

Projet C++ embarqué

Drone militaire

Contexte :

Nous sommes en 2024, 3^{ème} guerre mondiale. Un de nos drones a photographié une base militaire ennemie de nuit (voir annexe, fichier DroneIMG37337.png) et dissimulé les coordonnées du lieu (37°42'41.9"S 144°59'33.0"E). Cette image doit être récupérée et analysée pour déterminer les actions à mener.

Objectifs :

1. Transmettre l'image et son information cachée en utilisant de la stéganographie et le protocole mqtt
2. Récupérer l'image et découvrir les coordonnées du cliché

Organisation :

- Pour ce projet en groupe, vous serez séparés en deux équipes (équipe Drone et équipe Base). Vous allez partager un nouveau GitHub public et publier votre code et captures d'écran, sur ce repo en séparant les deux projets (Drone et Base).
- Vous proposerez des *pull requests*, *branches*, *commits* etc entre les membres de l'équipe, ou du groupe si vous êtes seul dans l'équipe.
- Un ReadMe sera disponible dans chaque projet, résumant l'utilisation du programme.
- Le PAQL en annexe devra être respecté.
- Aucun code commenté ne sera commité dans la branche principale release/main

Equipe Drone :

Vous aurez pour mission de créer un programme **console** avec Qt en C++ qui charge le fichier ***DroneIMG37337.png*** et y ajoute via un algorithme au choix autre que steghide, les coordonnées GPS ***37°42'41.9"S 144°59'33.0"E***.

Le fichier sera envoyé sur le broker : `mqtt://broker.emqx.io:1883`, topic `/ynov/bordeaux/NomDuGitHubDuGroupe` avec un QoS 2.

Le programme devra être le plus léger possible, chaque tentative d'optimisation faisant l'objet de la création d'une branche dans le GitHub.

Equipe Base :

Vous aurez pour mission de créer un programme graphique avec QtDesign en C++ qui se connecte au broker : `mqtt://broker.emqx.io:1883`, et s'abonne au topic `/ynov/bordeaux/NomDuGitHubDuGroupe` avec un QoS 2.

Une fois le fichier récupéré, vous devrez découvrir les coordonnées cachées dans l'image et afficher l'image, ainsi que les coordonnées GPS.

Le programme devra être le plus léger possible, chaque tentative d'optimisation faisant l'objet de la création d'une branche dans le GitHub.

Barème :

Barème par groupe		
Sous-partie	Détails	Note
Fonctionnement de la chaîne de transmission globale	<ul style="list-style-type: none">• Envoi de l'image avec coordonnées• Réception du fichier et décodage des coordonnées	/4
Participation active au GitHub	<ul style="list-style-type: none">• Chacun participe• Création de logo• Création de projets• Follow du compte	/2
Barème par équipe		
Sous-partie	Détails	Note
Echanges entre développeurs	<ul style="list-style-type: none">• Branche release• Branches RC (release candidate)• Commentaires dans les commits	/2
Répartition des tâches au sein d'une équipe	<ul style="list-style-type: none">• Partie mqtt• Partie stégano• Corrections• Tests• Captures d'écran• Optimisation (chacun doit essayer)	/2
Barème individuel		
Sous-partie	Détails	Note
Respect du PAQL	<ul style="list-style-type: none">• Préfix des variables• Définition de nouveaux types• CamelCase• Indentation et accolades	/4
Qualité du code	<ul style="list-style-type: none">• Création de classes• Méthodes• Construction/ destruction• Fuites mémoire• Variables non initialisées• Code lisible	/4
Optimisation du code	<ul style="list-style-type: none">• Intérêt des modifications apportées• Taille du binaire	/2
Total		/20