UNIVERSITÉ de Picardie Jules Verne

Diaporama du Projet de Simulation de Ruche

Licence informatique 2 eme année

RAMDANI Rachel
HOCINE Abdelaziz

Projet de Simulation d'une Ruche

Présentation du projet de simulation d'une ruche, un exemple d'application de la programmation orientée objet pour simuler le comportement des abeilles dans un environnement virtuel.

Objectif du Projet

- Simuler le comportement des abeilles (employées, observatrices, éclaireuses) dans une ruche.
- Illustrer la collecte de nectar à partir de sources et le retour à la ruche.

Classe Principales

- **Abeilles :** Classe parente pour tous les types d'abeilles.
- **Employées, Observatrices, Éclaireuses :** Classes filles
- **Sources :** Classe représentant les sources de nectar.
- **Ruches :** Classe représentant la ruche où résident les abeilles.

Classe abeille parente et ses caractéristiques

```
class Abeilles {
    int x, y, radius = 4;
    Color c;
```

Position, couleur, quantité de nectar dans la source

```
public class Sources {

    Random random = new Random();
    private int x, y, i;
    private Color color;
    private int nectar;
```

classe Ruches

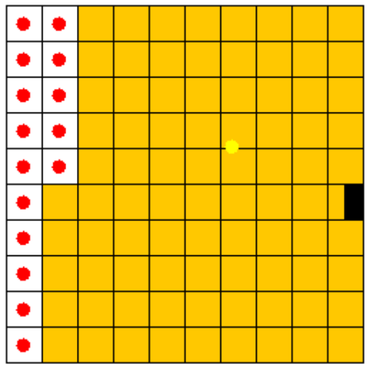
```
public class Ruches {

    private int x, y;
    private Color color;
    private int RectSize = 200;
    private int nbSquares = 10;
    private int squareSize = RectSize / nbSquares;
    private int[][] matrix;
```

```
        // Initialiser la matrice de la ruche
        matrix = new int[nbSquares][nbSquares];
        for (int i = 0; i < nbSquares; i++) {
            for (int j = 0; j < nbSquares; j++) {
                matrix[i][j] = 0;
            }
        }
    }
```

Fonctionnement du Code

- **Initialisation**
- **Déplacement des Abeilles**
- **Collecte de Nectar**
- **Affichage Graphique**

Une employée a choisi la source 2



➔ Nectar: 7ml
Source 1

➔ Nectar: 14ml
Source 2

➔ Nectar: 3ml
Source 3

Concepts de programmation utilisés

- **Modularité :** Division du code en classes distinctes pour une gestion plus claire et une maintenance aisée.
- **Abstraction :** Masquage des détails complexes pour fournir des fonctionnalités essentielles.
- **Encapsulation :** Regroupement des données et méthodes pour assurer un accès contrôlé et cohérent.

Justification des Choix de Conception

- **Modularité :** Facilite la maintenance et l'extension du code.
- **Abstraction** : Simplifie la compréhension et la gestion du code.
- **Encapsulation :** Garantit la sécurité et la cohérence des données.

Conclusion

- **Le projet de simulation d'une ruche est un exemple d'application de la programmation orientée objet.**
- **Il démontre l'efficacité des concepts de modularité, d'abstraction et d'encapsulation dans la conception de systèmes informatiques.**
- **Ce projet peut être étendu ou adapté pour des applications éducatives ou de recherche dans le domaine de l'apiculture.**