

# Mini Projet IGI 641

(Séances TP 3 – 4 – 5 – 6)

## Bavardage et autres échanges

### Objectifs d'apprentissage :

Maîtriser le mécanisme de communication par événements, et s'initier aux interfaces graphiques.

### Problème proposé :

Nous souhaitons la construction d'une application style messagerie instantanée (en plus simple, bien sûr) que nous allons appeler ePapotage. Cette application doit permettre aux bavards de papoter avec un concierge (lui envoyer des messages), et d'autre part au concierge de répéter ce qu'il entend (messages reçus) à tous les autres bavards. Les utilisateurs de ce système sont donc un concierge, et plusieurs bavards, nous aurons en plus un bâtiment (gestionnaire) qui regroupe tout ce petit monde. Chaque bavard peut s'enregistrer auprès du concierge afin de poster des messages et de recevoir les messages d'autres bavards. La communication entre ces utilisateurs se fera en utilisant le mécanisme d'événements. Les bavards et le concierge échangeront des PapotageEvent. En réponse à la réception de ce message, le concierge le fait suivre à tous les bavards connectés.

### Travail demandé

Trois niveaux sont proposés pour ce sujet, correspondant à trois versions du système – « à l'essentiel », « fonctionnelle », « étendue »

#### Version « à l'essentiel »

1. Gestion d'événements
  1. Implémentez la classe PapotageEvent, sachant que les messages envoyés par les bavards ont un sujet, et un corps.
  2. Implémentez l'interface PapotageListener, associé à PapotageEvent.
  3. Donner la définition de la classe Bavard. Elle a un double rôle :
    1. source d'événements PapotageEvent - créateur de ce type d'événement
    2. écouteur d'événements PapotageEvent émis par d'autres bavards (reçus via le concierge).
  4. Donnez la définition la classe Concierge, qui agit en tant qu'intermédiaire des bavards :
    1. écouteur de messages de type PapotageEvent en provenance de bavards
    2. source de messages de type PapotageEvent – une fois qu'un événement de type PapotageEvent est reçu de la part d'un bavard, il est automatiquement transmis à tous les bavards ayant souscrit.
  5. Donnez la définition la classe Bâtiment, qui agit en tant que gestionnaire des bavards et du concierge :
    1. Gestionnaire de bavards
      1. Création du concierge
      2. Création de nouveaux bavards
      3. Permet aux bavards de se connecter et de manifester leur intérêt à recevoir les messages du concierge

2. Interfaces Graphiques Proposez un ensemble d'interfaces graphiques permettant à l'utilisateur d'accéder aux différentes fonctionnalités
  1. Une interface bâtiment permettant la création de bavards, et leur connexion
  2. Une interface bâtiment concierge permettant la visualisation des messages reçus
  3. Chaque bavard doit avoir une interface à sa disposition, lui permettant d'envoyer un message et d'accéder à travers une liste, au contenu de différents événements reçus (seule une version abrégée du message est visible dans la liste).

Vous êtes libres dans le choix d'éléments graphiques.

## **Version « fonctionnelle »**

Gestion des utilisateurs connectés : Il y a une gestion des utilisateurs en ligne qui s'ajoute. Quand un utilisateur se connecte, un message `OnLineBavardEvent` est envoyé aux bavards connectés. L'interface des Bavards est enrichie par l'affichage de la liste des utilisateurs connectés. Quand un utilisateur se déconnecte, un message `OffLineBavardEvent` informe les autres bavards.

## **Version « étendue »**

Vous pouvez ajouter des fonctionnalités supplémentaires. Voici quelques suggestions, mais vous êtes libres d'en proposer d'autres :

1. Filtre de messages : un utilisateur peut filtrer les informations en provenance de certains utilisateurs ;
2. Messages à thème : chaque bavard doit spécifier le thème de son message (sport, cinéma, etc.). Le système doit ainsi permettre aux bavards de ne recevoir que des messages dans des thèmes qu'ils auraient choisis.
3. Invitation pour discussion « en privé » (pour ceux qui sont très —en avance)