

MODULE 06

SÉANCE SYSTÈME 08

TP D'INFORMATIQUE

Durée 2h30

INTERFACE GRAPHIQUE : SUPERVISION DU PARKING EN TEMPS RÉEL

BLOC DE COMPÉTENCES

U6 - VALORISATION DE LA DONNÉE ET CYBERSÉCURITÉ

COMPÉTENCE(S)

C08 - CODER

OBJECTIFS PÉDAGOGIQUES

Codage d'une interface graphique munie de curseurs (scrollbar) permettant de piloter des appareils d'éclairage de scène.

CONNAISSANCES ISSUES DU RÉFÉRENTIEL

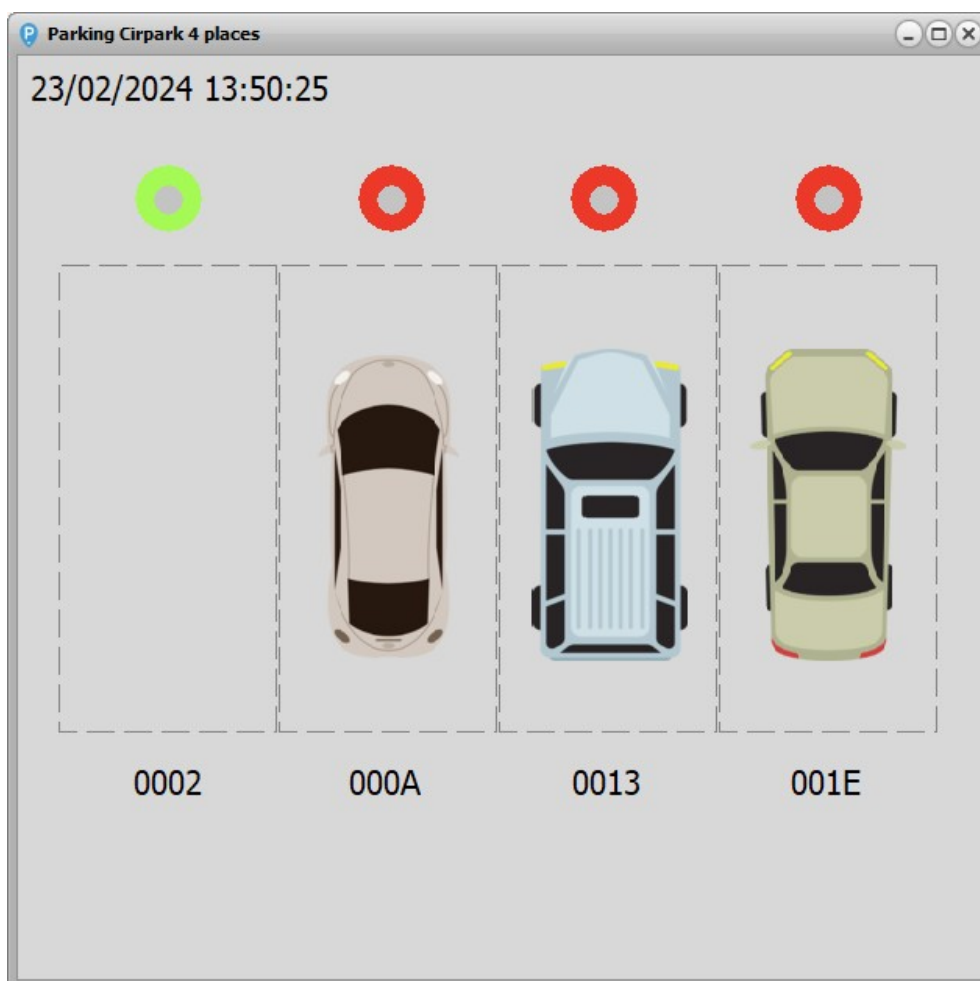
- | | |
|--|----------|
| • Langages de développement, de description, de création d'API et les IDE associés | Niveau 3 |
| • Programmation multitâche | Niveau 3 |
| • Programmation orientée objet | Niveau 3 |

CONNAISSANCES OPÉRATIONNALISÉES

- | | |
|--|----------|
| • Concevoir une interface graphique sous Windows | Niveau 2 |
| • Codage d'un thread en C++ | Niveau 2 |
| • Versionner un code | Niveau 2 |

TP

Nouvelle fiche : vue du parking



Conception de la vue du parking

Ajouter une nouvelle fenêtre de conception qui contiendra la vue du parking : Fichier -> nouveau -> Fiche VCL. Une nouvelle classe TForm3 est créée avec son .h (unit3,h) et son .cpp (unit3,cpp). Inclure unit3.h dans unit1.cpp et unit1,h dans unit3.cpp.

Ajouter dans la partie conception de TForm3 les objets graphiques permettant de visualiser la parking de 4 places :

- 2x4 TShapes ronds pour la représentation des capteurs
- 4 TShapes rectangulaires en pointillé pour les emplacements
- 4 TImages pour les véhicules : y charger 4 des images de véhicules fournies dans le répertoire Ressources.
- 4 TLabel pour les numéros des capteurs
- 1 TLabel pour la date et l'heure

Passage de la fenêtre de recherche des capteurs à la vue du parking

Ajouter le bouton « Vue parking » dans la fenêtre de recherche des capteurs. Dans l'événement OnClick de ce bouton, appeler la méthode Show() de l'objet Form3.

Lors du click sur l'un des quatre numéros de capteur de la vue parking (unit3.cpp), recopier le Caption du Label dans le Text du Combobox de la fenêtre de recherche. Par exemple :

```
Form1->ComboBox1->Text=Label1->Caption;
```

Appeler alors l'événement OnClick du chargement de la configuration d'un capteur : Par exemple :

```
Form1->Button2Click(this).
```

Appeler enfin la méthode Hide().

L'interface permet maintenant de rechercher les capteurs puis de visualiser le parking. Dans la vue parking, un clic sur le numéro d'un capteur permet de visualiser sa configuration.

TP

Actualisation de la vue parking en « temps réel » : timer de scrutation de l'état des capteurs

Un timer est un thread qui s'exécute selon une cadence prédéfinie.

Affichage de la date et de l'heure

Ajouter un TTimer (désactivé au démarrage) dans la fenêtre de conception de la vue parking. Dans l'événement OnClick du bouton « Configuration » de la fenêtre de recherche des capteurs : activer le timer de la vue parking.

Dans l'événement OnTimer (qu'il faut générer en double-cliquant sur l'objet graphique TTimer), affecter au caption du label date et heure : Date()+Time(). Tester et vérifier la présence et la mise à jour de la date et de l'heure.

Envoi de la requête « état du capteur » : place libre ou bien occupée

Quel code fonction permet de connaître l'état du capteur (libre ou occupée) ?

Quel est l'indice de l'octet de la réponse du capteur qui contient l'état du capteur ?

Quelle méthode de la classe IRSnifferCirpark permet de lire la réponse du capteur ?

Dans l'événement OnTimer, pour chaque label (numéro du capteur), envoyer la requête en donnant le numéro du capteur (utiliser ChaineHexVersInt()). Récupérer la réponse dans un pointeur sur char. Par exemple :

```
char * reponse=Form1->sniffer.Reponse();
```

Selon la valeur de l'octet « état » du tableau réponse, rendre l'image du véhicule visible ou non. De la même manière, modifier la couleur du Tschap correspondant au capteur.

Bonus : finalisation de l'application

Ajouter un icône à l'application et modifier l'apparence de l'application : dans les options du projet :

- Application, modifier l'apparence...
- Icône : choisir un icône (flaticon.com), convertir en icône 128x128 (site png to ico).

Rendre l'application indépendante de C++ Builder :

- Lieur : décocher Lier avec RTL dynamique,
- Packages d'exécution : décocher,

Versionner le code complet.

TP Bonus

Supervision d'un parking de plusieurs étages

- 1 à 28 places par étage -

