

Compte-rendu des séances

COMPTE-RENDU DE CHAQUE SÉANCE

Note : J'ai forcément oublié des éléments que vous avez réalisés durant les séances, car je faisais autre chose. N'hésitez pas à me le dire.

1 TP ELEC PROJET

1.1 Séance 1 (16/10)

- Réflexion sur la construction du robot :
 - Placement des moteurs, des capteurs.
 - Distances entre les éléments.
 - Nombre de capteurs nécessaires.
- Réflexion sur les stratégies de déplacement :
 - Méthodes pour prendre des virages et avancer efficacement.
 - Solutions retenues :
 - * Utilisation de 4 capteurs : 3 à l'avant (1 centré, 1 à gauche, 1 à droite) et 1 à l'arrière (centré avec celui à l'avant).
 - * Objectif : sortir du labyrinthe le plus vite possible.
- Méthode de virage :
 - Le robot continue d'avancer tout en tournant en courbe.
 - Le capteur arrière détecte la ligne sur laquelle il était avant le virage.
 - Une fois la ligne détectée, le robot se recentre pour finir le virage.
 - En cas de problème, une rotation classique de 90° pourra être utilisée.

1.2 Séance 2 (06/11)

- **Louis** : Organisation et branchement sur la carte Raspberry (à l'exception des moteurs). **push** du schéma Fritzing.
- **Yohann/Maël** : Découpage d'un modèle de flamme imprimé, collage sur le robot, perçage et peinture des murs en bleu et rouge.
- **Nathan/Dylan** :
 - Clonage du dépôt Git sur la carte.
 - Séparation des fichiers et dossiers.
 - Écriture d'un algorithme pour les déplacements du robot (`parcours_laby.c`).
 - Simplification de l'analyse descendante (fusion de fonctions redondantes comme `RotationGauche` et `RotationDroite`).
 - Implémentation de l'arrêt d'urgence : vitesse des moteurs mise à 0 et extinction.
- Détails du fonctionnement du robot :
 - **Se remettre droit** : Si un capteur latéral détecte une ligne mais pas les autres.
 - **Virages à gauche/droite** : Détection par le capteur central et un capteur latéral.
 - **Avancer** : Avancer à vitesse modérée pour limiter les arrêts.
 - **Sortie du labyrinthe** : Si aucun capteur ne détecte de ligne, le robot coupe les moteurs.
- Tests des moteurs et des roues, impression des motifs du robot.

1.3 Séance 3 (20/11)

- **Louis/Yohann** : Branchement des moteurs, montage du robot (fixation des éléments, soudures, etc.).
- **Dylan/Nathan/Maël** : Écriture du code en C pour les moteurs, l'écran LCD, le signal sonore, et le buzzer.
- Discussion sur l'arrêt d'urgence en virage.
- Préparation pour la séance suivante :
 - Finaliser le code pour que tout fonctionne.
 - Compiler et tester le parcours du labyrinthe avec les threads.

1.4 Séance 4 (04/12)

- Objectifs : Vérification des fonctions, tests sur le robot.

2 TD ALGO PROJET

2.1 Séance 1 (22/10)

- **Objectif** : Mise en place du projet GitLab et spécification du TAD Labyrinthe.
- **Travail réalisé** : FAIT avant la séance 2.

2.2 Séance 2 (05/11)

- **Objectif** : Analyse descendante des opérations suivantes :
 - Labyrinthe \rightarrow Liste<NaturelNonNul>
 - Liste<NaturelNonNul> \rightarrow Liste<Ordre>
- **Travail réalisé** : FAIT avant la séance 3.

2.3 Séance 3 (13/11)

- **Objectif** : Conception préliminaire et détaillée.
- **Travail réalisé** :
 - **Louis** : CP + CD Trouver le plus court chemin, écriture du TAD Noeud.
 - **Dylan/Nathan** : CP + CD Conversion des noeuds en ordres.
 - **Maël** : CP + CD Analyse de fichier et macros.
 - **Yohann** : CP + CD Création de passage.

2.4 Séance 4 (19/11)

- **Objectif** : Développement des fichiers .h, des fichiers .c vides, et des tests unitaires.
- **Travail réalisé** :
 - Clarification de l'analyse descendante et des principes des algorithmes.
 - Mise en ordre des conventions de codage (nommage des fonctions, accolades, etc.).
 - Discussions sur le TAD Noeud : suppression et modifications nécessaires.
 - Organisation des bibliothèques et fichiers.

2.5 Séance 5 (26/11)

- **Objectif** : Développement des fichiers .c.

2.6 Séance 6 (03/12)

- Objectifs et travail à définir.