

FISGIN Muharem Ege  
GAYE Amadou  
SARRAU Arthur



# Rapport de projet informatique

## Développement d'une calculatrice scientifique avancée



## Sommaire

Sommaire.....	1
Cahier des charges.....	1
Objectif principal.....	1
Fonctionnalités.....	2
Conception globale.....	2
Analyse descendante.....	3
Types.....	3
Signatures.....	4
Résolution d'équations.....	4
Calculs classiques.....	5
Statistiques.....	5
Unités et conversions.....	6
Historique.....	7
Guide d'utilisation.....	7
Bilan organisationnel.....	8
Planning.....	8
Répartition des tâches.....	8
Support et outils.....	8
Difficultés.....	9
Compte-rendu.....	9

## Cahier des charges

### **Objectif principal**

La réalisation de ce projet avait pour but de parvenir à développer une calculatrice scientifique codée en Pascal, avec un maximum de fonctionnalités disponibles. Notre motivation première reposait sur le fait de créer un support réellement utile et que nous pourrions réutiliser tout au long de notre parcours d'ingénieur. En réalité, nous sommes partis du postulat que les calculatrices scientifiques manquaient en général d'une offre d'outils, et que seules certains logiciels spécialisés dans un type de problème étaient en capacité de répondre à cette demande. De ce fait, nous avons décidé de nous lancer dans la réalisation d'une calculatrice scientifique offrant des fonctionnalités diverses afin de lui permettre de résoudre un maximum de problèmes.

D'autre part, une ligne directrice autour de laquelle notre projet s'est constamment

construit est l'aspect fonctionnel. Entre autres, nous souhaitons avant tout parvenir à une réalisation qui réponde aux attentes premières d'une calculatrice scientifiques avant de nous concentrer sur d'autres points tels que l'esthétique. Par conséquent, et en avançant dans le projet, nous avons préféré ajouter des fonctionnalités plutôt que de développer l'aspect graphique de l'interface avec la bibliothèque SDL. Ainsi, l'ensemble des interactions humain-machine se réalisera directement via le terminal.

## **Fonctionnalités**

En termes de fonctionnalités, nous nous sommes focalisés sur 5 principales. Celles-ci seront accessibles depuis un menu principal basé sur le modèle des calculatrices type Casio FX-92 Collège<sup>®</sup>.

Premièrement, la fonctionnalité la plus évidente pour une calculatrice est le calcul simple, soit le calcul numérique d'une opération composée uniquement de symboles opératoires et de nombres. L'opération est saisie manuellement par l'utilisateur qui peut reprendre ses calculs pour les rectifier. Une sauvegarde est automatiquement réalisée à chaque opération afin de pouvoir garder en mémoire l'ensemble des calculs une fois la session fermée.

Ensuite, la calculatrice offre aussi la possibilité de réaliser de l'analyse statistique avec le calcul de grandeurs caractéristiques ou de modèle mathématique basé sur une régression linéaire, le tout depuis un échantillon statistique. Ces derniers contiennent des données qui pourront être saisies manuellement dans le terminal.

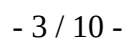
Après, la calculatrice contient également un menu consacré à la résolution d'équations différentielles d'ordre 1 et 2 ainsi que la résolution d'équations polynomiales d'ordre 2. Il ne s'agit là de ne traiter que les cas avec coefficients constants car nous n'avons pas mis en place de système permettant le calcul d'intégration au vu de la difficulté que cela représente. Ces équations seront saisies par l'utilisateur en veillant de bien respecter le format requis.

Enfin, l'une des fonctionnalités qui se distingue des calculatrices habituelles est la conversion d'unités et de constantes physiques directement intégrée. Dans ce menu, il est possible de réaliser n'importe quelle conversion d'unité vers une autre en incluant des ordres de grandeurs. Est également disponible la consultation de la valeur d'une constante ou de la dimension d'une grandeur.

Finalement, comme évoqué précédemment, un historique est disponible pour le menu calculs simples. Le tout est sauvegardé sous la forme d'un fichiers au format csv. Le contenu peut être consulté par l'utilisateur à n'importe quel instant.

## Conception globale

## INSA Rouen Normandie



## STPI 2

```
Type TRegressionResult = Record
  a, b, r_squared: Real
end;
```

```
type tableau1D=Array of Real;
```

```
type subtab= Array of Array of real;
```

Type tableau = Array of Array of String;

```
type ComplexNumber = record
  RealPart: String;
  ImaginaryPart: String;
end;
```

Type tableau = Array of Array of String;

```
Type constantes = Record
    constante, valeur, unite, dimension: string
end;
```

```
Type dimensions = Record
    grandeur, dim_SI: string;
end;
```

```
Type conversions = Record
    unite_i, unite_f, valeur_i: string
end;
```

```
Type fichiers = Record
    cv: conversions;
    ct: constantes;
    dm: dimensions
end;
```

## Signatures

### *Résolution d'équations*

-procedure Equation\_ordre\_1();

Grâce à cette procédure, l'utilisateur peut demander la dérivée de l'équation de 1er ordre puis cette procédure la résoudre quand on a seulement  $ay'=b$  et l'affiche.

-procedure Equation\_ordre\_2();

Grâce à cette procédure, l'utilisateur peut demander la dérivée de l'équation de 2er ordre puis cette procédure la résoudre pour solution homogène, et l'affiche.

-function deltasolution(var a,b,c:Real): Real;

Cette fonction prendre le coefficient de  $x^2$ ,  $x$  et 1 dans l'équation de  $x^2+bx+c=0$  et calcule la delta.

function y1(a, b: Real): String;

Cette fonction calculer solution pour la dérivée de l'équation de 1er ordre

- procedure polynomiale\_equation\_order();

Cette procédure prend une équation de  $x^2+bx+c=0$  et calcule les racines et l'affiche.

- procedure y2(a, b,c,d: Real ; var x1,x2:ComplexNumber);

Cette procédure calculer solution homogène pour la dérivée de l'équation de 2er ordre.

-procedure equation(var retour\_menu: Boolean);

Cette procédure est la procédure principale pour la partie de solution d'équation donc on appelle procédure la.

### *Calculs classiques*

-procedure lireChaine(var str: string);

Cette procédure demande à l'utilisateur l'opération mathématique qu'il souhaite effectuer et reçoit cette valeur sous forme de chaîne.

-procedure rules();

Cette procédure permet aux utilisateurs de prendre connaissance des règles de la calculatrice en termes de format de saisie.

-function parantehese(expression: String ; openPos : Integer): integer;

J'entre dans la procédure l'expression math et le début de la parenthèse et cette procédure me donne la coordonnée de la fin pour séparer l'expression principale et l'expression entre les parenthèses.

- *procedure calculsimple(var retour\_menu: Boolean);*

Cette procédure est la procédure principale pour le calcul simple donc on entre la valeur, faire le calcul mathématique et on l'affiche.

- *function calcul(s: String): Real; (var retour\_menu: Boolean): Real;*

Cette fonction est une fonction récursive donc elle prend une expression mathématique, le calcule et retourne la valeur de calcul.

### *Statistiques*

- *procedure statistiques (var retour\_menu: Boolean) ;*

Cette procédure représente le menu principal de la partie statistique. Elle permet à l'utilisateur de choisir parmi plusieurs options, telles que saisir un tableau, calculer la moyenne, la médiane, la variance, l'écart-type, réaliser une régression linéaire, enregistrer les résultats dans l'historique, ou quitter le menu statistique.

- *procedure SaisirTableau (var TabI: subtab; var n: integer) ;*

Description : Cette procédure permet à l'utilisateur de saisir un tableau de données statistiques. Elle demande le nombre de données et leur effectif, puis stocke ces informations dans le tableau 'TabI'.

- *procedure creationtableau (var tab\_med: tableau1D; const TabI: subtab) ;*

Cette procédure crée un tableau unidimensionnel 'tab\_med' à partir du tableau bidimensionnel 'TabI' en utilisant la méthode de répétition basée sur l'effectif.

- *function CalculerMoyenne (tab\_med: tableau1D): Real ;*

Cette fonction calcule et retourne la moyenne des données présentes dans le tableau unidimensionnel 'tab\_med'.

- *function CalculerMediane (tab\_med: tableau1D): Real ;*

- Description : Cette fonction calcule et retourne la médiane des données présentes dans le tableau unidimensionnel 'tab\_med'.

- *function CalculerVariance (tab\_med: tableau1D): Real ;*

Cette fonction calcule et retourne la variance des données présentes dans le tableau unidimensionnel 'tab\_med'.

- *function CalculerEcartType (CalculerVariance: Real): Real ;*

Cette fonction calcule et retourne l'écart-type à partir de la variance fournie en argument.

- *procedure regression\_lineaire (const TabI: subtab; var a, b, r\_squared: Real) ;*

Cette procédure réalise une régression linéaire sur les données de 'TabI' et renvoie les coefficients 'a' (coefficient directeur) et 'b' (ordonnée à l'origine) ainsi que le coefficient de corrélation 'r\_squared'.

### *Unités et conversions*

- *procedure unites (var retour\_menu: Boolean) ;*

Programme principal de l'unité unités et conversions. Permet de structurer et d'afficher proprement à l'écran. La variable de sortie 'retour\_menu' sert au programme principal global à savoir si l'utilisateur souhaite revenir au menu initial.

- *function parcoursfichier (var currentfile: TextFile; nom: String): String ;*

Détermine le nombre de lignes et colonnes que contient un fichier .csv de séparateur ' ; ' afin de dimensionner comme il le faut les tableaux de données ensuite. Le fichier est désigné par la variable 'currentfile' et son emplacement est donné par la variable d'entrée 'nom' à laquelle on rajoute le reste du chemin d'accès du dossier contenant les exécutable.

- *procedure chargementdonnees (var t:tableau; var nm: string; nom: String) ;*

Transforme un fichier .csv en un tableau dynamique Pascal noté 't'. Réutilise la procédure ci-dessus pour dimensionner le tableau de taille renseignée par 'nm' qui est égale à parcoursfichier(currentfile,'nom').

- *function conversion (var currentfile: TextFile): Extended;*

Réutilise les deux fonctions précédentes et permet de réaliser des conversions de n'importe quelles unités, et ce peu importe l'ordre de grandeur considéré (kilo, micro, peta...). Tant qu'il existe dans le fichier de référence 'currentfile' servant de base donnée, un lien entre l'unité considérée et une unité de référence, la conversion est réalisable. Elle s'appuie également sur le fichier contenant les équivalences entre ordres de grandeurs et puissances, qui n'a une portée locale uniquement.

- *function consulter (var currentfile: TextFile): String ;*

Permet simplement de consulter la valeur d'une constante ou de vérifier la dimension d'une grandeur. Le fichier désigné par la variable 'currentfile' correspond selon le choix d'utilisateur au fichier contenant les constantes ou bien celui contenant les dimensions.

- *procedure ajout (var f: fichiers) ;*

Offre l'opportunité d'ajouter directement depuis l'interface de la calculatrice des données à n'importe quel fichier du choix de l'utilisateur. Elle réutilise la procédure chargementdonnes(currentfile,<nom\_fichier\_choisi>).

## *Historique*

*procedure historique(var retour\_menu: Boolean);*

Programme principal du menu historique, qui permet à l'utilisateur de naviguer dans le menu et de choisir les fonctionnalités.

*procedure consultation(var save\_calc: Textfile);*

Permet de consulter l'historique de calcul directement depuis le terminal.

*procedure viderhistorique(var save\_calc: Textfile);*

Permet de vider l'historique si l'utilisateur n'en a plus besoin.

## Guide d'utilisation

Pour ce qui est d'un éventuel guide d'utilisation, nous avons estimé qu'il n'était pas nécessaire d'en présenter un dans ce rapport puisque notre calculatrice incorpore déjà de nombreuses indications claires et précises à chaque étape.



# Bilan organisationnel

## **Planning**

En termes de planification sur la durée du projet, on peut découper notre progression sur ce projet en 3 phases principales.

Tout d'abord, jusqu'à fin octobre, soit le premier mois et demi, nous avons choisi de nous pencher sur les fonctionnalités essentielles, nous avons donc développé les menus calcul simple, statistiques et conversions d'unités. Cette partie est celle qui a été la plus longue, puisque qu'il fallait s'assurer que tout s'effectue sans accroc, puisque nous avons anticipé de réutiliser ces premiers menus pour simplifier le développement des suivants.

Concernant ces derniers, pendant le mois de novembre, nous nous sommes ensuite penchés sur le menu résolution d'équations et sur l'historique. Cette phase n'a duré que quelques semaines, jusqu'à fin novembre, puisque comme dit précédemment nous sommes parvenus à « recycler » littéralement certaines parties de notre code d'autres unités.

Enfin, le mois de décembre fût consacré à une phase de réajustements et à de nombreux tests. Nous avons en même temps rajouté quelques nouvelles fonctionnalités supplémentaire, comme le chargement de tableau statistiques. Plus longue qu'espérée, cette phase d'assemblage d'unités, de programme principal et de tests, a été menée avec le maximum de sérieux pour nous assurer que tout fonctionnait comme nous le souhaitions et que dans aucun cas, notre programme n'était amené à boucler à « crasher ».

## **Répartition des tâches**

En terme d'organisation et de répartition des tâches nous avons fait le choix de distribuer le travail par menu, à l'exception de l'historique et unités de conversions pour certaines fonctions et procédures, qui était transversal et pour lequel a été fourni une collaboration davantage collective. Par conséquent, chaque personne a donc travaillé sur au moins deux menus dans leur intégralité :

- Ege : Calcul simple / Résolution d'équations
- Amadou : Statistiques / Historique
- Arthur : Unités / Programme principal

## **Support et outils**

Dès le début du projet il a en tout premier lieu mettre en place un système de partage de code efficace. De ce fait, nous avons opté pour la plateforme GitHub et de son terminal associé Git Bash, qui nous permet en seulement quelques lignes d'instructions de déposer le code modifié et d'avoir accès aux mises à jours des codes de l'ensemble du groupe de travail.

D'autre part, à propos de l'éditeur de texte choisi, nous nous sommes plutôt orientés vers Visual Studio Code qui offre des extensions réellement efficaces. Le détecteur automatique d'erreurs, la création d'un espace de travail avec tous les fichiers de données et codes, l'auto-prédiction des instructions et la compilation de l'ensemble des bibliothèques utilisées en une seule compilation d'un programme principale. Telles ont été les fonctionnalités et atouts qui nous ont convaincus d'utiliser cet éditeur.

## Difficultés

Puisqu'il s'agit de notre toute première expérience à mener un projet à l'informatique en groupe, nous avons rencontrées plusieurs difficultés lors de cette réalisation.

Techniquement déjà, l'appropriation et la maîtrise de l'outil GitHub, plus particulièrement du terminal Git Bash, a été complexe à mettre en place. Mais nous avons pris le temps de tout configurer comme il se doit, car nous savions que cela simplifierait grandement notre collaboration par la suite.

Après on peut également classer comme la difficulté le fait qu'Amadou, qui est un intégré, n'avait jamais codé en Pascal avant d'arriver à l'INSA. Bien qu'il ait déjà programmé en d'autres langages auparavant, il a donc fallu prendre un temps conséquent pour s'approprier ce nouveau langage, et ce au détriment parfois de l'avancement du projet dans un premier temps. Bien évidemment le reste du groupe lui est venu en aide dès qu'il en avait besoin et nous avons fait tout notre possible pour lui expliquer les concepts clés et fonctionnements de ce langage. Le fait qu'Amadou est travaillé sur l'unité statistique a facilité son avancée dans le projet car cet unité requiert une appropriation de la bibliothèque Math et se concentre sur des techniques de programmation classiques et transposables à d'autres langages.

Ensuite, s'ajoutent les défis que la collaboration au sein d'un groupe impose. En effet, les efforts fournis par chacun et l'avancement inégal du projet dans le temps ont été des facteurs auxquels nous nous sommes rapidement confrontés. Il s'agit d'enjeux de travaux de groupes généraux, mais qui se voient davantage accentués en informatique par le fait que l'avancement des uns peut limiter le développement des autres s'il n'a pas les outils requis en temps et en heure ou s'il y a des mésententes sur la portée de variables au sein de procédures ou fonctions.

Après un problème auquel nous ne nous attendions pas à faire face est l'application de chacun sur la syntaxe du code produit. Il s'agit d'un problème à priori mineur, toutefois il peut s'avérer être chronophage puisque la relecture de centaines de lignes de codes mal indentées ou agencées maladroitement est parfois compliquée lorsque nous sommes habitués à notre façon de structurer le code. Au milieu du projet, nous avons donc réalisé un bilan sur la façon dont devaient être présentées les instructions afin que nous nous harmonisions sur un même modèle, qui nous ferait sans aucun gagner du temps sur la relecture collective.

## Compte-rendu

Ainsi, ce projet informatique est une première expérience offrant un aperçu intéressant de la réalité du travail d'ingénieur. Le travail en équipe, la réponse à un cahier des charges et l'avancement structuré d'un projet représentent des qualités indispensables dans le monde professionnel.

Globalement, nous sommes réellement fiers de ce que nous avons produit, et cela répond à nos attentes initiales. Avec plus de temps à disposition, nous aurions certainement choisi de développer des fonctionnalités et menus supplémentaires : affichage graphique de fonction, résolution de systèmes linéaires et matriciels ou encore possibilité de produire des codes simples en Python qui pourraient être exécutés en externe, comme sur la calculatrice Numworks<sup>®</sup>.

En somme, ce projet nous a ouvert les yeux sur la praticité de nombreux outils et de l'importance de leur maîtrise ainsi que de la nécessité de communiquer et d'établir des standards au sein d'un groupe de travail. Ce projet nous a permis dans le même temps, de progresser en programmation pure et de développer de nouvelles compétences, qui dépassant le seul cadre de l'informatique.