Prosit Les bonnes ondes

…

[09/11/22 – ../../22]

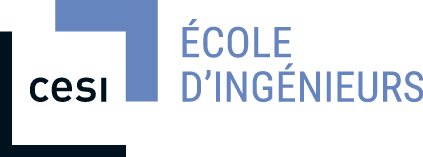


Table des matières

[1. Analyse du contexte 2](#_Toc83824561)

[2. Mots clés 2](#_Toc83824562)

[3. Problématiques 2](#_Toc83824563)

[4. Contraintes 2](#_Toc83824564)

[5. Livrables 2](#_Toc83824565)

[6. Généralisation 2](#_Toc83824566)

[7. Pistes de solution 2](#_Toc83824567)

[8. Plan d’action 2](#_Toc83824568)

[9. Réalisation du plan d’action 2](#_Toc83824569)

[10. Conclusion et retour sur les objectifs 2](#_Toc83824570)

[11. Bilan et synthèse du travail effectué 2](#_Toc83824571)

[12. Réferences bibliographiques 2](#_Toc83824572)

[12.1. Prosit 2](#_Toc83824573)

[12.2. Complémentaires 2](#_Toc83824574)

# Analyse du contexte

On travaille sur un projet de simulation informatique qui permet de reproduire le comportement d’une communication multiplexée (radio FM).

# Mots clés

Multiplexage :

Espace hertzien :

OS :

FM :

Canaux :

Buffer :

Mettre en pool :

Destructeur :

Bloc d’allocation discontinu :

Pgm evt :

Asynchrone :

# Problématiques

Comment régler les erreurs de la mise en place d’un système d’écoute asynchrone ?

Comment gérer un grand nombre d’utilisateurs ?

# Contraintes

Code donné

Classe réceptrice ne connait pas la classe émettrice

Utilisateur doit pouvoir s’abonner à plusieurs canaux

Grand nombres d’utilisateurs

# Livrables

Diagramme de classe

Code C++

# Généralisation

Comprendre les pointeurs de fonctions

Savoir la différence entre programme séquentiel et évènementiel

# Pistes de solution

Utiliser le destructeur pour supprimer un nombre d’objets pour garder de la mémoire

Utiliser des conteneurs pour pouvoir créer un grand nombre d’objets

Utiliser des listes chaînées

Héritage entre classe station

Chaque station correspond à une méthode de l’émetteur, appelée par le récepteur

Création de variables globales dans la classe émetteur

Simulation de l’envoi du signal dans un fichier.txt qui simule le réseau hertzien

Dans chaque récepteur mettre un tableau qui contient tous les émetteurs

# Plan d’action

Faire le diagramme de classe :)

Implémenter les classes en C++

Faire communiquer les classes entre elles (Tester les pistes de solutions)

Gérer le grand nombre d’utilisateurs

# Réalisation du plan d’action

# Conclusion et retour sur les objectifs

# Bilan et synthèse du travail effectué

# Réferences bibliographiques

## Prosit

## Complémentaires