

[4.2 Erreurs restantes 6](#_Toc130561526)

[5 Conclusions 6](#_Toc130561527)

[6 Annexes 6](#_Toc130561528)

[6.1 Sources – Bibliographie 6](#_Toc130561529)

[6.2 Journal de bord du projet 6](#_Toc130561530)

[6.3 Journal de travail 7](#_Toc130561531)

# Introduction

## Cadre, description et motivation

Dans le cadre du CPNV, dans le module ICT-431, enseigné par M.Chavey, je réalise la programmation du jeu 2048.

## Objectifs

L’objectif de ce module est de concevoir la programmation du jeu 2048 sur python.

## Planification initiale

La programmation du jeu est divisée en plusieurs partie, qu’on appelle les sprints.

Sprint 1 : création de la maquette + valeurs avec couleurs + affichage du jeu conforme au tableau en mémoire et à la maquette.

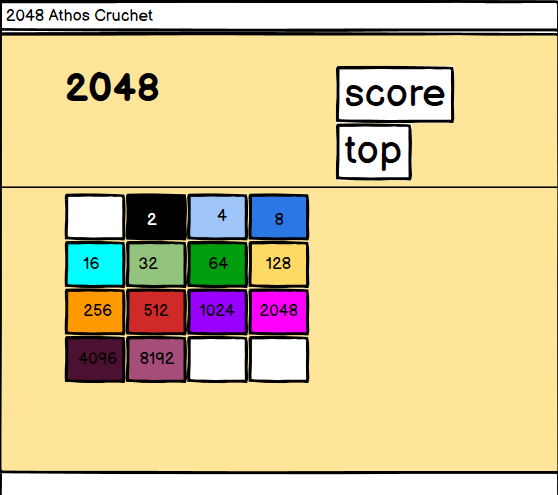
Sprint 2 : fonction tasse 4 + comptage des mouvements + tasser tout le jeu dans une direction avec les touches du clavier.

Sprint 3 : Apparition aléatoire de tuiles de 2 ou de 4 + Tests de fin de jeu (perdu/gagné)

Défense : Démonstration du projet devant la classe

# Analyse

## .Maquettes



## Stories / tests d’acceptation

Les tests ont été réalisés sur : https://icescrum.cpnv.ch/p/MA20MA2020/#/planning/3095/sprint/3107/details

## Stratégie de test

Ma stratégie de test est écrite sur icescrum.

# Implémentation

## Vue d’ensemble

Ce programme fonctionne sur un ordinateur et interagit avec le clavier pour les déplacements.

L’interface a été faite avec tkinter et python.

## Choix techniques

Je suis sur un système d’exploitation windows et j’utilise pycharm comme éditeur.

Comme logiciel tiers, dans le cadre du module, j’utilise également icescrum et github.

## Points techniques spécifiques

### L’assignation des touches « a, w, d, s»

Dans un premier temps, nous devons définir les mouvements avec des fonctions comme dans l’exemple ci-dessous :

Une image contenant texte

Description générée automatiquement

Il est nécessaire, par la suite, de répéter l’opération pour l’assignation de chaque touche voulue. (vers la droite, le bas et vers le haut)

Une image contenant texte

Description générée automatiquement

Par la suite, il faut assigner les fonctions définies avec les touches du clavier.

Fonction "gagner".

Voici une explication étape par étape:

Une image contenant texte

Description générée automatiquement

La première ligne contient "global first2048", ce qui signifie que nous allons utiliser une variable appelée "first2048" qui est définie en dehors de cette fonction. Cela permet à la fonction d'accéder à la variable même si elle n'a pas été déclarée à l'intérieur de la fonction elle-même.

La boucle "for line in range(len(numbers))" parcourt chaque ligne dans la grille de nombres.

La boucle "for col in range(len(numbers[line]))" parcourt chaque colonne dans la ligne actuelle.

La condition "if first2048 == 1" vérifie si la variable "first2048" est égale à 1. Si c'est le cas, cela signifie que le joueur n'a pas encore atteint le nombre 2048.

La condition "if numbers[line][col] == 2048" vérifie si le nombre actuel dans la grille est égal à 2048. Si c'est le cas, cela signifie que le joueur a atteint le nombre 2048 et a gagné le jeu.

Si le joueur a gagné, la fonction "messagebox.showinfo" affiche une boîte de dialogue contenant le message "Vous avez gagné!".

La ligne suivante "first2048 = 0" met à jour la variable "first2048" pour indiquer que le joueur a gagné et ne peut plus gagner à nouveau.

Enfin, la fonction "display()" est appelée pour mettre à jour l'affichage de la grille de nombres.

En résumé, cette fonction vérifie si le joueur a atteint le nombre 2048 dans la grille de nombres. Si oui, elle affiche un message de victoire et met à jour l'affichage de la grille.

# Tests

## Tests effectués

Tests effectués sur Icescrum. Les tests sont visible dans chaque sprint et pour chaque étapes.

Voici le lien de mon repository :

https://icescrum.cpnv.ch/p/MA20MA2020/#/planning/3095/sprint/3107/details

## Erreurs restantes

Programme sans erreur lors de rendre le sprint 1.

Programme sans erreur lors de rendre le sprint 2.

Programme sans erreur lors de rendre le sprint 3.

# Conclusions

Les objectifs donnés pour la réalisation du jeu ont été accompli.

Le module et le projet étant bien organisé grâce à une bonne préparation, le déroulement du code s’est bien passé.

Plusieurs points positifs :

1. Cela m’a fait beaucoup apprendre sur python, je suis plus à l’aise avec la programmation.
2. Le fait également que le jeu ait été fait sous forme de projet facilite l’organisation et la compréhension de la tâche.

J’ai rencontré quelques difficultés au cours du code comme certains bugs, comme par exemple le score qui comptait le nombre de déplacement plusieurs fois et qui, par conséquent, influençait dans le véritable score lors de la partie.

Nous n’avions pas besoin d’un budget pour la réalisation de ce projet.

Il est encore possible de réaliser quelques modifications ou personnalisations dans le jeu tel qu’un changement de couleur lors de la victoire ou, par exemple, insérer une mémoire qui stockerait les scores de différents joueurs. Cela donnerait la possibilité d’un « meilleur score ».

# Annexes

## Sources – Bibliographie

Carlos – pour l’affichage et plusieurs fonctions

Fakime – score, gagner/perdu

Esteban – plusieurs fonctions

## Journal de travail

|  |  |  |  |
| --- | --- | --- | --- |
| **Jour** | **Temps [h]** | **Description** | **Remarques** |
| 01/02/ | 1,5h | Maquette |  |
| 03/02/ | 5h | Affichage |  |
| 08/02/ | 1,5h | Avancement dans le programme |  |
| 10/02 | 2h | Rendre maquette + affichage |  |
| 22/02/ | 1,5h | Avancement dans le programme + doc (icescrum, github) |  |
| 24/02 | - | Absent |  |
| 08/03/ | 1,5h | Avancement dans le programme |  |
| 10/03/ | 5h | Fin sprint 2 |  |
| 15/03/ | 1,5h | Avancement dans le programme |  |
| 17/03/ | - | absent |  |
| 22/03 | 1,5h | Avancement dans le programme |  |
| 24/03 | 2,75h | Finalisation projet + doc |  |