

# Résumé des tâches 3 du mini projet TI1 2023/2024

## 1) Planning du suivi semaine par semaine des tâches du mini projet

- Quelques modules complémentaires peuvent être ajoutés au planning ci-dessous dans les prochaines versions du document résumé des tâches.
- **Dimanche 11 février 2024 avant 18h :**
  - Déposer la carte géographique tracée à la main et photographiée "jpg",
  - Déposer les 4 fichiers texte : zones.txt, lignes.txt, troncons.txt et flottebus.txt remplies à l'aide de bloc-notes ou équivalent.
- **Dimanche 18 février 2024 avant 18h :**
  - Déposer la déclaration des enregistrements et des 4 tableaux.
  - Déposer les modules de chargement, affichage et sauvegarde de zones, lignes.
  - Déposer le module qui utilise les autres modules décrits précédemment pour dérouler l'application.
- **Dimanche 25 février 2024 avant 18h :**
  - Déposer les modules de chargement, affichage et sauvegarde de bus leurs tâches, des tronçons et rues.
- **Dimanche 03 mars 2024 avant 18h :**
  - Déposer le module d'affectation d'une personne à une zone.
  - Déposer les modules de génération aléatoire, affichage, sauvegarde puis rechargement de personnes depuis le fichier personnes\_sauv.txt.
- **Dimanche 10 mars 2024 avant 18h :**
  - Déposer le module de recherche de ligne et sens pour aller d'une zone source à une zone destination (et son utilisation lors de la génération des activités de personnes et autre).
  - Déposer le module de calcul de la distance entre une zone source et une zone destination.
- **Dimanche 31 mars 2024 avant 18h :**
  - Déposer les modules d'initialisation et d'affichage de la carte géographique (en affichant le nombre de personnes par zone et bus "voir les cartes à droite dans les annexes", en tenant compte du cas du bus multiple).
  - Déposer le module de détermination du numéro de la tâche, si un bus donné va démarrer une tâche à un horaire donné
  - Déposer le module de déplacement d'un bus.
  - Déposer les modules du déroulement d'une semaine minute par minute
- **Dimanche 07 avril 2024 avant 18h :**
  - Déposer Modules du déroulement d'une semaine minute par minute avec plusieurs vitesses (passer une minute, 10 minutes, 1heure, 4heures, 1jour).
- **Dimanche 21 avril 2024 avant 18h :**

- Déposer le module de déplacement complet d'une personne d'un tableau personnes à un autre.
- **Dimanche 28 avril 2024 avant 18h :**
  - Déposer le module de mise à jour informations des personnes d'une zone.
  - Déposer le module de mise à jour informations des personnes de toutes les zones.
- **Dimanche 05 mai 2024 avant 18h :**
  - Déposer le module d'embarquement des passagers d'un bus.
  - Déposer le module de débarquement des passagers d'un bus.
  - Mettre à jour les autres modules afin d'intégrer le déplacement de personnes dans l'application.
- **Dimanche 12 mai 2024 avant 18h :**
  - Déposer le module de calcul de gain net de bus, lignes et total.
  - Déposer les modules d'affichage et de sauvegarde de statistiques de gain.

## 2) Déclarations des modules

Les étudiants sont invités à utiliser les déclarations suivantes :

(Il est à rappeler que les étudiants sont invités à **assister chaque semaine à la validation** de leurs dépôts, en cas **d'absence** le travail sera évalué comme **invalide** même si le dépôt a eu lieu)

**En rouge :** les nouveaux modules (à copier/coller).

**En bleu :** les modules partiellement modifiés (à copier/coller et faire les modifications nécessaires).

- ✓ void chargerZones(zone zones[], int \*nombreZones)
- ✓ void chargerLignes(ligne lignes[], int \*nombreLignes)
- ✓ void chargerTroncons(troncon troncons[], int \*nombreTroncons)
- ✓ void chargerFlotteBus(bus flotteBus[], int \*nombreBus)
- ✓ void afficherZones(zone zones[], int nombreZones)
- ✓ void afficherLignes(ligne lignes[], int nombreLignes)
- ✓ void afficherTroncons(troncon troncons[], int nombreTroncons)
- ✓ void afficherFlotteBus(bus flotteBus[], int nombreBus)
- ✓ void sauvegarderZones(zone zones[], int nombreZones, char\* nomFichier)
- ✓ void sauvegarderLignes(ligne lignes[], int nombreLignes, char\* nomFichier)
- ✓ void sauvegarderTroncons(troncon troncons[], int nombreTroncons, char\* nomFichier)
- ✓ void sauvegarderFlotteBus(bus flotteBus[], int nombreBus, char\* nomFichier)

void affecterPersonneZone(zone zones[], personne pers)

void genererPersonne(zone zones[], int nombreZones, ligne lignes[], int nombreLignes)

void afficherPersonnes(zone zones[], int nombreZones)

void sauvegarderPersonnes(zone zones[], int nombreZones, char\* nomFichier)

void chargerPersonnes(zone zones[], int\* nombreZones, char\* nomFichier)

void remplissageComplet(zone zones[], ligne lignes[], troncon troncons[], bus flotteBus[], int\* nombreZones, int\* nombreLignes, int\* nombreTroncons, int\* nombreBus)

void affichageComplet(zone zones[], ligne lignes[], troncon troncons[], bus flotteBus[], int nombreZones, int nombreLignes, int nombreTroncons, int nombreBus)

void sauvegardeComplete(zone zones[], ligne lignes[], troncon troncons[], bus flotteBus[], int nombreZones, int nombreLignes, int nombreTroncons, int nombreBus, listeLiaisons \*l, char\* nomFichier)

void initialiserCarte(casecarte cartegio[20][20], zone zones[], ligne lignes[], troncon troncons[], bus flotteBus[], int nombreZones, int nombreLignes, int nombreTroncons, int nombreBus)

void afficherCarteGeo(casecarte cartegio[20][20] , zone zones[], bus flotteBus[], int nombreZones, int nombreLignes, int nombreTroncons, int nombreBus)

int rechercherLigneSens(int idZoneSrc, int idZoneDest, ligne lignes[] , int nombreLignes, int idLigne)

int calculerDistance(int idZoneSrc, int idZoneDest, ligne lignes[] , int idLigne, troncon troncons[] , int nombreLignes, int nombreTroncons)

int numTacheSuivante(horaire horaireCourant, int idBusCourant, bus flotteBus[])

void deplacerBus(casecarte cartegio[20][20], int idBusCourant, zone zones[], ligne lignes[], troncon troncons[], bus flotteBus[] , int nombreZones, int nombreLignes, int nombreTroncons , int nombreBus, horaire horaireCourant)

void deplacerPersonne(personne src[], personnes dest[], int indicePersonneSrc, int\* nombrePersonnesSrc, int\* nombrePersonnesDest)

void majPersonnesUneZone(horaire horaireCourant, zone zones[]l, int idZone, ligne lignes[] , int nombreLignes , troncon troncons[], int nombreTroncons)

void majPersonnesToutesZones(horaire horaireCourant, listeLiaisons \*l, zone zones[], int nombreZones, ligne lignes[] , int nombreLignes , troncon troncons[], int nombreTroncons)

void embarquerPersonnes(zone zones[], bus flotteBus[] , int idZone, int idBus, horaire horaireCourant)

void débarquerPersonnes(zone zones[], bus flotteBus[] , int idZone, int idBus)

ticket venteTicket(horaire horaireVente, idZoneSrc, int idZoneDest, int idLigne, int sens, ligne lignes[] , troncon troncons[] , int nombreLignes, int nombreTroncons)

```

void deplacerEtReinitialiserTicketsBus(bus flotteBus[],int nombreBus, ticket tickets[])

int calculerGainNet(bus flotteBus[], int nombreBus, ligne lignes[], int nombreLignes)

void afficherStatistiques(bus flotteBus[], int nombreBus, ligne lignes[], int nombreLignes)

void sauvegarderRapport(char* nomFichier, zone zones[], ligne lignes[], troncon troncons[], bus
flotteBus[], int nombreZones, int nombreLignes, int nombreTroncons, int nombreBus)

void deroulerSemaine(casecarte cartegeo[20][20], ticket tickets[], zone zones[], ligne lignes[], troncon
troncons[], bus flotteBus[], int nombreZones, int nombreLignes, int nombreTroncons, int nombreBus)

int main() ou void main()

```

### 3) Autres fichiers demandés

- Carte géographique tracée à la main et photographiée "jpg"(voir remarques section 7).
- 4 fichiers texte zones.txt, lignes.txt, troncons.txt et flottebus.txt remplies à l'aide de bloc-notes ou équivalent (voir remarques section 7).

### 4) Enoncé et vidéos explicatives enregistrées

Voici ci-dessous les vidéos dont vous aurez besoins pour le suivi 2, en plus du cahier de charge :

- Cahier de charges du mini projet S2 2022/2023 Release 1 (25/03/2023) : <https://bit.ly/42ryZdw>
- Formation de base en langage c : <https://bit.ly/3ogRsaI>
- Cours programmation 1 officiel : <https://bit.ly/3aIc3N3>
- Vidéos explicatives sur la playlist youtube :  
[https://www.youtube.com/playlist?list=PLMVD45wSmrxyo4uENZHG1K\\_vc4BNTr79b](https://www.youtube.com/playlist?list=PLMVD45wSmrxyo4uENZHG1K_vc4BNTr79b)
- Lien vidéos explicatives téléchargeables : <https://bit.ly/3HPs1pd>
- Tutoriel montage vidéo en ligne Canva : <https://bit.ly/3Af3R1M>

### 5) Remarques, informations et liens utiles:

- Après avoir installé le codeblocks, exécuter les opérations suivantes (sinon il ne marchera pas) :
  - 1 - Entrez à : "Settings" -> "Compiler..." -> "Toolchain executables"
  - 2 - Puis dans le champ de saisie "Compiler's installation directory" : cliquez sur "Auto-detect" ce qui remplacera la ligne "C:\MinGW" par la ligne "C:\Program Files (x86)\CodeBlocks\MinGW".
- Si vous n'avez de code à déposer mais juste des ressources comme les fichiers textes, vous pouvez, dans le module main, juste mettre un main vide : `int main(){return 0;}`
- Pour pouvoir voir les extensions des fichiers dans windows 10, ouvrez n'importe quel répertoire, cliquez sur l'onglet le plus à droite "affichage" (des onglets en haut à gauche), puis à droite cochez la case "Extensions de noms de fichiers".
- Les modules de sauvegarde devront fonctionner en mode écriture avec écrasement de l'ancien contenu.
- On ne va pas utiliser les .h dans le mini projet, en effet, tout le code sera dans un seul fichier .c.

- Pour les boucles de numérotation horizontale et verticale de la carte, il est recommandé d'utiliser `printf("%-2d"` pour prendre en charge l'écriture de tous les nombres sur 2 caractères et le `'-'` pour l'alignement du nombre à gauche..

## 6) Divers

- Serveur de dépôt de travaux : uniquement sur <http://miniprojet.maktabati.tn>
- En cas de problème, réclamation ou question, contactez l'email suivant : [miniprojet.isetbz@gmail.com](mailto:miniprojet.isetbz@gmail.com)

## 7) Historique d'anciennes remarques, informations et liens utiles :

### a) Remarques importantes 9

- ok ■ Noter Bien : Le fichier de la carte géographique va être nommé "carte.jpg"
- ok ■ Noter Bien : Les fichiers de sauvegarde vont porter un nom de la forme "zones\_sauv.txt lignes\_sauv.txt"
- ok ■ Important : On va utiliser les fonctions `fscanf` pour la lecture des données des fichiers, et `fprintf` pour l'écriture.
- ok ■ Important : On va utiliser la fonction `feof` pour boucler sur le `fscanf` :  
`while( !feof(fic))`
- ok ■ Important : pour les valeurs aléatoires on va utiliser la fonction `rand` (ne pas oublier d'initialiser le `srand` à `time` dans le `main`, et bien sûr ne pas oublier les includes nécessaires) : pour se documenter et comprendre le fonctionnement : <https://www.shiksha.com/online-courses/articles/rand-and-srand-functions-in-c-programming/>
- Prévoir de sauvegarder et charger les zones et personnes en même temps dans le fichier "personnes\_sauv.txt"

### b) Remarques importantes 8

- Le code doit être exécutable pour être accepté dans la plateforme.
- Les bouts de code présentés dans les vidéos explicatives, représentent un brouillon de code dans l'objectif de comprendre l'idée, les mécanismes et les techniques à utiliser et n'est pas nécessairement un code valide qu'on peut utiliser sans le réviser, le compléter ou le corriger.
- Dans le module `main`, ajouter au menu précédent la possibilité de déroulement d'une semaine.

### c) Remarques importantes 7

- Vu la remarque précédente de non utilisation des bibliothèques non standard spécifiques à windows comme `windows.h` et `conio.h`, et étant donné que `getch()` appartient à la bibliothèque non standard `conio.h` et que `getc()` et `getchar()` ne sont pas des équivalents à `getch()`, donc on va utiliser pour l'instant la fonction `scanf` de la bibliothèque `stdio.h`.
- L'avancement dans le déroulement d'une semaine est toujours minute par minute même en cas de vitesses supérieures à 1 minute : dans le cas de 10 minutes par exemple on boucle et on exécute "tout le code de déplacement et

autre" minute par minute comme d'habitude, le seul changement est que l'affichage ne se fait qu'une seule fois par 10 minutes de façon qu'on a l'impression que le déplacement a avancé de 10 minutes en une seule fois, et ainsi de suite pour les autres vitesses.

- Le paramètre nombreBus dans le module de numTacheSuivante et le module deplacerBus ne seront pas utilisés pour l'instant, car dans les deux cas on a déjà la case du bus qu'on va traiter idBusCourant.

#### d) Remarques importantes 6

- Compléter les tâches des bus dans le fichier flotteBus pour couvrir toute la journée, et on pourra appliquer cet horaire à toutes les journées.
- Faire une petite recherche sur la manière d'afficher la numérotation à gauche de la grille sur deux caractères même si le numéro est composé d'un seul chiffre, pour ne pas avoir des décalages dans la carte , par exemple :

```
9|      ||
10|     ||
```

Et non pas :

```
9|      ||
10|     ||
```

- Agrandir la taille de la fenêtre d'exécution : cliquer sur l'icône dans le coin haut à gauche de la fenêtre, puis sur "Propriétés" puis l'onglet "Configuration" puis mettre 167 comme largeur et 44 comme hauteur pour un écran 15 pouces, ou autres valeurs qui s'adaptent avec la taille de votre écran. Enfin valider et fermer la fenêtre d'exécution et la relancer.

#### e) Remarques importantes 5

- Réponse à la question : pourquoi mon code compile sur codeblocks et non pas sur le serveur miniprojet :
  - Le compilateur du serveur miniprojet est le gcc sur linux, tandis que codeblocks dont le lien sur le groupe facebook utilise mingw qui est la version windows du compilateur gcc, donc le compilateur est le même, c'est quoi le problème alors?
  - Le premier problème peut provenir de l'utilisation des bibliothèques .h qui sont propres à windows ou bien qui ne sont pas standards comme windows.h, conio.h ou quelques fonctions qui sont devenu obsolètes comme itoa et équivalents... **La solution** est simple est de ne pas utiliser ces bibliothèques et fonctions cités, vu aussi qu'on n'a pas besoin vraiment de les utiliser.
  - Le deuxième et plus grand problème est que code blocks accepte un code qui utilise par exemple printf et qui n'inclut pas stdio.h, c'est comme il fait des recherches des bibliothèques et des includes automatiques en arrière-plan, c'est-à-dire ce code qui a des includes manquants ne sera pas compilé sur aucun autre compilateur et ne sera pas un code obéissant aux standards. **Solution** : Compléter les includes nécessaires pour les fonctions non reconnues par le compilateur du serveur.

#### f) Remarques importantes 4

- C'est très important de lire tous les documents qui sont publiés dans le groupe facebook (une copie est publiée sur l'UVT), pour ne pas avoir à travailler des codes qui ne sont pas demandés ou d'une façon qui ne correspond pas à ce qui est demandé.
- Ne pas utiliser la bibliothèque conio.h et la fonction itoa et ses semblables, elles ne sont plus parmi les standards. En plus, on n'en a pas besoin.
- Ne pas utiliser windows.h et SetConsoleOutputCP(CP\_UTF8), concernant les codes runes veuillez consulter les documents de remarques précédents pour plus de détails.
- Dans les modules d'affichage et de sauvegarde de personnes : ne pas oublier les activités.
- Dans les modules de sauvegarde : mettre les données seulement sans aucun message.
- Le remplissage total inclut les différents chargements et la génération de personnes.
- Le dépôt de la carte et des fichiers .txt peut se faire tout seul ou avec du code, si le code est non compilable le code ne sera pas déposé mais le dépôt de la carte et des fichiers .txt va se faire dans tous les cas s'ils ne dépassent pas les limites de tailles bien sûr.
- Rappel sur la non-utilisation des variables globale, sauf la déclaration des constantes et des types d'enregistrement va être globale.

### **g) Remarques importantes 3**

- Le sens et ligne dans les activités des personnes à initialiser à 0 dans le pour le moment, à modifier ultérieurement.
- Les noms des fichiers de sauvegarde seront zonesSauv.txt, lignesSauv.txt, tronconsSauv.txt, flotteBusSauv.txt, personnesSauv.txt
- Dans le main, à part la déclaration des tableaux, on commence par le chargement puis on affiche un petit menu qui nous permet de choisir ou bien l'affichage ou bien la sauvegarde.
- Tout ce qui se rapporte au ticket n'est pas demandé pour le moment.

### **h) Remarques importantes 2**

- La numérotation des lignes et colonnes de la carte géographique commencent par 0.
- La carte géographique doit contenir au moins un rond-point.
- Chaque Bus doit avoir 4 tâches au minimum.

- Le chargement doit se faire à l'aide d'une boucle while ou do-while car le programme ne connaît pas le nombre de lignes des différents fichiers.
- Le chargement concerne la lecture des données depuis les fichiers txt et le remplissage des tableaux correspondants en utilisant la fonction fscanf.
- L'affichage concerne les champs des enregistrements dans les tableaux et non pas les fichiers.
- La sauvegarde concerne l'écriture des données depuis les tableaux vers les fichiers txt correspondants en utilisant la fonction fprintf.

## i) Remarques importantes 1

- Les procédures d'enregistrement utilisent des fichiers **différents** de ceux utilisés pour le chargement (zones.txt, lignes.txt, troncons.txt et flottebus.txt).
- Les **enregistrements** à déclarer juste **après les includes**, tandis que les **tableaux** sont à déclarer **dans le main** et à passer en paramètres des modules appelés qui en ont besoin.
- Pour prendre des photos des cartes géographiques à utiliser l'application **clearscanner** ou équivalent pour les smartphones android pour avoir des photos de taille moins que 200KO.
- Pour afficher les caractères spéciaux pas besoin de la bibliothèque windows.h qui ne sera pas accepté par le compilateur du site de dépôt mais il suffit juste de procéder comme l'exemple suivant : **printf("%c", 205) ;**
- Si les caractères spéciaux ne s'affiche pas encore après l'application de la remarque précédente, c'est qu'il y a un problème dans la configuration du terminal sur lequel se lance l'exécutable (à en informer l'enseignant dans les séances d'encadrement).
- Lien du logiciel codeblocks que l'on utilisera pour la matière miniprojet et pour la programmation c d'une façon générale : <https://sourceforge.net/projects/codeblocks/files/Binaries/20.03/Windows/codeblocks-20.03mingw-setup.exe> .
- Après avoir installé le codeblocks, exécuter les opérations suivantes (sinon il ne marchera pas) :
  - 1 - Entrez à : "Settings" -> "Compiler..." -> "Toolchain executables"
  - 2 - Puis dans le champ de saisie "Compiler's installation directory" : cliquez sur "Auto-detect" ce qui remplacera la ligne "C:\MinGW" par la ligne "C:\Program Files (x86)\CodeBlocks\MinGW".