**Zamader Achraf, Babich Alexandre**

Enseignant responsable : Cyril Girard

Document réalisé grâce à l’exemple fourni dans le Classroom

**Travail personnel individuel (TPI)**

AlgotRI

Rapport TPI et documentation technique

# Table des matières

[**Table des matières 1**](#_Toc136862678)

[**Table des versions 1**](#_Toc136862679)

[**Introduction 2**](#_Toc136862680)

[**Rappel de l’énoncé 2**](#_Toc136862681)

[Organisation 2](#_Toc136862682)

[Livrables 2](#_Toc136862683)

[Matériel et logiciels à disposition 2](#_Toc136862684)

[Description de l’application 2](#_Toc136862685)

[**Méthodologie 3**](#_Toc136862686)

[1. S’informer 3](#_Toc136862687)

[2. Planifier 3](#_Toc136862688)

[3. Décider 3](#_Toc136862689)

[4. Réaliser 3](#_Toc136862690)

[5. Contrôler 4](#_Toc136862691)

[6. Évaluer 4](#_Toc136862692)

[**Planification 5**](#_Toc136862693)

[**Généralités concernant l’implémentation 6**](#_Toc136862694)

[Structure du projet 6](#_Toc136862695)

[**Outils externes 7**](#_Toc136862696)

[Git 7](#_Toc136862697)

[**Plan de test et tests 7**](#_Toc136862698)

[Périmètre des tests 7](#_Toc136862699)

[Scénarios de test 7](#_Toc136862700)

[Évolution des tests 10](#_Toc136862701)

[**Conclusion 10**](#_Toc136862702)

[Difficultés rencontrées 10](#_Toc136862703)

[Améliorations possibles 10](#_Toc136862704)

[Bilan personnel 10](#_Toc136862705)

# Table des versions

|  |  |  |  |
| --- | --- | --- | --- |
| **No de version** | **Date** | **Auteur(s)** | **Notes** |
| **1.0** | 01.06.2023 | Zamader Achraf, Alexandre Babich | Version finale du document pour le rendu du TPI |

# Introduction

Ce rapport décrit différents points du projet AlgoTri. Ce projet est le travail final de notre 3ème année au CFPT et a pour but de nous entraîner à la réalisation de notre TPI en 4ème année.

AlgoTri est une application qui a pour but de faire expérimenter à l’utilisateur différentes méthodes de tri.

# Rappel de l’énoncé

*Les informations suivantes sont extraites du cahier de charges du TPI.*

## Organisation

|  |  |  |
| --- | --- | --- |
| **Élève** |  | **Enseignant responsable** |
| Zamader Achraf, Babich Alexandre  <achraf.zmdr@eduge.ch> <alexandre.bbch@eduge.ch> | Cyril Girard  <edu-girardcy@eduge.ch> |

## Livrables

* Planning détaillé du projet
* Rapport de projet contenant le code source au format PDF
* Journal de bord
* Résumé du TPI (1 page A4)
* Pour le formateur uniquement :
  + L’accès au repository distant du projet avec les droits de « clone »
  + Un readme explicitant l’installation du projet en local
  + Un dump de la base de données contenant la structure ainsi qu’un set de données de test

## Matériel et logiciels à disposition

* Un PC standard école avec Windows 10, 2 écrans
* Visual Studio : Project C# - Windows Form
* Suite Office

## Description de l’application

AlgoTri est une application C# qui permet d’expérimenter différents algorithmes de tri. Elle permet également de les manier pour mieux les comprendre.

Les fonctionnalités suivantes doivent être implémentées :

* **Méthodes de tri**

Chaque Form permet d’utiliser une méthode de tri distincte.

* **Représentation des valeurs**

Dans chaque Form, lorsque l’utilisateur clique sur le bouton « Trie », le tri des nombres commence et les valeurs sont représentées grâce à un graphique à barres. Les rectangles sont disposés de manière horizontale et avec une couleur différente pour chacun d’eux. La hauteur des rectangles représente leur poids.

* **Pseudo-code**

Dans chaque Form, l’utilisateur peut suivre le tri en cours d’exécution grâce à l’affichage du pseudo-code ligne par ligne.

* **Vitesse d’exécution**

L’utilisateur peut régler la vitesse d’exécution du tri :

Pas à pas – Très Lent – Lent – Normal – Rapide

* **Trie, Stop, Continuer**

L’utilisateur peut cliquer sur le bouton trier, lorsque le bouton trier est cliqué, les boutons stop et continuer sont activés. Ils permettent individuellement de stopper le tri temporairement et de continuer celui-ci.

# Méthodologie

Pour la planification, nous avons usé de la méthode en 6 étapes.

## Une image contenant dessin, dessin humoristique, guitare, instrument de musique Description générée automatiquement1. S’informer

Nous devions prendre connaissance de ce que le projet nous demandait de réaliser et notre enseignant responsable nous a également aider à comprendre en détails l’application que nous devions coder.

## 2. Planifier

Nous avons planifié la réalisation des étapes du projet notamment grâce à l’exemple donné dans le Classroom (fichier Excel).

## 3. Décider

Nous avons ensuite décidé comment réaliser de manière efficace ce projet (priorités des tâches, qui fait quoi)

## 4. Réaliser

Après réflexion, nous avons réalisé l’application demandée, le journal de bord et la documentation.

## 5. Contrôler

Nous avons tout au long de la conception, réalisé des tests pour vérifier si tout était fonctionnel et contrôler que toutes les tâches étaient faites.

## 6. Évaluer

Grâce au journal de bord, nous avons pu évaluer l’état de notre projet et notre manière de travailler. Nous avons récapitulé chaque jour ce que nous avons fait, comment nous l’avons fait, si tout c’était passé comme prévu et ce que nous devions faire le lendemain.

# Planification

Sur ce tableau excel, nous avons planifié nos tâches et le temps nécessaire pour les réaliser. Nous les avons découpé par « front-end », « back-end » et « administration ». Nous étions largement en avance sur notre planning et cela nous a grandement avantagé pour la finition de notre projet.

# 

# Généralités concernant l’implémentation

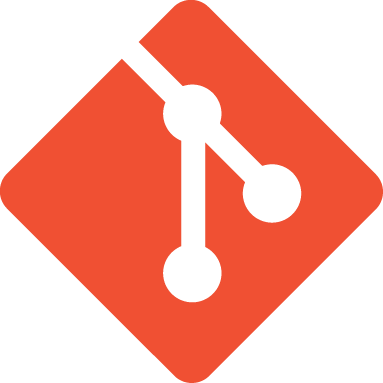
## Structure du projet

La structure du projet est simple, la form principal AlgoTri permet de rediriger grâce des boutons sur la form de tri que l’utilisateur souhaite. Ensuite, chaque form correspond à une méthode de tri distincte.

* **AlgoTri :** Menu permettant d’accéder à la méthode de tri que l’on souhaite
* **FrmBubble.cs :** Form permettant d’utiliser la méthode de tri à bulle.
* **FrmComb.cs :** Form permettant d’utiliser la méthode de tri à peigne.
* **FrmInsert.cs :** Form permettant d’utiliser la méthode de tri par insertion.
* **FrmSelect.cs :** Form permettant d’utiliser la méthode de tri sélectif.
* **FrmShell.cs :** Form permettant d’utiliser la méthode de tri shell.

# Outils externes

## Git

Git a été utilisé pour faciliter la conception du travail et la gestion des versions de notre application.

# Analyse des fonctionnalités majeures

## Triage

La fonctionnalité principale et qui est l’essence même du projet est le triage. La méthode de tri diffère selon les Form. Pour nous aider à développer les algorithmes de tri, notre enseignant responsable nous a montré et recommandé le site [*http://lwh.free.fr/*](http://lwh.free.fr/). Site qui a pour but de démontrer différentes méthodes de tri et leur code en C, Pascal, etc. Pour l’algorithme on :

Initialise une variable booléenne "hasSwapped" à "false".

Effectue une boucle sur les éléments non triés du tableau et échanger les éléments adjacents s'ils sont dans le mauvais ordre. On et "hasSwapped" à "true" si un échange a été effectué.

Si aucun échange n'a été effectué ou si tous les éléments sont déjà triés, on termine l'algorithme.  
Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Une image contenant texte, capture d’écran

Description générée automatiquement

## Vitesses d’exécution

Le programme permet également de gérer la vitesse d’exécution du triage. L’utilisateur peut cocher la vitesse qu’il souhaite avant le tri. La vitesse d’exécution est gérée grâce au Timer, on change la valeur de l’intervalle :

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Pour la méthode Pas à pas, on initialise une variable « currentStep » qui permet de savoir à quelle étape on est dans l’exécution du programme. [[*Voir l’exemple dans le tri à bulle*](#_Triage)]

## Graphiques

La représentation du tri se fait grâce à un graphique à barres, les valeurs sont inscrites en dessous des rectangles.

Une image contenant Caractère coloré, conception

Description générée automatiquement

Voici le code qui permet de réaliser cela : Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Ces lignes de code permettent de dessiner des rectangles représentant les valeurs d'un tableau, avec chaque rectangle ayant une hauteur proportionnelle à la valeur de l'élément correspondant. Les rectangles sont positionnés horizontalement et verticalement, et chaque rectangle est coloré pour faciliter la vue. Enfin, le nombre correspondant à chaque valeur est affiché en dessous du rectangle dans une police et une couleur spécifiée.

# Plan de test et tests

## Périmètre des tests

Le périmètre de test va englober tous les systèmes de tri, leur visuel ainsi que les fonctionnalités qui permettent de manier le déroulement de l’exécution comme les boutons trier, stop, continuer ainsi que les vitesses d’exécution.

## Scénarios de test

Voici les scénarios de test que nous avons préparé avec la planification des tâches :

|  |  |
| --- | --- |
| **Test** | **1.0 Visuel des Forms** |
| **Tâche** | Faire le visuel des Forms |
| **Evénement** | Lorsque l’utilisateur lance le programme, il peut apercevoir 5 boutons menant chacun à la Form de tri correspondant. Lorsqu’il arrive sur les Forms de tri, le bouton « trie » est affiché et activé, les boutons « stop » et « continuer » sont affichés mais désactivés. Les vitesses d’exécution sont affichées à gauche et le textbox du pseudo-code est affiché à droite. |
| **Résultante** | Je lance le programme, les 5 boutons menant chacun à la Form de tri correspondant sont affichés. Lorsque je clique sur l’un d’entre eux, une nouvelle fenêtre s’ouvre, le bouton « trie » est affiché et activé, les boutons « stop » et « continuer » sont affichés mais désactivés. Les vitesses d’exécution sont affichées à gauche et le textbox du pseudo-code est affiché à droite. |
| **Etat** | ✔ OK |
|  |  |
| **Test** | **1.1 Vitesse d’exécution** |
| **Tâche** | Réglage de la vitesse d’exécution dans tous les tris |
| **Evénement** | A gauche de la fenêtre, les vitesses d’exécution sont affichées et lorsque l’utilisateur coche l’une d’entre elles et clique sur le bouton « trie ». La vitesse d’exécution correspond à la vitesse que l’utilisateur a cochée. |
| **Résultante** | Je coche la vitesse « Pas à pas » et clique sur le bouton « trie » : L’algorithme avance à chaque fois que j’appuie sur le bouton « continuer »  Je coche la vitesse « Très Lent » et clique sur le bouton « trie » : le tri se fait très lentement.  Je coche la vitesse « Lent » et clique sur le bouton « trie » :  le tri se fait lentement.  Je coche la vitesse « Normal » et clique sur le bouton « trie » :  le tri se fait à une vitesse normale.  Je coche la vitesse « Rapide » et clique sur le bouton « trie » :  le tri se fait rapidement. |
| **Etat** | ✔ OK |

|  |  |
| --- | --- |
| **Test** | **2.0 Représentation graphique** |
| **Tâche** | Représenter les tris graphiquement |
| **Evénement** | Lorsque l’utilisateur clique sur le bouton « trie », le graphique à barres s’affiche, les rectangles sont affichés à l’échelle de leur poids et les valeurs des rectangles sont affichés en dessous de ces derniers. |
| **Résultante** | Je clique sur le bouton « trie », le graphique à barres s’affiche, la hauteur des rectangles est proportionnelle à leur valeur et les nombres sont affichés en dessous de leur rectangle respectif |
| **Etat** | ✔ OK |
| **Test** | **2.1 Bouton Stop** |
| **Tâche** | Bouton stop durant les tris |
| **Evénement** | Pendant le déroulement du tri, l’utilisateur peut à n’importe quel moment cliquer sur le bouton « stop » et cela met en pause le tri |
| **Résultante** | Je clique sur le bouton « stop » durant le tri et l’algorithme s’arrête |
| **Statut** | ✔ OK |
|  |  |
| **Test** | **2.2 Ligne de code affichée durant le tri** |
| **Tâche** | Ligne de pseudo-code durant le tri |
| **Evénement** | Lorsque l’utilisateur appuie sur le bouton « trie », le textbox à droite affiche en direct la ligne du pseudo-code exécutée. La ligne de code se met à jour au fur et à mesure que le tri avance |
| **Résultante** | J’appuie sur le bouton « trie » et le textbox de la ligne de pseudo-code se met à jour continuellement au fur et à mesure que l’algorithme s’exécute. |
| **Etat** | ✔ OK |

## Évolution des tests

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test | **J0** | **J1** | **J2** | **J3** | **J4** | **J5** | **J6** | **J7** | **J8** | **J9** | **J10** |
| 1.0 | ✘ | ✘ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| 1.1 | ✘ | ✘ | ✘ | ✘ | ✘ | ✘ | ✘ | ✔ | ✔ | ✔ | ✔ |
| 2.0 | ✘ | ✘ | ✘ | ✘ | ✘ | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |
| 2.1 | ✘ | ✘ | ✘ | ✘ | ✘ | ✘ | ✔ | ✔ | ✔ | ✔ | ✔ |
| 2.2 | ✘ | ✘ | ✘ | ✘ | ✘ | ✘ | ✘ | ✘ | ✔ | ✔ | ✔ |

✘ NOT OK ✔ OK

# Conclusion

## Difficultés rencontrées

Au cours de notre projet, nous avons fait face à plusieurs difficultés et ralentissements :

* Le mode pas à pas était très difficile à configurer, nous avons rencontré de multiples bugs par exemple lorsque le mode pas à pas était activé le graphique ne s’affichait plus. Cela nous a pris jusqu’au 7ème jour pour finaliser les réglages des vitesses d’exécution et donc du mode pas à pas.
* Nous ne savions pas comment réaliser le visuel du graphique et nous ne savions pas comment faire pour faire en sorte que les rectangles soient à l’échelle de leur valeur.

## Améliorations possibles

* Améliorer le visuel du graphique en créant des animations lorsque le tri change de place des rectangles. Cela pourrait aider l’utilisateur à mieux apercevoir ce qu’il se passe lors du tri.
* Lorsque l’algorithme se lance, l’utilisateur ne peut voir que le pseudo-code et ne peut pas suivre le tri avec d’autres langages comme le C+, le Pascal, le caml, etc.

## Bilan personnel

Nous avons particulièrement aimé faire ce projet étant donné qu’il nous prodigue un lot de connaissances qui nous sera sans doute utile pour la conception de notre TPI ainsi que tout au long de notre dernière année au CFPT. Ce projet nous a permis d’affiner nos compétences en C# ce qui nous aidera grandement lors de notre 4ème année au vu du nombre de périodes de C# que nous aurons.

COde source



















