Pierrick Antenen Juin 2017

Documentation Technique

Tchat’me

TPI

Contenu

[Introduction 2](#_Toc479774392)

[ Généralités sur le sujet : 2](#_Toc479774393)

[Etude d’opportunité 3](#_Toc479774394)

[ Pourquoi ce sujet : 3](#_Toc479774395)

[ Ce que mon projet à de plus : 3](#_Toc479774396)

[Analyse fonctionnelle 3](#_Toc479774397)

[ Description globale des fonctionnalités du projet : 3](#_Toc479774398)

[ Description détaillée de la liste des fonctionnalités : 3](#_Toc479774399)

[ Description de l’interface : 4](#_Toc479774400)

[Analyse organique 6](#_Toc479774402)

[ Description globale de l’architecture du projet : 6](#_Toc479774403)

[ Description des méthodes de réalisation : 6](#_Toc479774404)

[ Argumentation des éventuels choix de méthode de résolution : 6](#_Toc479774405)

[ Organigramme : 7](#_Toc479774406)

[Test et protocole de test 8](#_Toc479774407)

[ Plan de tests : 8](#_Toc479774408)

[ Rapport de test : 9](#_Toc479774409)

[Améliorations possibles 10](#_Toc479774410)

[Conclusion 10](#_Toc479774411)

[Bibliographie 10](#_Toc479774412)

# Introduction

## Généralités sur le sujet :

2048 est un jeu vidéo de type puzzle développer par Gabriele Circulli en mars 2014 et publié le 9 mars 2014.

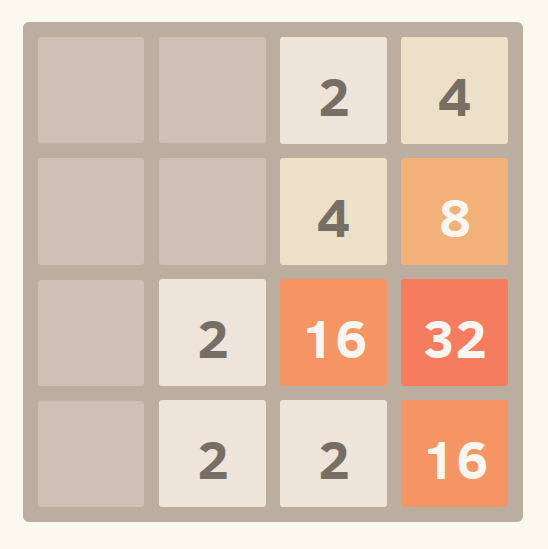
Le but du jeu est de déplacer des cases sur d’autres cases de même valeur sur une grille de 4x4 jusqu’à arriver à la case 2048.

Au début, il y a deux cases par défaut placé aléatoirement avec les chiffres 2 ou 4.

On peut déplacer les cases vers la gauche, la droite, le haut ou le bas. Lors d’un des mouvements, l’ensemble des cases sont déplacées dans la direction choisie jusqu’à rencontrer les côtés ou une autre case sur leur chemin. Si deux cases, qui ont la même valeur, se rencontrent durant le mouvement, elles fusionnent en une nouvelle case de valeur double.

Il faut qu’une case atteigne 2048 pour gagner. On peut ensuite continuer le jeu infiniment.

Le jeu officiel est disponible sur le site internet du développeur via GitHub[[1]](#footnote-1)



# Etude d’opportunité

## Pourquoi ce sujet :

Tout à commencer le jour où l’on a dû, dans le cours de Monsieur Jossy, faire un projet. J’ai longuement réfléchis et pour finir j’ai décidé de faire une petite messagerie avec base de données car j’aime beaucoup ce style d’application tels que *WhatsApp* ou encore *Messenger* de Facebook. Sur ce site, on pouvait juste envoyer des messages à un seul membre présent dans la base. Le tout était très basique et sans graphisme. Après avoir finis ce petit projet, j’ai décidé de baser mon TPI sur une messagerie de groupe. Les salles de tchat me sont très vite venues à l’esprit et tout ça aussi dans une base de données.

## Ce que mon projet à de plus :

Tchat’me permet la réunification de plusieurs membre sur un sujet particulier. Il suffit juste d’être ami pour pouvoir rejoindre une salle contrairement à d’autres sites du style dont seuls les utilisateurs qui sont dans la salle peuvent ajouter d’autres membres. Le site permet aussi

# Analyse fonctionnelle

## Description globale des fonctionnalités du projet :

Mon projet est un jeu qui se déroule sur une grille de 4x4. Des cases apparaissent aléatoirement au début et à chaque tour avec comme valeur 2 ou 4. On peut déplacer les cases à gauche, droite, en haut et en bas. Il y a un classement des scores.

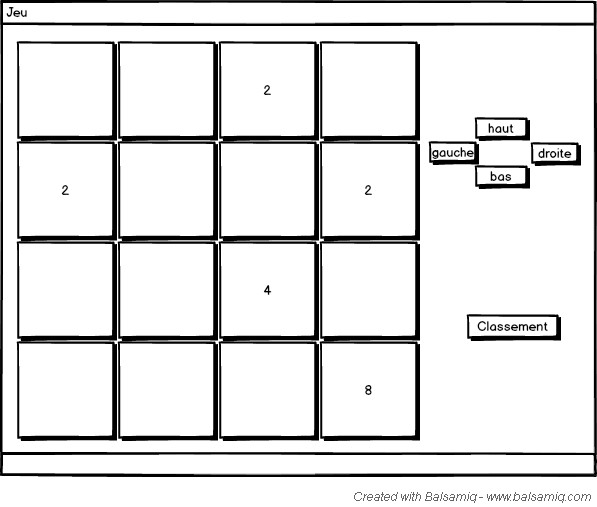
## Description détaillée de la liste des fonctionnalités :

Quand on lance l’application, une page qui nous explique les règles s’affiche. Quand on ferme cette page, le jeu s’affiche. Il y a une grille remplie de bouton en 4x4. Au lancement, il y a deux cases avec comme valeur 2 ou 4 choisies aléatoirement. On peut cliquer sur les boutons « left », « right », « top » et « down ». Ces boutons permettent de déplacer les cases selon le bouton. Si on clique sur le bouton « top » et qu’il y a aucune autre case au-dessus d’une case, alors la case va se mettre tout en haut. S’il y une autre case de la même valeur un peu plus haut, alors la case va se fusionner avec elle avec comme valeur x2 et se mettre tout en haut. Idem pour les autres boutons mais avec comparaison des lignes pour « right »et « left ». Quand on arrive avec une case à 2048, un message nous indique que l’on a gagné. On peut ensuite continuer. Quand les 16 cases sont remplies, une nouvelle page s’ouvre nous indiquant que l’on a perdu et nous demande d’inscrire un pseudo. Après l’inscription du pseudo. Une nouvelle page s’ouvre nous indiquant les meilleurs scores avec les Pseudo. On peut ensuite quitter l’application.

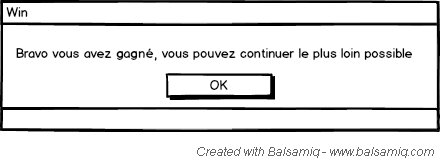
## Description de l’interface :

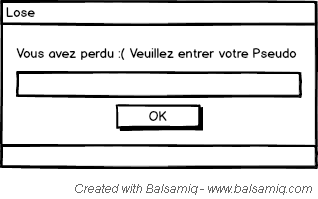
## E:\CFPT\3eme\2semestre\mercrediAprem\dossier_2048\regle.png

* Il y a un label avec le titre
* Il y a plusieurs label où sont écris les règles
* Il y a un bouton pour fermer cette Form

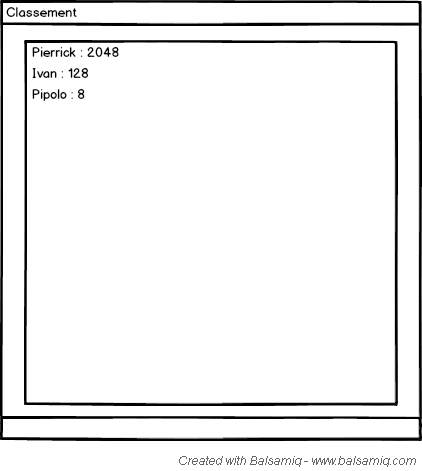


* Il y a Panel qui contient les cases.
* Il y a plusieurs boutons qui sont les cases (4x4).
* Il y a les boutons qui permettent de bouger les cases à gauche, à droite, en haut et en bas.
* Il y a un bouton qui permet d’afficher le classement.





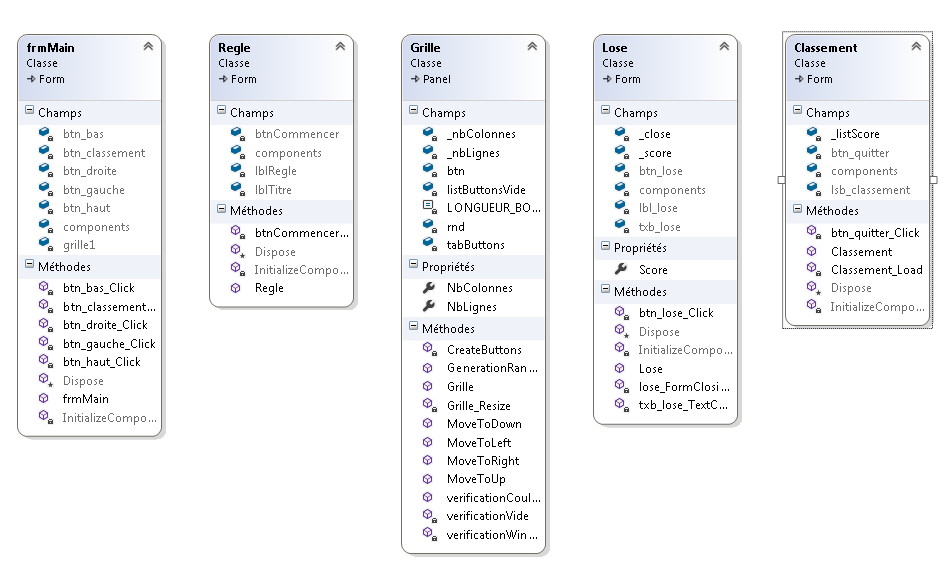
* Il y a un label pour indiquer la défaite
* Il y a un textBox pour que l’utilisateur puisse entrer le Pseudo
* Il y a un bouton qui permet de confirmer le pseudo



* Il y a une listeBox pour afficher le classe ment

# Analyse organique

## Description globale de l’architecture du projet :



Pour mon jeu principal. J’ai décidé de séparer en deux classes. La première étant la vue sous le nom de frmMain qui est relié à la classe  « Grille » qui elle, est le modèle qui va tout gérer. Il y a la classe Règle qui affiche juste les règles. Il y a aussi la classe « Lose » qui va permettre d’afficher du texte, d’entrer un pseudo et de faire le traitement pour écrire dans un fichier texte. Il y a pour finir la classe « Classement » qui elle, va permettre de lire le fichier texte et de l’afficher.

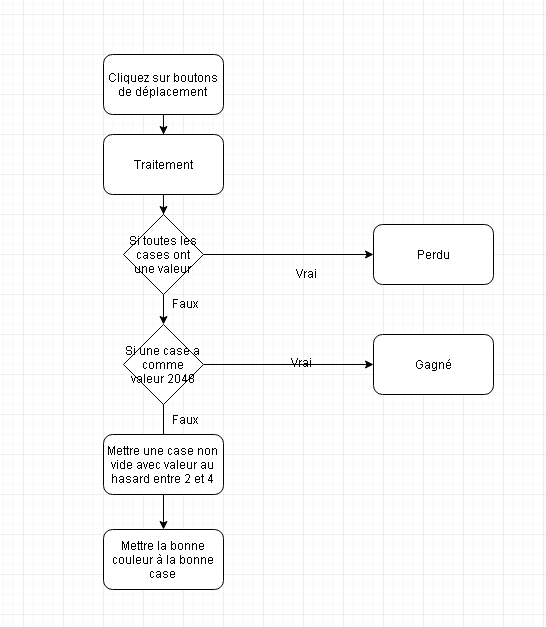
## Description des méthodes de réalisation :

Au lancement du jeu, un panel va se créer avec un tableau à double entré de bouton à l’intérieur. Il y a donc 16 boutons qui vont se créer dans le panel (4x4). Il y a ensuite une fonction qui permet de vérifier quel bouton n’a aucune valeur et va en faire une liste de boutons vide. Une autre fonction va alors placer au hasard en 2 et 4 une valeur dans un bouton vide. Quand on clique sur un des quatre boutons de déplacement, ça va appeler une des fonctions de déplacement qui va alors vérifier si d’autres cases se trouvent aux alentours et va alors faire le traitement adéquat. À chaque fin de traitement, la fonction pour placer au hasard de nouvelle case va alors être appelée. Il y a une fonction qui va vérifier à chaque tour si l’utilisateur gagne ou perd et ce qui permet d’ouvrir la Form « Lose » ou le messageBox par rapport aux situations. La Form « Lose » va permettre d’écrire dans un fichier txt le pseudo avec le score. Quand on clique sur le bouton « Classement local ». Ça va ouvrir la Form « Classement ». Cette form permet de lire un fichier txt. Elle va placer le contenu dans une liste et faire un Sort pour trier par rapport au meilleur score.

## Argumentation des éventuels choix de méthode de résolution :

Au début, j’avais décidé de placer à la main le panel et les boutons mais je me suis très vite rendu compte que c’était une mauvaise idée pour effectuer un traitement dessus. J’ai donc créé une classe qui permet de faire le traitement.

## Organigramme :



# Test et protocole de test

## Plan de tests :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N°** | **Date** | **Auteur** | **scénario** | **Résultat attendu** |
| T1 | 29.03.17 | Pierrick | Lancer l’application | La Form « Règle » devrait être ouverte. |
| T2 | 29.03.17 | Pierrick | Lancer l’application  Cliquer sur commencer | Le jeu devrait démarrer avec deux cases au hasard rempli de valeur 2 ou 4. |
| T3 | 29.03.17 | Pierrick | Lancer l’application  Cliquer sur commencer  Cliquer sur le bouton « left » | S’il y a aucune case à gauche d’une case, la case va se déplacer tout à gauche.  S’il y a une case de la même valeur sur la ligne, les deux cases vont en former qu’une en faisant x2 sur la valeur et en se déplaçant à gauche.  Il devrait avoir un 2 ou 4 qui apparait aléatoirement dans une case sans valeur à chaque appuie de ce bouton |
| T4 | 29.03.17 | Pierrick | Faire la même chose que le T3 mais pour « right », « top » et « down » | Le même resultat que pour T3 mais avec droite, haut et bas |
| T5 | 29.03.17 | Pierrick | Lancer l’application  Cliquer sur commencer  Pendant la partie, cliquer sur le bouton « Classement local » | La Form « Classement » devrait s’ouvrir en affichant dans un ordre décroissant (du plus haut score au plus bas) les scores avec le Pseudo. |
| T6 | 29.03.17 | Pierrick | Lancer l’application  Cliquer sur commencer  Arriver avec une case de 2048 | Un message devrait apparaître en indiquant à l’utilisateur qu’il a gagné. Il devrait pouvoir continuer de jouer. |
| T7 | 29.03.17 | Pierrick | Lancer l’application  Cliquer sur commencer  Faire que toutes les cases ont une valeur | La Form « Lose » devrait apparaître en demandant le pseudo de l’utilisateur |
| T8 | 29.03.17 | Pierrick | Lancer l’application  Cliquer sur commencer  Faire que toutes les cases ont une valeur  Mettre un pseudo et cliquer sur « confirmer » | La Form devrait se fermer et ça devrait afficher la Form « Classement » |
| T9 | 29.03.17 | Pierrick | Lancer l’application  Cliquer sur commencer  Faire que toutes les cases ont une valeur  Mettre un pseudo et cliquer sur « confirmer »  Cliquer sur « Quitter » | Toute l’application en entière devrait se fermer |

## Rapport de test :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N°** | **Date** | **Testeur** | **Résultat obtenu** | **OK/KO ?** |
| T1 | 12.04.17 | Pierrick | La Form « Regle » s’affiche correctement | OK |
| T2 | 12.04.17 | Pierrick | Le Jeu se lance correctement | OK |
| T3 | 12.04.17 | Pierrick | La case va correctement se déplacer tout à gauche s’il y a aucune case a gauche. S’il y a une case de la même valeur sur la ligne, ils vont correctement en faire qu’une en multipliant la valeur x2. Il y a bien des cases qui vont apparaître aléatoirement mais des fois les cases aléatoires se mettent à la place de certaines cases déjà occupés. | KO |
| T4 | 12.04.17 | Pierrick | Même résultat que le T3 | KO |
| T5 | 12.04.17 | Pierrick | La Form « Classement » s’ouvre correctement mais n’affiche pas dans un ordre décroissant les scores avec le pseudo (affiche correctement jusqu’à 128) | KO |
| T6 | 12.04.17 | Pierrick | Le message s’affiche correctement et l’utilisateur peut continuer à jouer | OK |
| T7 | 12.04.17 | Pierrick | La Form « Lose » s’affiche correctement et l’utilisateur peut entrer son Pseudo. Il ne peut pas continuer tant qu’il n’a pas entré un Pseudo | OK |
| T8 | 12.04.17 | Pierrick | La Form « Classement » s’affiche correctement | OK |
| T9 | 12.04.17 | Pierrick | Toute l’application se ferme | OK |

# Améliorations possibles

Je pense que je pourrais améliorer déjà le système de classement local. J’utilise un fichier txt pour stocker le classement ce qui n’est pas très pratique. Je pourrais faire une base de données pour permettre de faire un classement pour tous au lieu de local. Je pourrais utiliser les flèches du clavier au lieu des boutons, ce qui serait plus pratique et plus ergonomique. Je pourrai améliorer le système de la mise de case aléatoirement car il y a certains bugs qui font apparaître des 2 ou des 4 dans des cases qui contiennent d’autres valeurs.

# Conclusion

Au début, j’ai eu beaucoup de problèmes à commencer à coder. Je n’arrivais pas à créer de façon automatique les boutons et je ne voyais pas du tout comment faire pour les déplacer non plus. Après une demande d’aide à Monsieur Durrenmatt, puis au prof, j’ai réussi à avancer dans mon projet. Malheureusement, Je n’avais plus beaucoup de temps après tous ses cours sans avancement. Pour conclure, je suis content de ce que j’ai réussi à faire avec ce temps. Ce projet m’a permis d’améliorer mes connaissances en C# et surtout en programmation orientée objets. J’ai aussi appris qu’il ne faut pas attendre quand on est coincé et pas hésité à demander de l’aide car ça m’a beaucoup aidé et m’a permis de déstresser.

# Bibliographie

<https://v4-alpha.getbootstrap.com/components/forms/> Consulté le 07/07/2017

<https://v4-alpha.getbootstrap.com/components/buttons/> Consulté le 07/07/2017

<https://www.alsacreations.com/tuto/lire/1032-comment-centrer-verticalement-sur-tous-les-navigateurs.html/> Consulté le 07/07/2017

1. https://gabrielecirulli.github.io/2048/ [↑](#footnote-ref-1)