

Projet Programmation système

GROUPE 1

Détails et explications des design patterns

Groupe 1

12/12/2024

Afin de réaliser ce projet portant sur la réalisation d'une simulation de restaurant, nous avons utilisés les design patterns suivants :

- Observer
- MVC
- Factory

1. Design pattern : MVC

Celui-ci nous a permis de diviser les responsabilités de notre projet en 3 modules différents (modèle, vue, contrôleurs) et cela nous a facilité la gestion, la modularité et la réutilisabilité du code

- Le modèle : il gère les données, la logique métier et les règles de l'application. Il fournit un accès aux données, notifie les vues ou les contrôleurs lorsque les données changent. Il manipule les informations comme le nombre de client, le nombre de table, le nombre de carte et ainsi de suite.
- La vue : elle gère l'affichage des données à l'utilisateur. Elle reçoit des données venant du contrôleur ou du modèle et les affiche, elle écoute les événements du contrôleur pour mettre à jour l'interface utilisateur. Elle s'occupe d'afficher les entités comme les clients, les serveurs et les tables à l'écran et d'afficher leur changement de position.
- Contrôleur : il agit principalement comme un intermédiaire entre la vue et le modèle. Il reçoit les entrées de l'utilisateur depuis la vue ensuite il les transforme en actions applicables au modèle. Ici il récupère les paramètres entrés par l'utilisateur au lancement de l'application et notifie la vue des changements.

En gros ici on a le contrôleur qui récupère les données et mets à jour le modèle ensuite celui-ci notifie la vue des changements pour que celle-ci les affiche.

2. Observer :

C'est un design pattern comportemental que nous avons utilisé pour établir une relation entre des objets, où un sujet notifie automatiquement plusieurs ou un seul observateurs chaque fois qu'un événement ou un changement d'état se produit.

Ici le sujet (classe observable) conserve une liste de ses/son observer lorsqu'un changement survient chez celui-ci il va appeler sa méthode notify pour avertir tous les observer des changements et eux à leur tour vont appeler leur méthode update pour effectuer leurs changements. Ici on l'utilise surtout entre la vue et le modèle lorsqu'il y'a modification sur le modèle par exemple un changement de position d'un élément graphique celui-ci notifie la vue pour que ce changement se répercute de façon visuelle.

3. Factory

Le factory pattern ou pattern de fabrique est un pattern de création. Nous l'avons utiliser pour la logique de création d'objets, permettant ainsi au code client d'obtenir des objets sans avoir besoin de connaître les détails de leur instantiation. Nous l'avons utilisé pour créer les instances des tables dans le restaurant, les instances des couverts et des plats et aussi pour créer des instances de client tout au long du processus.