

Devoir Maison : PARIS X GAMES 2019

par MARTIN Baptiste et MICHELET Aymerick

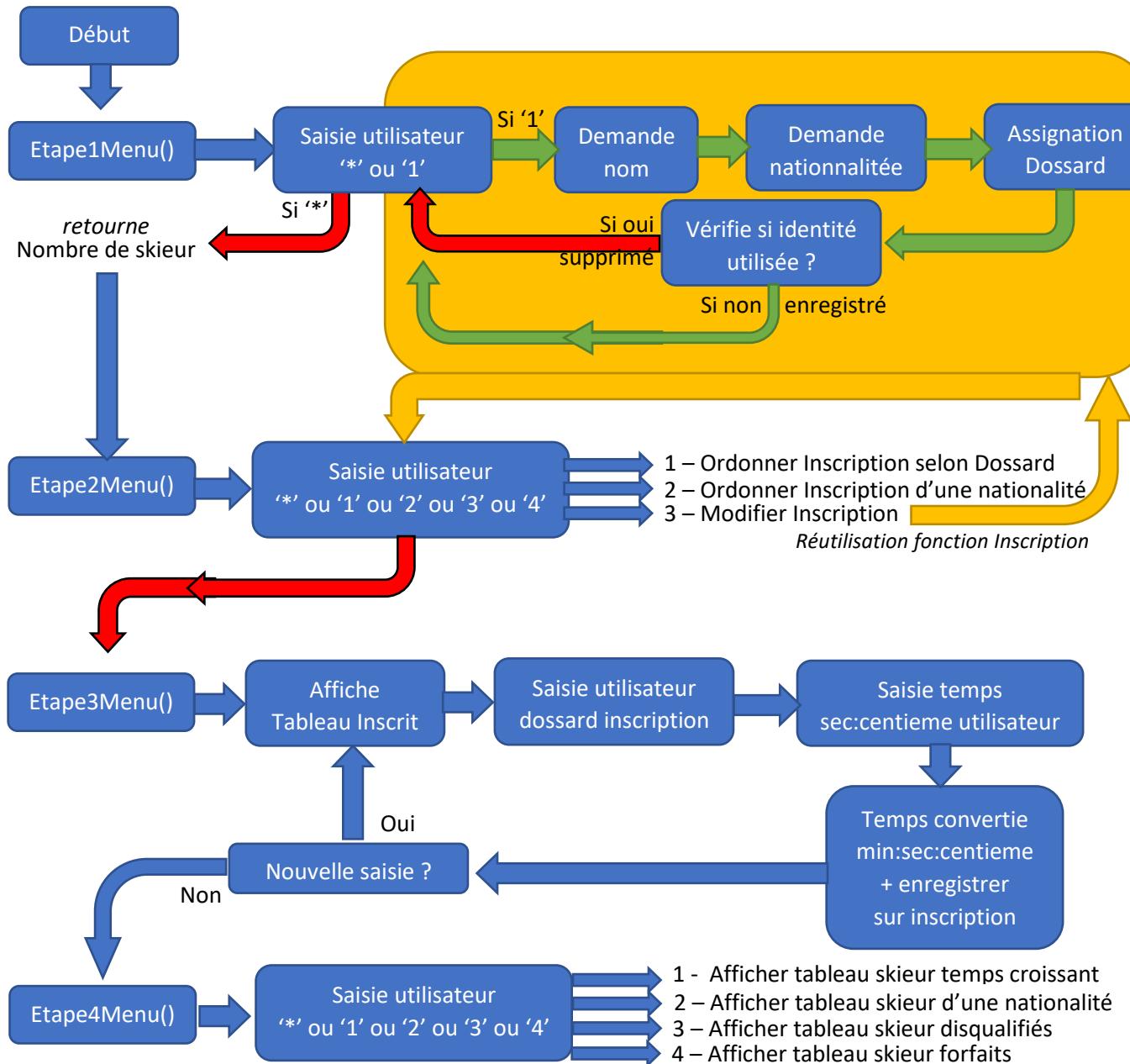
Notre programme a été créé spécialement pour l'événement « PARIS X GAMES 2019 », sous la demande du président de la station de ski des Buttes Chaumont : Bernard Laporte. Il a pour objectif d'enregistrer les inscriptions et les temps des skieurs afin de les classer facilement lors de l'affichage final.

Sommaire :

- **Algorithme générale**
(algorithme schématique générale du fonctionnement du programme)
- **Architecture technique**
(arborescence des fichiers, choix des structures de données, justification des variables principales, nomenclature des fonctions (prototype + rôle))
- **Notice utilisateur**
(rapide prise en main du logiciel)
- **Conclusion**
(auto-critique, perspectives potentielles d'évolution...)

- Algorithme générale

(algorithme schématique générale du fonctionnement du programme)



- Architecture technique

(arborescence des fichiers, choix des structures de données, justification des variables principales, nomenclature des fonctions (prototype + rôle))

Pour enregistrer l'ensemble des inscriptions des skieurs, nous avons utilisé un tableau de structure. La notation peut être fastidieuse, mais son utilisation est simple et claire. Facile d'ajouter un nouveau skieur (juste à définir la taille du tableau à son initialisation), facile de modifier un élément de la structure car elle n'a pas besoin d'être retourné.

Et grâce à ce tableau de structure, nous n'avons eu besoin que d'une seule variable générale, que l'on retourne lors de la fin des inscriptions, qui détermine le nombre de skieur en compétition, et donc la taille du tableau de structure.

L'arborescence des fichiers n'a pas été faite. De même, pour les prototypes car la partie organisation n'a pas été faite en temps et en heures. Nous avons préféré faire un programme qui fonctionne, qu'un programme dont l'arborescence des fichiers est bâclée et ne fonctionnant pas.

- Notice utilisateur

(rapide prise en main du logiciel)

Lancement du programme :

Pour lancer le programme, il vous suffit de double cliquer sur l'exécutable .exe nommé « ParisXGames2019 ». Une fenêtre alors apparaîtra, et toutes les indications dont vous aurez besoin apparaîtront. Cependant, nous ne sommes pas à l'abri de rencontrer un problème inattendu. Pour cela, et si vous rencontrez des problèmes lors de l'utilisation, vous pouvez nous contacter à l'aide de notre adresse email : micheletaymerick@gmail.com / contact@baptiste-mrt.fr

Déroulement du programme :

Le programme se déroule en 4 étapes :

- La première permet de récolter l'ensemble des inscriptions des skieurs (nom, nationalité + attribution automatique d'un numéro de dossard),
- La deuxième permet d'analyser et de corriger si besoin, les inscriptions récemment créées,
- La troisième permet d'entrer les temps de chaque skieurs (il y a la possibilité de déclarer certain skieur comme disqualifié – il vous suffira d'entrer 0 – et également comme forfait - par défaut si aucune valeur sont rentrées ou si vous entrez -1 comme valeur).
- Enfin, la quatrième et dernière étape nous permet de faire différent classement :
 - o Un classement par temps,
 - o Un classement par temps d'une nationalité précise,
 - o Un référencement des disqualifiés,
 - o Et un référencement des forfaits.

- **Conclusion**
(auto-critique, perspectives potentielles d'évolution...)

Le programme est fonctionnel et répond aux attentes du devoir. L'optimisation a été portée sur le nombre de variable utilisé car seul 2 variables générales sont utilisées (nombre de skieur + tableau de structure).

Cependant, le tableau de structure reste lourd... tout cela reste donc une question de point de vu et d'utilisation du programme (Quel support ? A quel usage ?...).

Comme nous l'avons précédemment expliqué, une arborescence ainsi que l'utilisation de prototype dans des headers auraient donné un devoir beaucoup plus soigné et lisible. Diviser le programme en plusieurs fichier .c également...

Pour les évolutions (excepté les améliorations déjà énoncé), nous imaginons même que le programme peut devenir un logiciel « non-console » (autrement dit un logiciel graphique), avec une petite interface utilisateur simple et efficace. On aurait par exemple accès à une documentation utilisateur beaucoup plus facilement, et également un coté plus « pro » du programme car évidemment un programme console reste très rudimentaire.