25/01/2019

Cyril Goldenschue & Ilan Ruiz-De-Porras

CPNV

Projet C#

Table des matières

[1 Introduction 2](#_Toc536186887)

[1.1 Cadre et Description 2](#_Toc536186888)

[1.2 Organisation 2](#_Toc536186889)

[2 Analyse 2](#_Toc536186890)

[2.1 Cahier des charges détaillé 2](#_Toc536186891)

[2.2 Stratégie de test 2](#_Toc536186892)

[2.2.1 Test Unitaire 3](#_Toc536186893)

[2.2.2 Test Performance 3](#_Toc536186894)

[2.2.3 Test de Validation 3](#_Toc536186895)

[2.3 Planification 3](#_Toc536186896)

[2.4 Explication des jeux 3](#_Toc536186897)

[2.4.1 Bataille 3](#_Toc536186898)

[2.4.2 Solitaire 3](#_Toc536186899)

[2.4.3 Morpion / Tic-Tac-Toe 4](#_Toc536186900)

[2.5 Analyse concurrentiel 4](#_Toc536186901)

[2.5.1 Tic-tac-toe 4](#_Toc536186902)

[2.5.2 Le solitaire 5](#_Toc536186903)

[2.5.3 La bataille 5](#_Toc536186904)

[3 Conception 6](#_Toc536186905)

[3.1 Dossier de Conception 6](#_Toc536186906)

[3.2 Maquettes graphiques 6](#_Toc536186907)

[3.3 MCD et MLD 6](#_Toc536186908)

[3.4 Diagramme de classe 6](#_Toc536186909)

[3.5 Réalisation 6](#_Toc536186910)

[3.5.1 Morpion 6](#_Toc536186911)

[3.5.2 Bataille 7](#_Toc536186912)

[3.5.3 Solitaire 8](#_Toc536186913)

[4 Conclusion 9](#_Toc536186914)

[4.1 Objectif attends 9](#_Toc536186915)

[4.2 Objectif non-attends 9](#_Toc536186916)

# 1 Introduction

## 1.1 Cadre et Description

Ce projet consiste à développer une application en C#. Le sujet est totalement libre.

Notre application est un regroupement de mini-jeux. De base l’application contiendra trois jeux ; La bataille, le morpion et le solitaire. L’objectif est d’intégrer ces trois jeux sous forme de formulaires et si possible implémenter de nouvelles fonctionnalités et jeux.

## 1.2 Organisation

|  |  |  |
| --- | --- | --- |
| Nom | Rôle | Mail |
| Ilan Ruiz De Porras | Élève | Ilan.ruiz-de-porras@cpnv.ch |
| Cyril Goldenschue | élève | Cyrile.Goldenschue@cpnv.ch |
| Julien Ithurbide | Chef de projet | Julien.Ithurbide@cpnv.ch |
| Frederique Andolfatto | Expert | Frederique.Andolfatto@cpnv.ch |

# 2 Analyse

## 2.1 Cahier des charges détaillé

1. Nous devons travailler en binôme et choisir le sujet de l’application.
2. L’application devra introduire une notion de programmation avancé tel qu’une connexion a une base de donnée ou pouvoir communiquer sur le réseau
3. Implémentation de l’analyse et conception de l’application : Après validation du chef de projet, rédiger un documents (markdown, docx, …) contenant un chapitre analyse et conception. Il faut prendre le template de **documentation pour application** comme template de base.
4. Implémentation de l’application : Tout en implémentant, rédiger les chapitres nécessaires à cette étape.
5. Phase de validation : Dernière phase du projet qui consiste à livrer le produit final au chef de projet. Le rendu comportera :

* L’application (tous les fichiers)
* Documentation final contenant :
* Toutes les phase d’analyse, de conception et d’Implémentation de l’application

## 2.2 Stratégie de test

Nous ferons trois type de testes : - Test unitaire - Test de performance - Test de validation.

### 2.2.1 Test Unitaire

Ces tests seront les plus fréquents et ne sont pas effectué par les testeurs. Il s’agit de testes qui déterminerons si tel et tel chose fonctionne correctement.

Exemple : tester le traitement de string, Correction du code, etc…

### 2.2.2 Test Performance

Ces tests sont effectués afin de déterminer les performances nécessaires afin de faire tourner notre application.

Ils seront simplement effectué à l’aide de l’interface de Visual studio et la vitesse d’exécution de l’application.

Ces tests montrent également si l’application est optimisée ou non.

### 2.2.3 Test de Validation

Ces tests permettront de valider une fonctionnalité de l’application. Ces tests seront effectués à l’aide d’une liste de testes à effectuer pour valider la fonctionnalité.

## 2.3 Planification

La planification se trouve sur GitHub, dans le répertoire du projet :

[Lien du répertoire du projet](https://github.com/ICMiniGames/MiniGames)

## 2.4 Explication des jeux

### 2.4.1 Bataille

La bataille est un jeu de carte relativement basique se jouant à deux joueur.

Les règles sont assez simple. On divise un paquet de 52 cartes en deux et on distribue la moitié au deux joueur. Chacun va poser la carte se trouvant au-dessus du tas et le poser face visible. Le but est que votre carte soit d’une valeur supérieur à celle de votre adversaire. Celui qui aura obtenu le plus grand score gagne.

Pour plus d’information : https://fr.wikipedia.org/wiki/Bataille\_(jeu)

### 2.4.2 Solitaire

Le solitaire est un jeu de carte que on retrouve sur de nombreux navigateur et il est également de base dans le système d’exploitation Windows.

Le but est de formé ce qui s’appelle des familles. Une famille est composée de toutes les cartes allant de l’as au roi du même symbole. Il s’agit d’un jeu en continue et le but est d’avoir formé toutes les familles de tous les symboles.

Pour plus d’information : https://fr.wikipedia.org/wiki/Solitaire\_(patience)

### 2.4.3 Morpion / Tic-Tac-Toe

Le morpion aussi appelé le Tic-Tac-Toe, est un jeu très basique se jouant à deux.

Le but est de formé une ligne grâce au placement de symbole dans un plateau. Le jeu se joue au tour par tour. Chaque joueur placera son symbole dans une des cases du plateau et ainsi de suite jusqu’à ce qu’une ligne se forme ou qu’il n’y a aucune.

Pour plus d’information : https://fr.wikipedia.org/wiki/Tic-tac-toe

## 2.5 Analyse concurrentiel

La plupart des jeux proposé par notre application se trouve sur navigateur web tel que **Chrome**. La bataille est le seul jeu de la liste n’y figurant pas.

### 2.5.1 Tic-tac-toe

Il existe un tic-tac-toe sur Google. Il suffit simplement de taper **tic-tac-toe** dans la barre de recherche Google pour y accéder. Il se présente sous cette forme :

Le gros avantage de cette version du tic-tac-toe est qu’il est accessible directement via navigateur. L’accès est donc direct et il n’y a pas besoin de téléchargement.

La partie graphique est basic mais pour un jeu aussi simple que celui-ci on n’a pas besoin d’un désigne complexe.

Une des fonctionnalités intéressantes de cette version c’est la possibilité de jouer contre une intelligence artificielle. Il est même possible de choisir le niveau de difficulté. La possibilité de jouer contre un autre joueur est aussi possible. Par contre on ne peut pas jouer d’une machine à l’autre.

#### 2.5.1.1 Conclusion Tic-tac-toe

|  |  |
| --- | --- |
| Avantage | Inconvénient |
| possibilité de jouer contre un bot | besoin d’une connexion à internet pour y jouer |
| plusieurs niveau de difficulté | pas de multijoueur sur réseau |
| Accès direct par navigateur | - |
| Multijoueur local | - |

Les fonctionnalités que cette version ne possède pas vont être implémenter dans notre version.

### 2.5.2 Le solitaire

Tout comme le Tic-tac-Toe, le solitaire est disponible via le moteur de recherche Google et aussi via les jeux de Windows. Ici nous analysons la version de Google.

Pour commencer, il est accessible directement via Google donc pas besoin de téléchargement pour pouvoir y jouer. Il s’agit d’une fenêtre pop-up qui sert d’interface de jeu.

Le style des carte et l’interface est graphiquement épuré. C’est fluide et on a la possibilité de déplacer les cartes. Ce n’est pas Just “je clique et ça apparait ou j’ai cliqué”.

Le hud (Head up display) est simple et affiche le score, le nombre de déplacement et le temps que vous mettez pour finir le jeu.

Il est également possible de choisir sa difficulté. On a le choix entre facile et difficile. La différence entre les deux c’est que lorsque on pioche une carte en difficile, on en tire trois et non une.

#### 2.5.2.1 Conclusion du solitaire

|  |  |
| --- | --- |
| Avantage | Inconvénient |
| Choix de difficulté | une connexion à internet est requis |
| accès direct via navigateur | - |
| beau graphiquement | - |
| hud et options bien pensé | - |

Nous nous inspirons du hud de cette version et nous aurons la possibilité de pouvoir jouer en hors-ligne grâce à notre version.

### 2.5.3 La bataille

Nous n’avons pas trouvé d’application de jeu sur la bataille. En même temps il s’agit d’un jeu très simpliste et le seul facteur possible pour gagner est la chance.

Nous introduirons un système de paris qui n’existe pas dans le jeu de base. Nous resterons simples dans l’interface graphique et nous respecterons les règles de cette version du jeu.

# 3 Conception

## 3.1 Dossier de Conception

## 3.2 Maquettes graphiques

Menu de l’application :

Interface du morpion :

Interface de la bataille :

Interface du solitaire :

## 3.3 MCD et MLD

MLD :

## 3.4 Diagramme de classe

## 3.5 Réalisation

### 3.5.1 Morpion

#### 3.5.1.1 Use case et Scénarios.

Action de base :

|  |  |
| --- | --- |
| Action | Résultat |
| Le joueur 1 clique sur une case | - La case est remplacé par le symbole du joueur 1 **-** Au tour du deuxième joueur |
| Le joueur 2 clique sur une case | - La case est remplacé par le symbole du joueur 2 **-** Au tour du joueur 1 |
| Le joueur 1 clique sur une case déjà prise par le joueur 2 | - Rien ne se passe **-** Le tour du joueur 1 continue tant qu’il n’a pas cliquer sur une case libre |
| Le joueur 2 clique sur une case déjà prise par le joueur 1 | - Rien ne se passe **-** Le tour du joueur 2 continue tant qu’il n’a pas cliquer sur une case libre |
| Le joueur 1 clique sur une case déjà prise par lui-même | - Rien ne se passe **-** Le tour du joueur 1 continue tant qu’il n’a pas cliquer sur une case libre |
| Le joueur 2 clique sur une case déjà prise par lui-même | - Rien ne se passe **-** Le tour du joueur 2 continue tant qu’il n’a pas cliquer sur une case libre |

Victoire :

|  |  |
| --- | --- |
| Action | Résultat |
| Un des deux joueurs forme une ligne avec ses symboles | Le jeu s’arrête et annonce qui est le gagnant **-** Le jeu propose au joueur si il veut rejouer ou quitter. |
| Tous les case ont été remplie mais pas de gagnant | Le jeu annonce qu’il y a égalité et propose de rejouer ou quitter |
| Un joueur clique sur Continuer | Le jeu recommence |
| Un joueur clique sur Quitter | Le jeu s’arrête et retour au menu |

#### 3.5.1.2 Listes des tests.

|  |  |  |  |
| --- | --- | --- | --- |
| Action | Résultat | Description supplémentaire |  |
| Les joueurs ne peuvent pas sélectionné une case déjà prise | - | - |  |
| Le jeu se termine quand les trois symboles sont alignés (Dans toutes les directions possible) | - | - |  |
| Le jeu se termine quand il y a match nul (Aucune ligne formée par les symboles) | - | - |  |
| Le menu de fin de partie s’affiche | - | - |  |
| Le choix continue remet à zéro le plateau | - | - |  |
| Le joueur gagnant obtient un point à son score | - | - |  |
| Le choix quitter le jeu ferme la fenêtre du morpion | - | - |  |
| Le nom des joueurs est affiché vers les bords supérieur de la fenêtre | - | - |  |

### 3.5.2 Bataille

#### 3.5.2.1 Use case et Scénarios.

|  |  |
| --- | --- |
| Action | Résultat |
| Les joueurs se sont enregistré | La fenêtre de pari s’ouvre |
| Les joueurs parient sur un bot | La partie se lance, avec l’affichage du jeu |
| Un joueur à gagné | Une fenêtre s’ouvre et indique quelle joueur à gagné |

#### 3.5.2.2 Listes des tests.

|  |  |  |  |
| --- | --- | --- | --- |
| Action | Résultat | Description supplémentaire |  |
| Le joueur choisi un des bots | Possibilité de lancer la partie | - |  |
| le bot choisi à gagne la manche | le joueur gagne un point | - |  |
| le bot choisi perd la manche | le joueur ne gagne pas de point | - |  |
| le joueur a cinq points | le joueur gagne la partie | - |  |
| un joueur gagne | possibilité de recommencer ou d’arrêter | - |  |

### 3.5.3 Solitaire

#### 3.5.3.1 Use case et Scénarios.

|  |  |
| --- | --- |
| Action | Résultat |
| Le joueur s’est enregistré | La partie se lance, affichant le plateau de jeu |
| Le plateau s’est affiché | Les cartes sont retourné et caché selon les règle du jeu |
| Le joueur prend une carte en maintenant le clic droit | la carte se déplace avec les curseur |
| Le joueur “lâche” la carte au-dessus de rien (plateau en vert) | La carte retourne à sa position d’origine |
| le joueur “lâche” la carte sur une autre carte | - |
| Si la carte est posé est de couleur différent et de valeur inférieur | La carte sa position au-dessus |
| - | Sinon la carte retourne à sa position d’origine |
| Si une carte a été posse | La carte face caché qui se trouvait en dessous de celle si se retourne |
| Le joueur “lâche” une carte sur l’emplacement de carte | La carte se positionne dessus |
| Si toutes les cartes ont été réunis | La partie se termine |

#### 3.5.3.2 Listes des tests.

|  |  |  |  |
| --- | --- | --- | --- |
| Action | Résultat | Description supplémentaire |  |
| Le joueur maintien le clique gauche sur la carte | La carte se déplace avec le curseur | - |  |
| Le joueur lâche la carte sur le plateau | Elle retourne à sa position d’origine | - |  |
| Le joueur lâche une carte sur une autre | Vérifie qu’elle soit de couleur différent et de valeur inférieur | Si oui elle se position dessus, Sinon elle retourne à sa position d’origine |  |
| Si il y a une carte sous celle qui a été déplacé | Elle se retourne si la carte est posé sur une autre | - |  |

# 4 Conclusion

## 4.1 Objectif attends

Tous les objectifs présents dans le cahier des charges ont été attends. La majeure partie de nos objectifs ont également été attends.

## 4.2 Objectif non-attends

Nous avions comme objectif de pouvoir jouer en réseau mais malheureusement les temps que nous avions à disposition n’a pas suffi. La cause en est que nous avons perdu du temps sur des détails sans grande importance et donc nous avons concentrés nos efforts dans quelques points que nous aurions dû régler à la fin du projet