|  |  |  |
| --- | --- | --- |
| Atelier  Projet | Second semestre  2017 | |
| Documentation technique du  « EditHexa » | | Groupe:  Dürrenmatt Cédric,  Cugni Zoé,  Oliveira Ricardo |
|  |  |  |

# 

# 

# 

[**Description du projet**](#_4upaikoqt81z) **3**

[**Analyse fonctionnelle**](#_v0665v8kll1n) **3**

[Description globale des fonctionnalités du projet](#_14kbsn86knfy) 3

[Description détaillée de la liste des fonctionnalités](#_7d71tdtnkwki) 3

[Description de l’interface (Balsamiq)](#_bn6quqeu3w12) 4

[**Analyse organique**](#_lh5f6dky2isf) **5**

[Description globale de l’architecture du projet](#_1y0wt0y9qdgh) 5

[Description des méthodes de réalisation](#_ucyyqik83ggf) 5

[Conclusion](#_6e8vzwtseauj) 6

# 

# 

# **Description du projet**

Le but de ce projet est de reproduire un programme existant, EditHexa, permettant d’afficher en hexadécimal (ainsi qu’en ascii) les octets d’un fichier, d’avoir des informations détaillées dessus et de les modifier.

# **Analyse fonctionnelle**

### *Description globale des fonctionnalités du projet*

Notre logiciel permet de visualiser et changer le code hexadécimal d’un fichier.

### *Description détaillée de la liste des fonctionnalités*

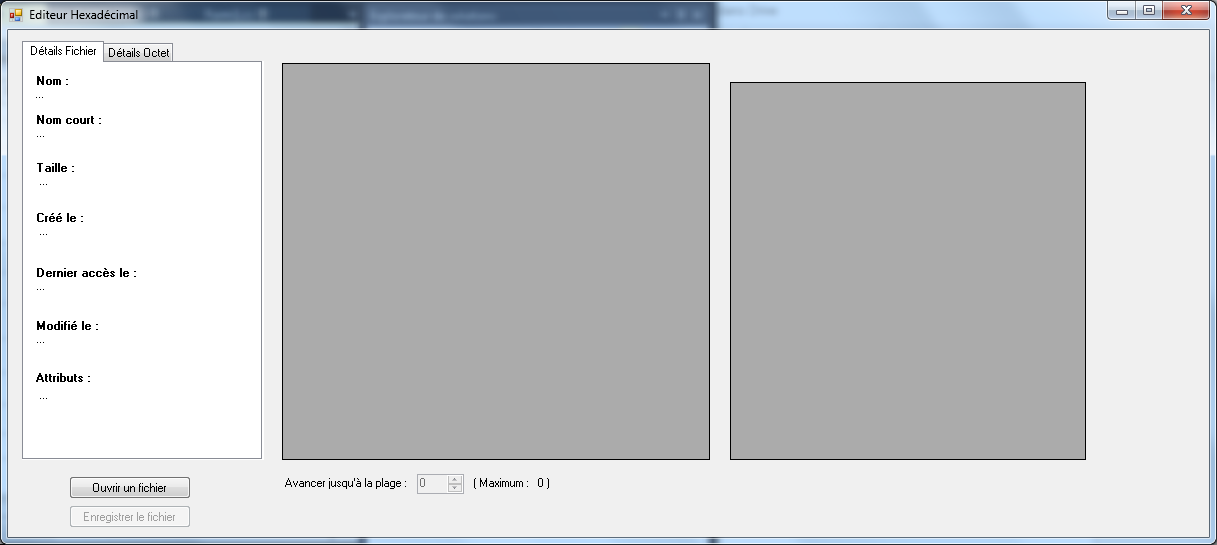
Notre projet a les fonctionnalités suivantes:

* Afficher le code hexadécimal et ascii d’un fichier
* Modifier le code hexadécimal et ascii d’un fichier
* Sauvegarder le fichier modifié
* Annuler la modification d’une cellule (avec la touche DELETE ou BackSpace)
* Afficher les métadatas (Nom, Nom DOS, Taille, date de création,...) du fichier
* Faire des conversions de hexadécimal à:
  + Ascii
  + Binaire
  + Octal
  + 8 bits (signé)
  + 8 bits (non-signé)
  + 16 bits (signé)
  + 16 bits (non-signé)
  + 32 bits (signé)
  + 32 bits (non-signé)
  + 64 bits (signé)
  + Float
  + Double

### Description de l’interface (Balsamiq)

Il contient:

* 1x TabControl + 2x TabPage
* 2x Button
* 2x GridView
* 3x Label
* 1x NumericUpDown
* 1x OpenFileDialog
* 1x SaveFileDialog



# 

# **Analyse organique**

## *Description globale de l’architecture du projet*

## Ce programme suit la philosophie Modèle-Vue, il est donc composé d’un modèle (une classe), HexaEditModel, et d’une vue, Form1.

HexaEditModel contient aussi une autre classe FileData, permettant de gérer les metadata d’un fichier.

Le model travaille principalement avec des DataTable, et la Form avec des DataGridView.

## *Description des méthodes de réalisation*

**Modèle :**

Le modèle génère des types de données sous trois formats:

* Ascii
* Hexadécimal
* Bytes (donné par le File.ReadAllBytes)

De ces trois formats, il gère la bonne entente entre le modèle et la vue (modification, exportation, récupération, conversion).

* Le modèle convertit une donnée de la DataTable hexadecimal (représenté par le Point (y,x) reçu en paramètre) grâce aux “Convert” et “BitConverter”.
* Le modèle gère le système de changement (effectuer un changement ou l’annuler).
* Les changements sont automatiquement affecté aux trois formats.
* L’offset, affiché sur la gauche du composant TabControl, est géré pendant la création de la table à 2 dimensions « Hexadécimal ».

**Affichage :**

* L’intérieur des deux DataGridView est remplis dynamiquement en demandant au modèle les données plage par plage.
* La plage demandée est mise à jour soit lors d’un scroll de souris, soit lors d’un changement de valeur du Numeric Up Down
* Les GridViews n’ayant pas le même nombre de colonne (l'hexadécimal en a 1 de plus), toute les méthodes jonglent avec un +1/-1 en fonction des situations.
* Quand une cellule est sélectionnée dans une GridView, sa cellule équivalente est automatiquement sélectionnée dans l’autre GridView.
* Le redimensionnement est effectué programmatiquement et non de façon automatique en utilisant les variables des GridView car celles-ci ont facilement des comportements étranges. Pour restreindre un peu le redimensionnement et empêcher le programme de devenir illisible, de nombreuses tailles minimum/maximum ont été définies.
* Une liste est utilisée pour garder en mémoire les cellules à afficher en rouge après un retour sur une plage précédemment modifiée.
* Les autres fonctionnalités de la Form n’ont pas de spécificités particulières et ne sont donc pas décrite ici mais sont amplement documentées dans le code.

## **Conclusion**

Bien qu’il ne suive pas entièrement son exemple, le programme est fonctionnel et atteint son but.