Client-serveur 2020 – chat graphique

Modifier la syntaxe d'une commande

Problème : Je veux modifier la commande *rectangle* pour tenir compte du trait.

Solution : je fais une rétro-ingénierie du code pour le modifier.

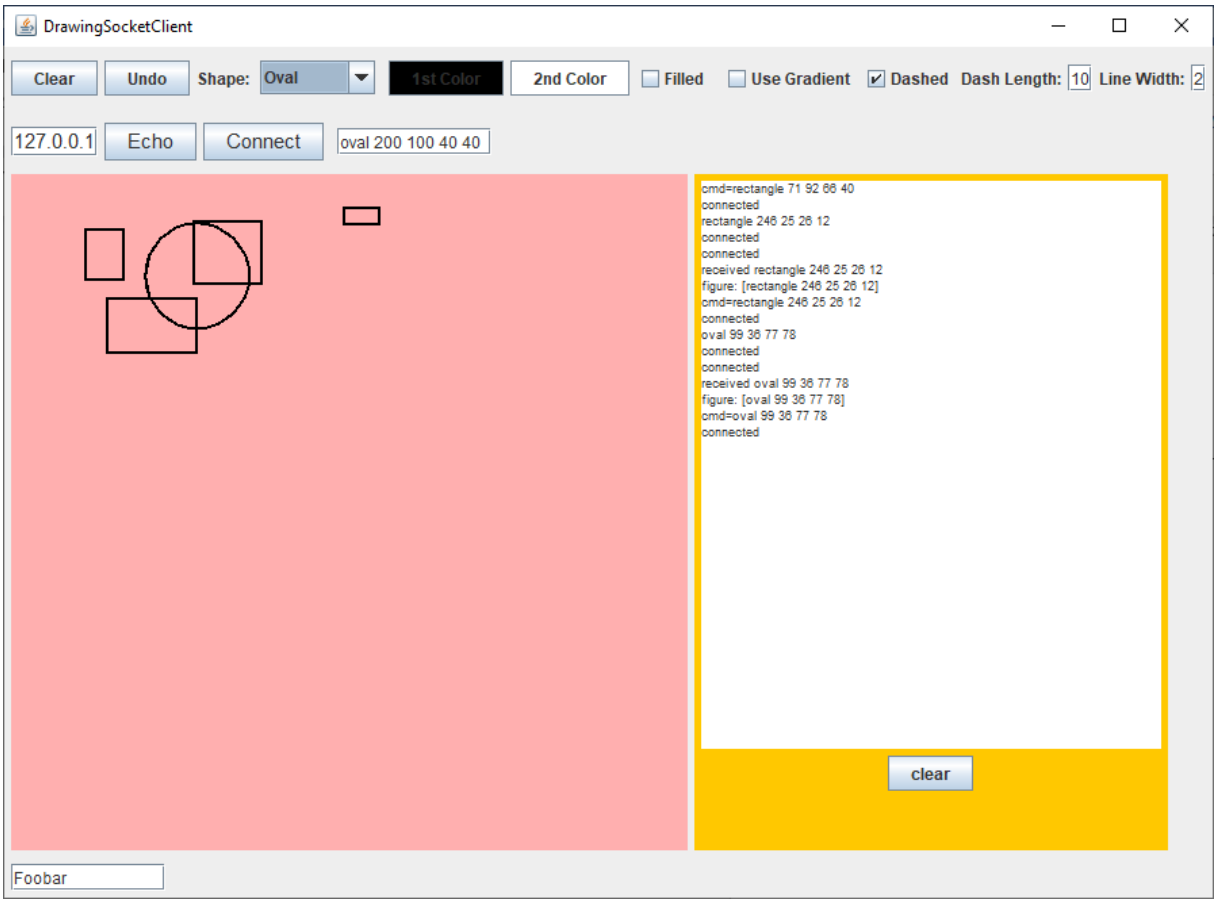

Figure 1: le client en exécution

Je recherche la commande dans la trace :

cmd=rectangle 246 25 26 12

Je recherche le code de la trace :

```
@Override
public void addCommand(String message) {
	log("cmd="+message);
	if (message.startsWith("oval")) {
		addOval(message);
	} else	if (message.startsWith("rectangle")) {
		addRectangle(message);
	}
}
```

```
294    @Override
295    public void addCommand(String message) {
296        log("cmd="+message);
297        if (message.startsWith("oval")) {
298            addOval(message);
299        } else if (message.startsWith("rectangle")) {
300            addRectangle(message);
301        }
302    }
303
```

Figure 2: Je pose un point d'arrêt en debug

```
279    private void addRectangle(String message) {
280        String[] p = message.split(" ");
281        int x = Integer.parseInt(p[1]);
282        int y = Integer.parseInt(p[2]);
283        int w = Integer.parseInt(p[3]);
284        int h = Integer.parseInt(p[4]);
285        Attributes attr = new Attributes();
286        IFigure f = new RectangleFigure(attr);
287        Dimension size = new Dimension(w,h);
288        Point location = new Point(x,y);
289        f.setSize(size);
290        f.setLocation(location);
291        addFigure(f); //oval 234 141 31 57
292    }
```

Figure 3: Je regarde le parseur pour la syntaxe, je vois qu'il y a une classe Attributes qui capture les attributs du tracé.

```
public class Attributes {

    private Color foreground, background;
    private boolean gradient, filled, dashed;
    public int lineWidth;
    public int dashLength;

    public Attributes(Color foreground, Color background, boolean gradient, boolean filled, boolean dashed,
            int lineWidth, int dashLength) {
        this.foreground = foreground;
        this.background = background;
        this.gradient = gradient;
        this.filled = filled;
        this.dashed = dashed;
        this.lineWidth = lineWidth;
        this.dashLength = dashLength;
    }

    public Attributes() {
        this.foreground = Color.black;
        this.background = Color.white;
        this.gradient = false;
        this.filled = false;
        this.dashed = false;
        this.lineWidth = 2;
        this.dashLength = 20;
    }
```

Figure 4: Il faudra ajouter une fonction au parseur pour propager les attributs dashed et dashlength

Il faudra donc parser une chaîne de caractère qui transmettra l'attribut dashed et l'attribut dashLength, et préalablement il faudra aussi générer cette chaîne à l'amission pour enrichir la syntaxe initiale de la commande « rectangle ».

La nouvelle grammaire deviendrait par exemple :

ancienne syntaxe : *rectangle 246 25 26 12*

Nouvelle syntaxe *rectangle_dashed 246 25 26 12 10*

où 10 représente dashLength

rectangle[_dashed] x y w h dash_length