

## Rapport TicketToRide:

Je n'ai malheureusement pas réussi à coder un bot fonctionnel.

J'ai tenté deux approches différentes:

### APPROCHE 1:

Cette approche est basée sur l'utilisation de l'algorithme Djisktra suivie de deux fonctions permettant de stocker et récupérer toutes les informations des chemins à prendre. En fonction de ces chemins stockés dans un tableau `t_track Tracks`, `choosemove` prend la décision de `drawCard`, `drawBlindcard` ou `claimRoute` si les cards stockés dans la structure `t_player` conviennent.

```
if(You.nbObjectives!=0)//if the player has objectives we run the algorithm
{
    //we use nbObjectives, (we decrease the nbObjectives when one is taken)

    algorithme(obj[You.nbObjectives-1].city1,nbCities,obj[You.nbObjectives-1].city2,G,D,Prec);//Djisktra's Algorithm

    //We store the tracks we need to fulfill the objectives
    displayway2(obj[You.nbObjectives-1].city1,obj[You.nbObjectives-1].city2,nbCities,Prec,Tracks);

    //we retrieve the infos about the tracks (length, color, taken)
    trackstotake2(nbCities,Tracks,obj[You.nbObjectives-1].city2,obj[You.nbObjectives-1].city1,nbTracks,AllTracks);

    //we use the infos about the tracks to choose between drawCard, drawBlindcard or claimRoute
    move.type=choosemove2(&You,nbCities,Tracks,&move,G,faceUp);
}
```

### J'explique mon raisonnement:

- Si c'est la première fois que le joueur joue, **firstplay = 0**, et donc le premier coup joué sera obligatoirement un **move.type=4 (drawObjectives)**. Une fois ce premier coup joué, `firstplay = 1`.
- Si on a `move.type=4` alors le prochain coup sera **move.type=5 (chooseObjectives)**. Mon programme choisit de prendre les 3 objectifs proposés et de les stocker dans un tableau de `t_objective obj[10]`.
- Je crée une structure `t_player You` où je stocke le nombre d'objectif: **nbObjectives=3**.
- Si on a plus d'objectifs que 0, on applique l'algorithme de Djisktra (fonction `algorithm`) pour trouver le chemin le plus court, suivie de `displayway` et `trackstotake` qui permettent de récupérer les informations des chemins à prendre dans `t_track Tracks`.
- On utilise les informations de `Tracks` pour choisir entre `drawBlindcard`, `drawCard` et `claimRoute` (ainsi que les arguments de chacun de ces choix) dans `choosemove`.

- Si on a le nombre de cartes de la bonne couleur d'un des chemins que nous souhaitons prendre nous ferons un claimroute, avec le choix entre la première couleur, la seconde couleur si elle existe ou multicolore (locomotives).
- Si on a pas assez de cartes de la bonne couleur alors nous regardons si dans les cartes faceUp il y a une carte de la couleur d'un des chemins que nous souhaitons récupérer sinon nous prenons une carte face cachée.
- A chaque coup de l'adversaire et coup du player, on met à jour le tableau de cartes faceUp et si une route a été prise, les informations de AllTracks qui sont réutilisées à chaque coup. (AllTracks est un tableau de t\_track contenant toutes les informations des routes du jeu (initialisé en début de partie)).
- De plus pour empêcher le choix de changer avant d'être envoyé au serveur, un booléen choose=0 ou 1 bloque la décision prise.

### **Problèmes rencontrés:**

Mon problème s'est posé dès l'algorithme Djisktra, et après l'avoir réécrit plusieurs fois, utiliser des pointeurs, des tableaux... Je ne comprends pas pourquoi il fonctionne de façon si aléatoirement. Valgrind me dit qu'il y a un problème d'initialisation, pourtant je n'utilise que des tableaux de int que j'initialise justement avec un memset.

Au début, l'algorithme ne rentrait jamais dans la boucle qui remplit le tableau Prec[i] puis le problème inverse s'est posé, la boucle devenait infinie.

Actuellement l'algorithme marche de manière très aléatoire, parfois il fait directement une segmentation fault, parfois il part dans une boucle infinie dans le while(u!=dest) et parfois il remplit correctement Prec[i]...

Lorsque Prec[i] n'est pas rempli, le reste des fonctions ne peut plus obtenir de résultat correct et finissent par provoquer des segmentation fault ou des boucles infinies.

### **APPROCHE 2:**

J'ai voulu essayer d'implémenter la dernière fonction choosemove pour créer un bot qui poserait les chemins qu'il rencontre libre (sans passer par un tri algorithmique).

J'ai bien vérifié qu'Alltracks (tableau de t\_track initialisé dans fillarray et qui contient l'entière des chemins du jeu) était correctement rempli.

Mais mon bot après avoir pioché à plusieurs reprises, donne de mauvaises informations pour le claimRoute, pourtant je lui donne les informations d'AllTracks...

Autre fait que je ne comprends pas, AllTracks étant un tableau de t\_tracks, il semble logique de l'utiliser avec un indice (AllTracks[i].city1) par exemple. Pourtant j'ai l'impression que peu importe la valeur de i, il renvoie la même information pour le claimRoute.

Je ne comprends pas pourquoi il n'y arrive pas malgré ça.