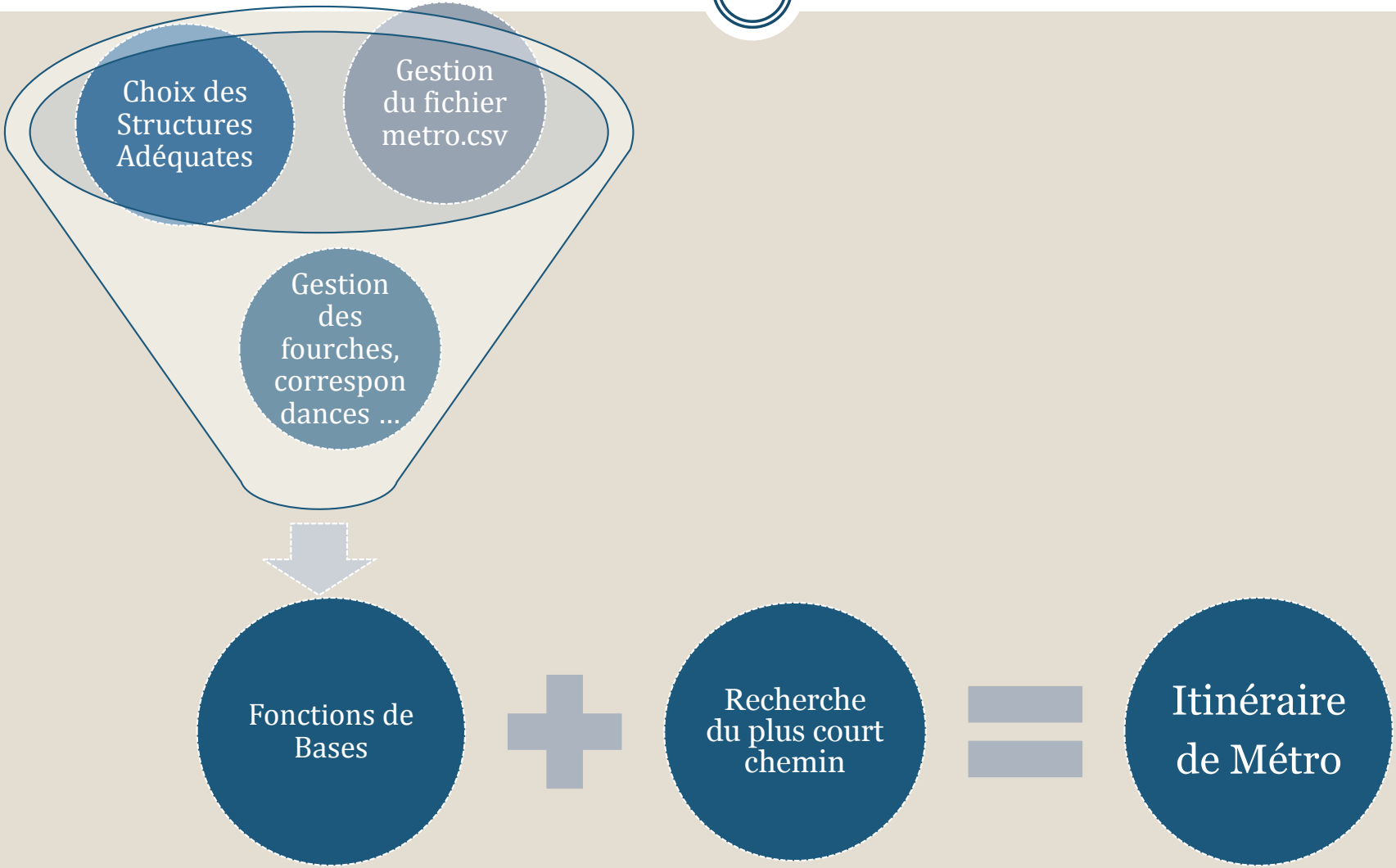
Itinéraire de métro

2016-2017

KONATE Moussa
Gueye Serigne Abdou Khadre
Télécom 1, Sup-Galilée, UP13

Enjeux du Projet



Organisation

Numéro Ligne	Temps : Monceau-Courcelles	Nom : Monceau	Heure Premier Train	Heures Minutes Secondes
Première Station		Préc : Villiers	Heure Dernier Train	
Dernière Station	Temps : Monceau-Villiers	Suiv : Courcelles		
		Poids		

Structure de Données : Listes doublements chainées

Fonctions de Base


```
void printStation(STATION* station);
void printLigne(LIGNE * maLigne);
void initialiseTemps(LIGNE maLigne);
void appendToLigne(LIGNE *ligne, STATION nouvelleStation);
void loadStationInLigne(LIGNE *pLigne, FILE *pFile, char *ligneStart, char
*ligneEnd);
void insert(LIGNE *pLigne, STATION *station);
void shortestPath(LIGNE metro[], LIGNE reserve, STATION *stationDepart,
STATION *stationDarrivee);
int getSizeBeetweenStartAndEnd(FILE *pFile, char *ligneStart, char *ligneEnd);
int nombreDelement(LIGNE *ligne);
int listeVide(LIGNE* maLigne);
int convertHeureToMin(HORAIRE time);
STATION *getStation(FILE *file);
STATION *removeTailFromList(LIGNE *maLigne);
STATION *removeHeadFromList(LIGNE *maLigne);
STATION *removeStationX(LIGNE *ligne, char* nomStation);
LIGNE reserve(FILE *file);
```

Implémentation de l'algorithme de Djikstra


```
void shortestPath(LIGNE metro[], LIGNE Reserve, STATION *stationDepart, STATION *stationDarrivee)
{
 while (varParcour != NULL)  {
   varParcour->parent = NULL; varParcour->poids = 2000; varParcour = varParcour->suiv;
 }
 stationDepart = removeStationX(&Reserve, stationDepart->nomStation);
 while ((listeVide(&Reserve) != 1) && arrive) {
   appendToLigne(&itineraire, *varParcour1Reserve);
   if (!strcmp(stationDarrivee->nomStation, varParcour1Reserve->nomStation)) {
    STATION*varParcourAffichage = itineraire.derniereStation; arrive = 0;
     while (varParcourAffichage != NULL && varParcourAffichage->poids > 0.0)  {
      printStation(varParcourAffichage); printf("->"); varParcourAffichage = varParcourAffichage->parent;
    }
    for (i = 0; i < NombreLigne; i++) {
      while (varParcour1metro != metro[i].derniereStation)

        while (varParcour2Reserve != NULL) {
           if (!strcmp(varParcour2Reserve->nomStation, varParcour1metro->suiv->nomStation)) {
                varParcour2Reserve->poids = varParcour1Reserve->poids + varParcour2Reserve->poids;
             if (nouveauPoids < varParcour2Reserve->poids) {
                 toRemove2 = removeStationX(&Reserve, varParcour2Reserve->nomStation);
                 toRemove2->poids = nouveauPoids; toRemove2->parent = varParcour1Reserve;
                 insert(&Reserve, *toRemove);
              }
           }
        varParcour2Reserve = varParcour2Reserve->suiv;
        }
       varParcour1metro = varParcour1metro->suiv;
      }
    }
    varParcour1metro = removeHeadFromList(&Reserve); insert(&itineraire, *varParcour1Reserve);
 }
}
```

Conclusion

- Choix Structures
- Gestion du fichier
  - Reste les cas particuliers à gérer
- Fonctions de bases
- Djikstra :