

PCO Laboratoire 4 - Télécabine

Auteurs: Arthur Bécaud et Bruno Egremy

Description des fonctionnalités du logiciel

Ce laboratoire 4 à pour objectif de nous familiariser avec les sémaphores vu durant le cours PCO. Le but de ce projet consiste à gérer une télécabine pouvant accueillir N skieurs pour monter ses passagers en haut d'une montagne puis redescendre pour répéter ce procédé.

Le logiciel possède ainsi deux fonctionnalités :

- Gestion du comportement d'une télécabine (necessitant aussi l'implémentation de la classe de télécabine `CableCar`).
- Gestion du comportement d'un skieur.

Choix d'implémentation

Comportement d'un skieur (`SkierBehavior`)

Le thread d'un skieur va chercher à monter la montagne avec une télécabine puis descendre de celle-ci dans un temps aléatoire. Une fois arriver en bas, le skieur répète ce procédé jusqu'à la fermeture de la télécabine.

```
void SkierBehavior::run()
{
    // Monte avec le télécabine et descend
    // tant que le télécabine est en service
    while (cableCar->isInService()) {
        cableCar->waitForCableCar(id);
        // Vérifie que le télécabine soit encore en service
        if (cableCar->isInService() == false) {
            break;
        }
        // Monte dans le télécabine puis attend d'arriver en haut de la montagne
        cableCar->goIn(id);
        cableCar->waitInsideCableCar(id);
        cableCar->goOut(id);
        // Descend de la montagne en un temps aléatoire
        goDownTheMountain();
    }
}
```

Comportement d'une télécabine (`CableCarBehavior`)

Une télécabine commence son cycle de vie en bas de la montagne et attend au minimum au skieur avant monter la montagne. Une fois en haut, la télécabine décharge ces passagers puis redescend de la montagne pour recommencer son cycle.

```
void CableCarBehavior::run()
{
    // Monte des skieurs en haut de la montagne
    // tant que la télécabine est en service
    while(cableCar->isInService()) {
        // Charge les skieurs dans le télécabine
        cableCar->loadSkiers();
        // si vide -> attente x secondes
        // Monte et décharge les skieurs en haut de la montagne
        cableCar->goUp();
        cableCar->unloadSkiers();
```

```

    // Redescend pour recommencer la boucle
    cableCar->goDown();
}
}

```

Implémentation du télécabine (CableCar)

La classe de la télécabine possède une section critique pour le compte de skieurs en attente en bas de la montagne. Nous devons protéger cette section car chaque threads (des skieurs ou de la télécabine) peut in/decrémenter sa valeur.

Avec cela cette classe **CableCar** possède deux barrières pour forcer l'attente des skieurs :

1. Barrière statique pour l'attente des skieurs en bas de la montagne.
2. Barrière pour l'attente des skieurs à l'intérieur de la télécabine (non-statique).

Implémentation des fonctions d'attente pour les skieurs

Chaque skieurs venant attendre la télécabine doit se mettre dans la file d'attente ce qui est représenté par une variable statique nommé `nbSkiersWaiting`. Celle-ci sera modifiée par chaque skieur et est donc une section critique qui doit être protégée par un mutex (implémenté à l'aide d'un sémaphore) nommé `mutexNbSkiersWaiting`.

Afin de faire attendre les skieurs à l'entrée de la télécabine ainsi qu'à l'intérieur de celle-ci durant le trajet, il convient d'utiliser des sémaphores initialisés à 0 afin de stopper les threads skieur. Ceux-ci sont uniquement libérés par la télécabine lorsque qu'elle est prête à charger ou décharger ses skieurs.

```

void PcoCableCar::waitForCableCar(int id) {
    // Vérifie si la télécabine est encore en service
    if(isInService() == false) {
        return;
    }

    // Increment du nombre de skieurs en attente
    mutexNbSkiersWaiting.acquire();
    nbSkiersWaiting++;
    mutexNbSkiersWaiting.release();

    qDebug() << "Skieur" << id << "attend en bas";

    // Fait attendre le skieur
    skiersWaitingOutside.acquire();
}

void PcoCableCar::waitInsideCableCar(int id) {
    skiersWaitingInside.acquire();
}

// Non utilisé
void PcoCableCar::goIn(int id) { }

// Non utilisé
void PcoCableCar::goOut(int id) { }

bool PcoCableCar::isInService() {
    return inService;
}

```

Implémentation des fonctions de (dé)charges des skieurs

Afin de ne pas partir à vide, la télécabine attend par cycle de 5 secondes l'arrivée de nouveaux skieurs.

Pour laisser les skieurs entrée dans la télécabine, celle-ci libère le sémaphore concerné (`skiersWaitingOutside`) pour chaque skieur dans la limite de la capacité maximale.

Lorsque que la télécabine laisse entrée les skieurs qui l'attendait, elle doit encore décrémenter le compteur `nbSkiersWaiting` et donc, doit utiliser le mutex correspondant. En effet, de nouveaux skieurs pourrait arriver et se mettre en attente au même moment.

La télécabine possède un compteur de passager nommé `nbSkiersInside`. Celui-ci ne nécessite pas de mutex car la télécabine est la seule à y accéder.

Pour laisser les skieurs sortir, le procédé est plus simple. Il suffit de libérer le sémaphore `skiersWaitingInside` tant que la cabine n'est pas vide. Il est possible de savoir si elle l'est ou non grâce au compteur `nbSkiersInside` qui est mise à jour à chaque entrée ou sortie de skieur dans la télécabine.

```
void PcoCableCar::loadSkiers() {
    // Pour ne pas partir à vide, la télécabine
    // attend 5 secondes après chaque vérification
    // du nombre de skieurs en attente
    while (nbSkiersWaiting == 0 && isInService()) {
        PcoThread::usleep(5000000);
    }

    // Rempli la télécabine tant qu'il y a de la place
    // et qu'il y a des skieurs en attente
    while (nbSkiersInside < capacity && nbSkiersWaiting != 0) {

        // Autorise un skieur à monter dans la télécabine
        skiersWaitingOutside.release();

        // Décremente le nombre de skieurs en attente (section critique)
        mutexNbSkiersWaiting.acquire();
        nbSkiersWaiting--;
        mutexNbSkiersWaiting.release();

        // Incrémente le nombre de skieurs dans la télécabine
        nbSkiersInside++;
    }
}

void PcoCableCar::unloadSkiers() {
    // Vide la télécabine tant qu'il y a des skieurs à l'intérieur
    while (nbSkiersInside != 0) {
        // Autorise un skieur à sortir de la télécabine
        skiersWaitingInside.release();
        // Décremente le nombre de skieur à l'intérieur de la télécabine
        nbSkiersInside--;
    }
}
```

Implémentation de la fin de service d'une Télécabine

Pour notifier les skieurs de la fermeture, il suffit de les libérer de l'attente en bas de la piste. Leur comportement, défini dans `SkierBehavior`, leur permettra de comprendre que la télécabine est fermée que leur journée est terminée.

```
void PcoCableCar::endService() {
    // Ferme le service du télécabine
    inService = false;

    // Débloque les skieurs en attente
    for (unsigned int i = 0; i < nbSkiersWaiting; i++) {
        skiersWaitingOutside.release();
    }
}
```

Tests effectués

Nous avons testé le fonctionnement du logiciel en l'exécutant à multiple reprises avec des paramètres différents (nombre de skieur et capacité de la télécabine). Puis en vérifiant manuellement la sortie du logiciel réalisé avec `qDebug()`.

Voici un exemple de sortie du logiciel avec les paramètres par défaut.

```
Démarrage de la simulation de skieurs et télécabine ...
Nombre de skieurs : 10
Capacité du télécabine : 5
Appuyez sur ENTER pour terminer

[START] Thread du skieur 4 lancé
[START] Thread du skieur 10 lancé
[START] Thread du skieur 7 lancé
[START] Thread du skieur 8 lancé
[START] Thread du télécabine lancé
[START] Thread du skieur 5 lancé
Skieur 10 attend en bas
[START] Thread du skieur 2 lancé
[START] Thread du skieur 1 lancé
Skieur 5 attend en bas
Skieur 4 attend en bas
Skieur 8 attend en bas
Skieur 7 attend en bas
[START] Thread du skieur 6 lancé
Telecabine monte avec 5 skieur(s), 0 attend(s) en bas
[START] Thread du skieur 9 lancé
Skieur 1 attend en bas
[START] Thread du skieur 3 lancé
Skieur 9 attend en bas
Skieur 6 attend en bas
Skieur 2 attend en bas
Skieur 3 attend en bas
Telecabine descend
Skieur 4 est en train de skier et descend de la montagne
Skieur 7 est en train de skier et descend de la montagne
Skieur 8 est en train de skier et descend de la montagne
Skieur 5 est en train de skier et descend de la montagne
Skieur 10 est en train de skier et descend de la montagne
Skieur 10 est arrivé en bas de la montagne
Skieur 10 attend en bas
Telecabine monte avec 5 skieur(s), 1 attend(s) en bas
Skieur 7 est arrivé en bas de la montagne
Skieur 7 attend en bas
Skieur 5 est arrivé en bas de la montagne
Skieur 5 attend en bas
Telecabine descend
Skieur 6 est en train de skier et descend de la montagne
Skieur 9 est en train de skier et descend de la montagne
Skieur 3 est en train de skier et descend de la montagne
Skieur 2 est en train de skier et descend de la montagne
Skieur 1 est en train de skier et descend de la montagne
Skieur 4 est arrivé en bas de la montagne
Skieur 4 attend en bas
Skieur 8 est arrivé en bas de la montagne
Skieur 8 attend en bas
Skieur 9 est arrivé en bas de la montagne
Skieur 9 attend en bas
Telecabine monte avec 5 skieur(s), 1 attend(s) en bas

Skieur 3 est arrivé en bas de la montagne
Skieur 1 est arrivé en bas de la montagne
Telecabine descend
Skieur 8 est en train de skier et descend de la montagne
Skieur 4 est en train de skier et descend de la montagne
Skieur 10 est en train de skier et descend de la montagne
Skieur 5 est en train de skier et descend de la montagne
Skieur 7 est en train de skier et descend de la montagne
Skieur 6 est arrivé en bas de la montagne
Skieur 7 est arrivé en bas de la montagne
Skieur 2 est arrivé en bas de la montagne
Skieur 4 est arrivé en bas de la montagne
[STOP] Thread du skieur 4 a terminé correctement
```

Skieur 8 est arrivé en bas de la montagne
Skieur 10 est arrivé en bas de la montagne
[STOP] Thread du skieur 10 a terminé correctement
[STOP] Thread du skieur 7 a terminé correctement
[STOP] Thread du skieur 8 a terminé correctement
[STOP] Thread du télécabine a terminé correctement
Skieur 5 est arrivé en bas de la montagne
[STOP] Thread du skieur 5 a terminé correctement
[STOP] Thread du skieur 2 a terminé correctement
[STOP] Thread du skieur 1 a terminé correctement
[STOP] Thread du skieur 6 a terminé correctement
[STOP] Thread du skieur 9 a terminé correctement
[STOP] Thread du skieur 3 a terminé correctement
Press <RETURN> to close this window...