

|  |
| --- |
| 2048 |

Arifi, Elod

Elod.pro1@gmail.com



SI-C1b

02.02.2023

Table des matières

[1 Introduction 3](#_Toc125451210)

[1.1 Cadre, description et motivation 3](#_Toc125451211)

[1.2 Objectifs 3](#_Toc125451212)

[1.3 Planification initiale 3](#_Toc125451213)

[2 Analyse 3](#_Toc125451214)

[2.1 Maquettes 3](#_Toc125451215)

[2.2 Stories / tests d’acceptation 3](#_Toc125451216)

[2.3 Stratégie de test 3](#_Toc125451217)

[3 Implémentation 3](#_Toc125451218)

[3.1 Vue d’ensemble 3](#_Toc125451219)

[3.2 Choix techniques 3](#_Toc125451220)

[3.3 Points techniques spécifiques 3](#_Toc125451221)

[3.3.1 Point 1 3](#_Toc125451222)

[3.3.2 Point 2 3](#_Toc125451223)

[3.3.3 Point … 3](#_Toc125451224)

[4 Tests 3](#_Toc125451225)

[4.1 Tests effectués 3](#_Toc125451226)

[4.2 Erreurs restantes 3](#_Toc125451227)

[5 Conclusions 3](#_Toc125451228)

[6 Annexes 3](#_Toc125451229)

[6.1 Sources – Bibliographie 3](#_Toc125451230)

[6.2 Journal de bord du projet 3](#_Toc125451231)

# Introduction

## Cadre, description et motivation

Mon jeu a été créé dans le cadre d’un projet personnelle proposé en classe.

2048 est un jeu de logique où l’on doit arriver à la dernière tuile qui est pour le mien c’est 8192. Il faut assimiler toutes les paires faites depuis le nombre 2. Si toutes les cases sont pleines, vous avez perdu.

Je suis motivé à faire ce projet parce que nous pouvons avancer dans notre rythme, travailler sur un jeu complexe pour notre niveau et permettre de mettre notre originalité dans ce jeu avec nos propre couleurs etc.

## Objectifs

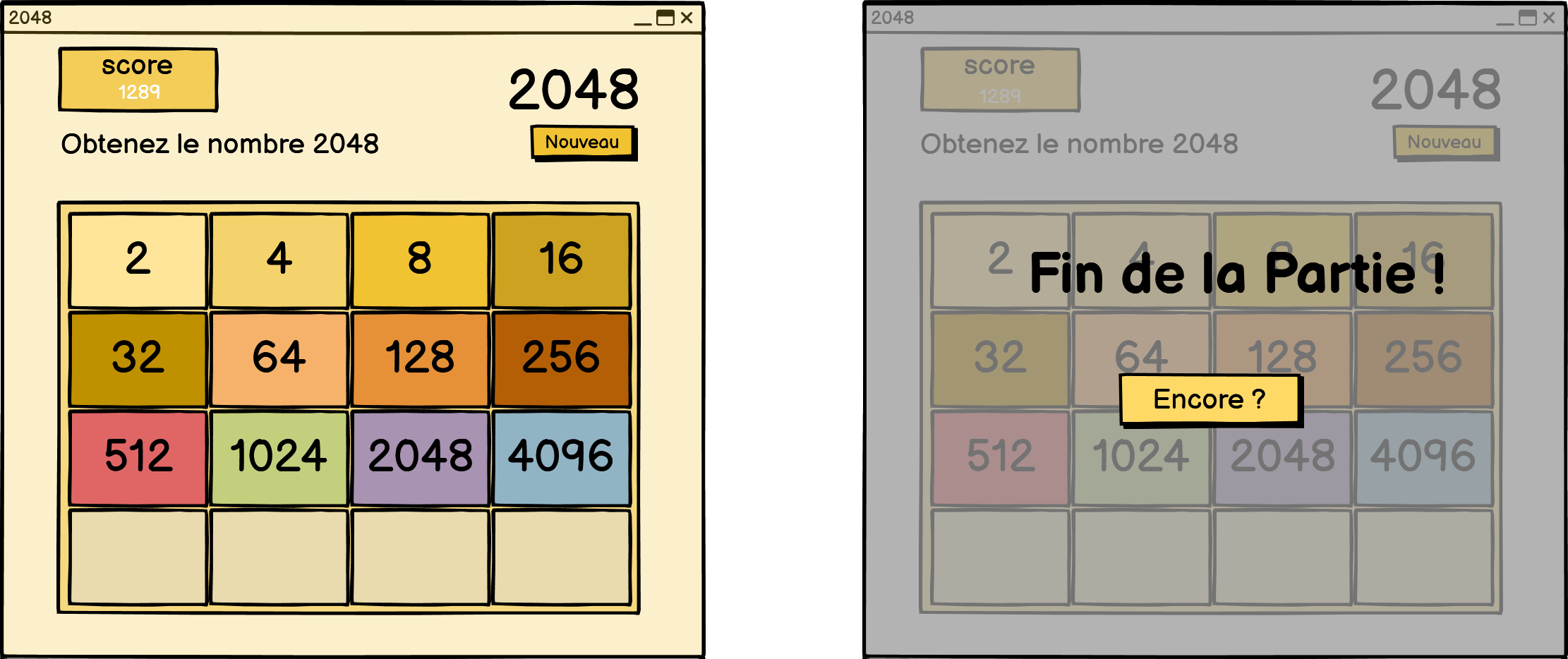
Mon objectif est d’avoir le moins possible de beug dans mon jeu pour avoir une expérience utilisateur fluide.

## Planification initiale

|  |  |  |
| --- | --- | --- |
| Sprint | Date de fin | Livrable |
| Sprint 1 | Semaine 2 :  10 février 2023 | v0.1 Etape 1 + 2  1 maquette (Balsamiq)  Code 2048\_v0.1 dans le disque perso. |
| Sprint 2 | Semaine 5 :  10 mars 2023 | v0.2 Etape 3 + 4  Code 2048\_v0.2 dans le perso  Les 2 premiers sprints dans icescrum |
| Sprint 3 | Semaine 7 :  24 mars 2023 | V0.3 Etape 5 + 6  Code 2048\_v0.3 dans un github  Les 3 sprints dans icescrum |
| Défense | Semaine 8 + 9 :  30-31 mars et 6 avril  (pas de vendredi) | Démonstration devant la classe |

# Analyse

## Maquettes



## Stories / tests d’acceptation

https://icescrum.cpnv.ch/p/JEU2048/#/project

## Stratégie de test

* Les tests on été fait la plupart du temps avec des print pour voir le fonctionnement interne des fonctions par exemple lorsque rien ne s’affichait
* Carlos a pu tester mon jeu lorsqu’il beugait et lorsqu’il marchait bien

# Implémentation

## Vue d’ensemble

Voici ce que vous avez besoin de savoir pour jouer au jeu :

Boutons :

* «Nouveau» Permet de relancer une partie n’importe quand.
* «Encore ? » Permet également de relancer une partie lorsqu’on gagne.

Touches :

* «w» Permet de déplacer en haut les tuiles.
* «a» Permet de déplacer à gauche les tuiles.
* «s» Permet de déplacer en bas les tuiles.
* «d» Permet de déplacer à droite les tuiles.

## Choix techniques

Système d’exploitations et logiciels utilisé

* Windows 11
* Pycharm, Icescrum, Balsamiq, Word, Firefox, Python 3.8

## Points techniques spécifiques

### Point 1

Fonction display()

Une image contenant texte

Description générée automatiquement

Permet d’afficher l’état de la grille à chaque mouvement.

print(numbers)

Pour la ligne de la rangée de la taille grille et pour la colonne de la rangée de la taille de la grille.

*for* line *in* range(len(numbers)):  
 *for* colon *in* range(len(numbers[line])):

Si la grille est égale à 0 donc si il n’y a rien, ajouter la couleur de base des fonds des grilles

*if* numbers[line][colon] == 0:  
 labels[line][colon].config(text="", bg="#E9DBAE")

Sinon on prend le chiffre qui est sur la grille, on lui applique sa couleur définit dans le dictionnaire qui s’appelle «colors»

*else*:  
 *# construction de chaque label sans le placer* labels[line][colon].config(text=numbers[line][colon], bg=colors[numbers[line][colon]])

### Point 2

Fonctions win()

Une image contenant texte

Description générée automatiquement

Permet d’importer des variables externes

*global* msg\_2048  
*global* fin\_label  
*global* btn\_restart\_end

Pour la ligne de la rangée de la taille grille et pour la colonne de la rangée de la taille de la grille.

*for* line *in* range(len(numbers)):  
 *for* colon *in* range(len(numbers[line])):

Si la grille est égale à 8192 : et bien on fait apparaître un texte qui dit qu’on a gagné et ajoute un bouton pour recommencer la partie.

*if* numbers[line][colon] == 8192:  
 fin\_label.place(x=40, y=400)  
 btn\_restart\_end.place(x=40, y=440)

Si la variable est égale à 1 et si la grille est égale à 2048 : on affiche le message ci-dessous et on met la variable à 0 pour que le message ne s’affiche 2 fois

*if* msg\_2048 == 1:  
 *if* numbers[line][colon] == 2048:  
 messagebox.showinfo("Félicitation", "Vous avez obtenu 2048")  
 msg\_2048 = 0

# Tests

## Tests effectués

https://icescrum.cpnv.ch/p/JEU2048/#/planning/3090/sprint/3109/story/6243/tests

## Erreurs restantes

Pas d’erreur rendue pour le sprint 2

# Conclusions

* Tous mes objectifs ont été atteint, donc avoir un jeu le plus fluide possible sans beug, avoir un petit ajout bonus que je voulais et qu’il soit agréable visuellement.
* Il avait été prévu qu’on ait 1 moi et demi pour terminer le projet et je trouve que nous avons même beaucoup de temps, donc ça laissait une grande marge de manœuvre pour tout le monde.
* Point positif : j’ai pu développer mes connaissances en python et développer ma logique en développement. Par rapport aux fonctions, l’utilisation du «global» ou des «.config».
* Pour la suite, en dehors du cours, je souhaiterai apprendre à gérer les animations des tuiles.

# Annexes

## Sources – Bibliographie

Aide de Carlos et w3school

## Journal de bord du projet

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

## Journal de travail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Jour** | **Semaine** | **Temps [h]** | **Type** | **Description** | **Remarques** |
| 03.02.2023 | - | 2h30 | Code | Optimisation de la première partie (sprint1) | - |
| 03.02.2023 | - | 1h30 | iceScrum | Création des taches pour le sprint 2 | - |
| 08.02.2023 | - | 1h | Code | Commencement du sprint 2 | - |
| 10.02.2023 | - |  |  | absent | - |
| 22.02.2023 | - | 1h | Code | Création de la fonction random | - |
| 24.02.2023 | - | 3h | Code | Compréhension de la fonction tasse 4 et concéption | - |
| 24.02.2023 | - | 1h30 | Code/aide | Reliement du bouton «Nouveau » avec la fonction random et création de la fonction display. | - |
| 01.03.2023 | - |  |  | absent | - |
| 03.03.2023 | - |  |  | absent |  |
| 08.03.2023 | - |  |  | absent |  |
| 10.03.2023 | - | 2h45 | Code/aide | Ajustement des fonctions pour chaque touche de mouvement, réglage de la fonction random et ajout sur github |  |
| 10.03.2023 | - | 30min | Icescrum | Création des tests, correction de l’ancien sprint, ajout de description et done |  |
| 10.03.2023 | - | 2h | Word | Remplire le document du projet | - |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 10.03.2023 | - | 2h | Word | Remplire le document du projet | - |
| 10.03.2023 | - | 1h30 | Icescrum | Ajout des storys et tests | - |
| 10.03.2023 | - | 20min | Python | Ajout fonctions loose | - |