

Patrick Cloudy - Interaction Humain-Machine (IHM)

Ultime Patrick Cloudy de la saison, essayons de partir du bon pied

Concepts Clés et Explications

1. Gestionnaires de Placement en JavaFX

Concepts :

Les gestionnaires de placement (ou Layout Managers) sont essentiels pour organiser les éléments d'une interface graphique en JavaFX. Voici les principaux :

- **BorderPane** : Divise la scène en cinq régions : haut, bas, gauche, droite et centre.
 - Utilisation : Idéal pour des interfaces avec une structure fixe où certains éléments doivent rester aux mêmes endroits (ex. : une barre de menus en haut).
- **GridPane** : Organise les éléments en une grille avec des lignes et des colonnes.
 - Utilisation : Parfait pour les formulaires ou les interfaces nécessitant un alignement en grille.
- **HBox** : Aline les éléments horizontalement en une seule ligne.
 - Utilisation : Pour aligner des boutons ou des champs de texte horizontalement.
- **VBox** : Aline les éléments verticalement en une seule colonne.
 - Utilisation : Pour empiler des éléments les uns sur les autres, comme des étiquettes et des champs de texte.

Widgets : Les widgets sont les composants d'interface que vous placez dans les gestionnaires de placement.

- **Button** : Bouton cliquable pour déclencher des actions.
- **Canvas** : Surface pour dessiner des graphiques.
- **ListView** : Liste déroulante pour afficher une liste d'éléments.

Graphe de Scène :

C'est une représentation hiérarchique de tous les éléments d'une interface. Chaque nœud représente un composant de l'interface (ex. : un bouton, un conteneur HBox, etc.).

2. Gestion des Événements

Concepts :

La gestion des événements en JavaFX permet de répondre aux actions de l'utilisateur, telles que les clics de souris ou les pressions de touches.

- **MouseEvent** : Gère les événements de souris (clic, déplacement, etc.).
- **ActionEvent** : Gère les événements d'action, comme un clic sur un bouton.

Méthodes Utiles :

- `setOnAction(EventHandler<ActionEvent> event)` : Associe une action à un bouton.
- `setOnMouseClicked(EventHandler<MouseEvent> event)` : Associe une action à un clic de souris.

Exemples de Code :

1. Listener avec Lambda :

```
button.setOnAction(event -> {
    // Action à exécuter
});
```

2. Désactivation du Bouton :

```
button.setDisable(true); // Désactive le bouton
```

3. Détection de Clic sur la Cible :

```
canvas.setOnMouseClicked(event -> {
    if (currentTarget != null && currentTarget.check(event.getX(),
event.getY())) {
        clearAll();
        button.setDisable(false); // Réactive le bouton
    }
});
```

4. Ajout à ListView :

```
listview.getItems().add(currentTarget);
```

5. Highlight d'une Cible Sélectionnée :

```
listview.setOnMouseClicked(event -> {
    Target selectedTarget =
listview.getSelectionModel().getSelectedItem();
    if (selectedTarget != null) {
        clearAll();
        selectedTarget.highlight();
    }
});
```

3. Ergonomie des Interfaces

Concepts :

L'ergonomie des interfaces se concentre sur la création de systèmes utilisables et agréables pour l'utilisateur.

Critères Ergonomiques :

- **Visibilité** : Les options et fonctions doivent être facilement identifiables.
- **Feedback** : Le système doit informer l'utilisateur de ses actions et de l'état actuel.
- **Contraintes** : Limiter les actions possibles pour éviter les erreurs.
- **Consistance** : Les éléments similaires doivent fonctionner de manière cohérente.

Théorie de la Gestalt :

Les principes de la Gestalt expliquent comment les humains perçoivent les formes et les objets dans un contexte visuel.

- **Proximité** : Les éléments proches sont perçus comme appartenant à un même groupe.
 - **Similarité** : Les objets similaires sont perçus comme un groupe.
 - **Continuité** : Les éléments alignés sont perçus comme un ensemble continu.
 - **Clôture** : Les formes incomplètes sont perçues comme des formes complètes.
-

4. Keystroke-Level Model (KLM)

Concepts :

Le modèle KLM permet de prédire le temps nécessaire pour accomplir une tâche en décomposant les interactions en opérations élémentaires.

Opérateurs KLM :

- **K (Keystroking)** : Frappe de touche au clavier.
- **P (Pointing)** : Pointage avec la souris.
- **H (Homing)** : Déplacement de la main entre le clavier et la souris.
- **D (Drawing)** : Dessin ou traçage.
- **M (Mental)** : Activité mentale de l'utilisateur (décision, réflexion).
- **R (Response Time)** : Temps de réponse du système.

Exemple :

Pour une tâche impliquant la sélection d'un bouton et la saisie d'un texte :

1. P (pointer sur le bouton)
 2. K (cliquer sur le bouton)
 3. M (décider du texte à saisir)
 4. K (frapper les touches pour saisir le texte)
-

Application des Concepts

Gestion de l'Interface

- **BorderPane** : Utilisé pour une disposition fixe.
- **GridPane** : Utilisé pour des formulaires et des dispositions en grille.
- **HBox/VBox** : Utilisés pour des alignements horizontaux et verticaux simples.

Gestion des Événements

- Associer des actions spécifiques aux événements de l'utilisateur.
- Utiliser des lambda expressions pour une syntaxe plus concise.

Ergonomie et Théories

- Appliquer les principes ergonomiques pour améliorer la convivialité.
- Utiliser les lois de la Gestalt pour une perception visuelle optimale.

KLM

- Décomposer les tâches utilisateur pour évaluer et optimiser les temps d'exécution.

