ChatGPT

FalconCoreLabeling 实时更新遮罩透明度解决方案

为实现拖动透明度滑块时所有实例的遮罩透明度即时更新，并且不需要选中或悬停即可生效，需要同时修改 **LabelingWidget** 类中的透明度更新逻辑，以及 **Canvas** 绘制遮罩的逻辑。以下是具体修改方案：

修改 LabelingWidget 中透明度同步更新逻辑

在 `label_widget.py` 中修改 `LabelingWidget` 类的透明度变化处理函数，使其在滑块拖动时将新的透明度值应用到所有 LabelingWidget 实例的所有 Shape 遮罩上。具体实现为：全局更新 `Shape` 类的填充颜色透明度，并遍历所有已实例化的 LabelingWidget，将每个 Shape 对象的填充颜色 alpha 值更新为当前滑块值，并立即重绘画布。修改后的函数代码如下：

```
class LabelingWidget(LabelDialog):
    # ... 省略其他代码 ...

    @classmethod
    def _apply_fill_opacity_to_all(cls, value):
    """Apply mask opacity to every open labeling widget (update all shapes)."""
        # 更新全局 Shape 默认透明度和填充颜色的 alpha 值
        Shape.fill_opacity = value
        Shape.fill_color.setAlpha(value)
        Shape.select_fill_color.setAlpha(value)
        # 遍历所有活动的 LabelingWidget 实例，同步设置遮罩透明度
        for widget in list(cls._instances):
            # 更新配置中的透明度值
            widget._config["shape"]["fill_opacity"] = value
            # 同步更新各实例的滑块数值（避免递归信号触发）
            if widget.fill_opacity_slider.value() != value:
                widget.fill_opacity_slider.blockSignals(True)
                widget.fill_opacity_slider.setValue(value)
                widget.fill_opacity_slider.blockSignals(False)
            # 更新该实例所有 Shape 对象的填充颜色透明度
            for shape in widget.canvas.shapes:
                shape.fill_color.setAlpha(value)
                shape.select_fill_color.setAlpha(value)
            # 重绘画布，立即应用透明度更改
            widget.canvas.update()
            save_config(widget._config)

    def fill_opacity_changed(self, value):
    """滑块值改变时的回调函数，应用新的遮罩透明度。"""
        self._apply_fill_opacity_to_all(value)
```

上述修改确保了当滑块（`fill_opacity_slider`）的值变化时，会实时将新的透明度应用到所有窗口的所有Shape遮罩上，并立即触发重绘生效。

修改 Canvas 绘制逻辑以始终填充遮罩

同时，需要修改 `canvas.py` 中 Canvas 的绘制逻辑，使得无论Shape是否被选中或悬停，始终使用填充颜色绘制遮罩。找到 Canvas 类的 `paintEvent` 方法，在绘制Shape的循环中，将 `shape.fill = shape.selected or shape == self.h_hape` 修改为 **始终为 True**。修改后的关键代码片段如下（突出显示了修改部分）：

```
# 文件：canvas.py (Canvas.paintEvent 方法内部)
for shape in self.shapes:
    if (shape.selected or not self._hide_backround) and self.is_visible(shape):
        shape.fill = True  # 始终填充 Shape，不再依赖选中或悬停
        shape.paint(p)
```

通过上述改动，所有Shape将始终以填充颜色绘制（透明度由滑块值控制），不再需要鼠标悬停或选中才能显示遮罩效果。这样，拖动遮罩透明度滑块时，所有实例的遮罩透明度都会即时同步变化并在界面上实时体现。
