2018

Carbonara Christian – CIN4B

ETML

17/05/2018

Convertisseur de bases pour des nombres entiers et réels, codés sur 32 bits

|  |
| --- |
| Table des matières |

[1. Remerciements 3](#_Toc514333491)

[2. Spécifications 3](#_Toc514333492)

[2.1. Titre 3](#_Toc514333493)

[2.2. Description 3](#_Toc514333494)

[2.3. Matériel et logiciels à disposition 4](#_Toc514333495)

[2.4. Prérequis 4](#_Toc514333496)

[2.5. Cahier des charges 4](#_Toc514333497)

[2.5.1. Objectifs et portée du projet (objectifs FFOR) 4](#_Toc514333498)

[2.5.2. Caractéristiques des utilisateurs et impacts 4](#_Toc514333499)

[2.5.3. Fonctionnalités requises (du point de vue de l’utilisateur) 4](#_Toc514333500)

[2.5.4. Contraintes 4](#_Toc514333501)

[2.5.5. Travail à réaliser par l’apprenti 5](#_Toc514333502)

[2.5.6. Si le temps le permet 5](#_Toc514333503)

[2.6. Les points suivants seront évalués 5](#_Toc514333504)

[2.7. Validation et conditions de réussite 6](#_Toc514333505)

[3. Planification initiale 6](#_Toc514333506)

[4. Analyse 9](#_Toc514333507)

[4.1. Opportunités 9](#_Toc514333508)

[4.2. Document d’analyse et conception 9](#_Toc514333509)

[4.2.1. Logiciel utilisé pour coder 9](#_Toc514333510)

[4.2.2. Maquette graphique 9](#_Toc514333511)

[4.2.3. Conception du programme 9](#_Toc514333512)

[4.3. Conception des tests 10](#_Toc514333513)

[4.4. Planification détaillée 10](#_Toc514333514)

[5. Réalisation 14](#_Toc514333515)

[5.1. Dossier de réalisation 14](#_Toc514333516)

[5.1.1. Logiciels installé / Utilisés 14](#_Toc514333517)

[5.2. Modifications 14](#_Toc514333518)

[6. Tests 14](#_Toc514333519)

[6.1. Dossier des tests 14](#_Toc514333520)

[7. Conclusion 14](#_Toc514333521)

[7.1. Bilan des fonctionnalités demandées 14](#_Toc514333522)

[7.2. Bilan de la planification 14](#_Toc514333523)

[7.3. Bilan personnel 15](#_Toc514333524)

[8. Divers 15](#_Toc514333525)

[8.1. Journal de travail 15](#_Toc514333526)

[8.2. Bibliographie 15](#_Toc514333527)

[8.3. Webographie 15](#_Toc514333528)

[9. Annexes 15](#_Toc514333529)

# Remerciements

# Spécifications

## Titre

Convertisseur de bases (binaire, octal, décimal et hexadécimal) pour des nombres entiers et réels (à virgules fixes), codés sur 32 bits

## Description

Il s'agit d'implémenter une application, en C#, qui va permettre aux utilisateurs (des élèves et des enseignants), de saisir une base numérique (bases binaires, octales, décimales, hexadécimales, nombres à virgules fixes), un nombre et le programme affiche le même nombre dans les autres bases. De plus, le programme doit permettre d'accomplir des opérations élémentaires d'additions et soustractions sur des nombres entiers positifs et négatifs.

Le dispositif indiquera les détails des opérations, et se fera jusqu'à 32 bits signés pour les nombres binaires. En aucun cas l'utilisation de fonctions prédéfinies du langage ne sera acceptée. Les élèves doivent parfois résoudre des exercices, ou des situations, qui nécessitent des conversions de nombres d'une base à une autre. Ou encore, faire des additions ou des soustractions.

Par exemple, comment convertir un nombre réel à virgule fixe de décimal en hexadécimal (ou 134.4510 -> ?.?16).

Actuellement, à l'ETML, en informatique, un module nommé ELEOC-NUM pour électronique numérique, traite ce sujet. Les supports présentent de la théorie et des exercices. Toutefois, il conviendrait de pouvoir laisser les élèves résoudre des mêmes exercices, mais avec des données différentes, et de pouvoir vérifier leurs réponses de manière automatique.

L'idée est d'avoir une application qui permette aux élèves de s'entrainer pour les fonctionnalités suivantes, avec des mots de 32 bits :

* Conversions de nombres décimaux, binaires, octaux et hexadécimaux (entiers et/ou réels à virgules fixes) en nombres dans les 3 autres bases que la base courante.
* Additions et soustractions de 2 nombres binaires, octaux et hexadécimaux (entiers, positifs et/ou négatifs)
* Conversion d'un nombre réel à virgule fixe, décimal, positif ou négatif, en nombre binaire à virgule fixe et réciproquement.

Le candidat devra d'abord analyser les domaines d'applications du sujet pour chaque base considérée, afin que le tout soit cohérent. Par exemple, avec 32 bits binaires signé, quel minima et quel maxima peut-on utiliser pour convertir en hexadécimal ?

## Matériel et logiciels à disposition

1 ordinateur standard ETML, avec la structure habituelle.

## Prérequis

Avoir suivi les modules ELEOC, et de programmation en C#.

## Cahier des charges

### Objectifs et portée du projet (objectifs FFOR)

**F**orce :

**F**aiblesse :

**O**pportunité :

**R**isque :

### Caractéristiques des utilisateurs et impacts

Les utilisateurs sont en 1ère année de l’ETML ainsi que très probablement des utilisateurs d’autres années, certains n’ont pas de connaissances poussées de l’informatique, il faut donc que le programme soit simple d’utilisation et intuitif pour tous.

### Fonctionnalités requises (du point de vue de l’utilisateur)

Conversions des nombres décimaux, binaires, octaux et hexadécimaux (entier et/ou réels à virgules fixes) en nombres dans les 3 autres bases que la base courante

Additions et soustractions de 2 nombres binaires, octaux et hexadécimaux (entiers, positifs et/ou négatifs)

Conversion d'un nombre réel à virgule fixe, décimal, positif ou négatif, en nombre binaire à virgule fixe et réciproquement.

Possibilité de voir le calcul entier, que ce soit pour la conversion, l’addition et la soustraction.

### Contraintes

Les fonctions prédéfinies ne sont pas autorisées et ne doivent en aucun cas être utilisées.

### Travail à réaliser par l’apprenti

L’apprenti doit créer un programme permettant aux élèves ainsi qu’aux enseignants de corriger les exercices d’ELEOC, le programme doit donc permettre la conversion de nombre positifs ou négatifs à virgule fixe entre différents formats (décimal, binaire, octal et hexadécimal), en plus de cela il doit être possible de soustraire et additionner des nombres entiers positifs et/ou négatifs dans différents format (binaire, octal et hexadécimal), tout cela doit être simple d’utilisation pour chaque élèves et enseignants de l’ETML en plus d’être intuitif.

Une planification doit être faite au début du projet avec un Gantt, un journal de travail doit aussi être fait et tenu à jour. Un rapport doit être fait et complet, il doit contenir l’intégralité du projet.

Tout le projet doit être mis sur un dépôt Git que l’élève pourra choisir par lui-même.

Une analyse du programme concernant l’applications et ses fonctionnalités doit être faite, permettant de s’assurer de la cohérence du programme entre ses différentes fonctionnalités et utilisations.

### Si le temps le permet

## Les points suivants seront évalués

La grille d’évaluation définit les critères généraux selon lesquels le travail du candidat sera évalué (documentation, journal de travail, respect des normes, qualité, …).

En plus de cela, le travail sera évalué sur les 7 points spécifiques suivants (Point A14 à A20):

* + Conversion de nombres entiers binaires signés dans les autres bases
  + Conversion de nombres réel signés, à virgule fixe, dans les autres bases
  + Conversion de nombres décimaux signés dans les autres bases
  + Conversion de nombres décimaux signés, à virgule fixe, dans les autres bases
  + Conversion de nombres hexadécimaux signés dans les autres bases
  + Ergonomie intuitive et agréable pour tout type d'utilisateur de l'ETML
  + La réalisation doit prendre en compte au moins un point significatif des bonnes pratiques en matière d'écoconception

## Validation et conditions de réussite

# Planification initiale











# Analyse

## Opportunités

Approfondir ses connaissances en C#, opportunité de créer un programme qui sera utilisé en ELEOC, par les élèves de l’ETML en première année voir des années suivantes, ainsi que les enseignants qui pourront corriger des exercices.

## Document d’analyse et conception

### Logiciel utilisé pour coder

### Maquette graphique

### Conception du programme

Le programme a besoin de pouvoir effectuer différentes opérations de calculs (conversion, addition et soustraction), afin d’éviter que cela nécessite trop de ressource ces opérations seront séparées, le programme possèdera des « modes », cela permettra de n’effectuer que l’opération voulu en affichant uniquement l’interface du  « mode » choisi sans afficher l’interface des autres, pour changer de « mode » le programme possédera une barre d’action avec un « Split Button », il s’agit d’un bouton avec une liste déroulante qui permettra de choisir le « mode » désiré pour soit convertir, additionner ou soustraire.

Dans cette même barre d’action il y aura encore un autre « Split Button », celui-ci servira à choisir différentes options comme la possibilité de définir notamment si l’on veut convertir des nombres avec virgule ou non, permettant aux élèves n’étant pas à l’aise en ELEOC de ne pas s’embrouiller à remplir de zéro ou de ne rien mettre dans la case après la virgule, donnant par la même occasion un résultat qui n’affichera aucune virgule évitant aussi de s’embrouiller et les erreurs de manipulation en entrant des chiffres après la virgule par erreur.

Pour la conversion il y aura donc comme vu au-dessus la possibilité de le faire sans virgule donnant uniquement des conversions avec des chiffres entiers, il n’y aura donc qu’une boite de texte où l’utilisateur entrera la valeur qu’il veut convertir. Si l’utilisateur choisi d’effectuer une conversion avec virgule, une deuxième boite de texte apparaitra à droite de celle de base, un point s’affichera entre ces deux, il s’agit de la virgule pour ainsi les séparer en indiquant que la deuxième contiendra ce qu’il y a après la virgule, la séparation de ce qu’il y a avant la virgule et après permet d’éviter des erreurs du programme via notamment les regexs qui seront utilisé pour vérifier le format des valeurs entrés ainsi que faciliter la conversion de ce qu’il y a avant la virgule et après.

Au sujet de la façon dont les conversions seront faites, des tableaux seront utilisés pour convertir, un algorithme sera mis en place pour afficher à l’utilisateur le calcul effectué et la démarche pour réussir à obtenir le résultat, le but est que l’utilisateur puisse obtenir le résultat avec les démarches nécessaire au calcul mais aussi qu’il puisse comprendre comment l’atteindre, cela lui permettra de se corriger, de plus s’il a des difficulté pour effectuer le calcul de lui-même il faut que le programme en affichant le calcul puisse le guider et l’aider à comprendre comment il doit faire.

## Conception des tests

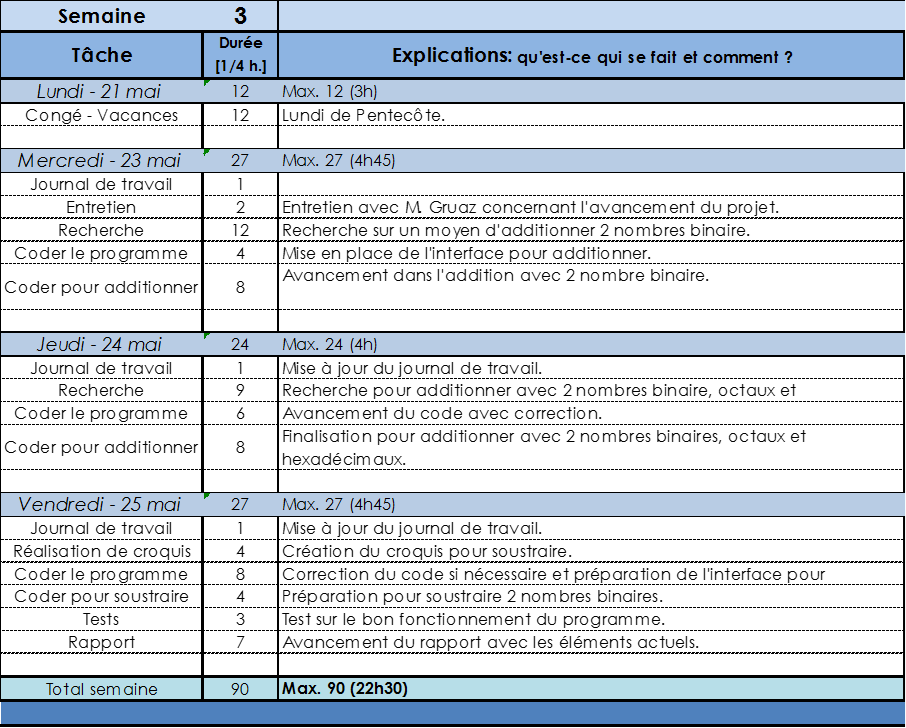
|  |  |
| --- | --- |
| **Test à effectuer** | **Ce qui est recherché** |
| Faire des conversion, addition et soustraction avec des lettres ou autre caractères lorsque le format ne doit pas le permettre | 1. Vérifier s’il y a des erreurs ou crash du programme 2. Empêcher une mauvaise manipulation |
| Vérifier les limites possibles concernant la taille des calculs ou des nombres | 1. Vérifier s’il y a des erreurs ou crash du programme 2. Limiter la taille maximal si besoin |
| Fournir le manuel d’utilisation avec le programme à plusieurs personnes pour qu’elles utilisent le programme | 1. Vérifier la simplicité d’utilisation du programme 2. S’assurer que le manuel d’utilisation soit clair et complet |
| Tester sur différent PC n’ayant pas la même configuration | 1. Voir comment le programme réagis, s’il est bien compatible et qu’il n’y a aucune erreur 2. Si ce n’est pas le cas il sera possible de soit corriger soit indiqué pourquoi il ne fonctionne pas |
| Essayer de convertir, additionner et soustraire avec un nombre supérieur à 32 bits | 1. Voir si le programme empêche bien la manipulation lorsque la limite est dépassée |
| Convertir, additionner et soustraire de petites valeurs jusqu’au maximum possible | 1. Vérifier si les fonctionnalités du programme fonctionnent correctement 2. Vérifier si une petite valeur pose problème et inversement avec une grande valeur |
| Effectuer les calculs avec nombres positifs et négatifs ainsi qu’avec virgule | 1. Vérifier le bon fonctionnement des calculs |

## Planification détaillée

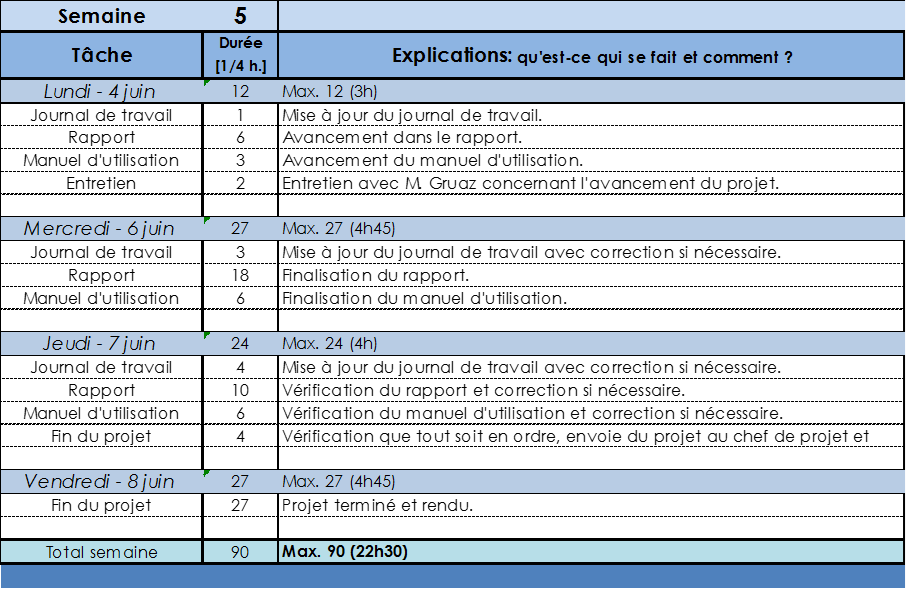
La colonne concernant les liens et références de la planification n’a pas été reprise ci-dessous, la raison étant qu’il n’y a aucune information écrite sous celle-ci, de plus en retirant ces colonnes cela permet une meilleure visibilité.











# Réalisation

## Dossier de réalisation

### Logiciels installé / Utilisés

Microsoft Visual Studio 2017

## Modifications

# Tests

## Dossier des tests

# Conclusion

## Bilan des fonctionnalités demandées

## Bilan de la planification

## Bilan personnel

# Divers

## Journal de travail

## Bibliographie

## Webographie

# Annexes