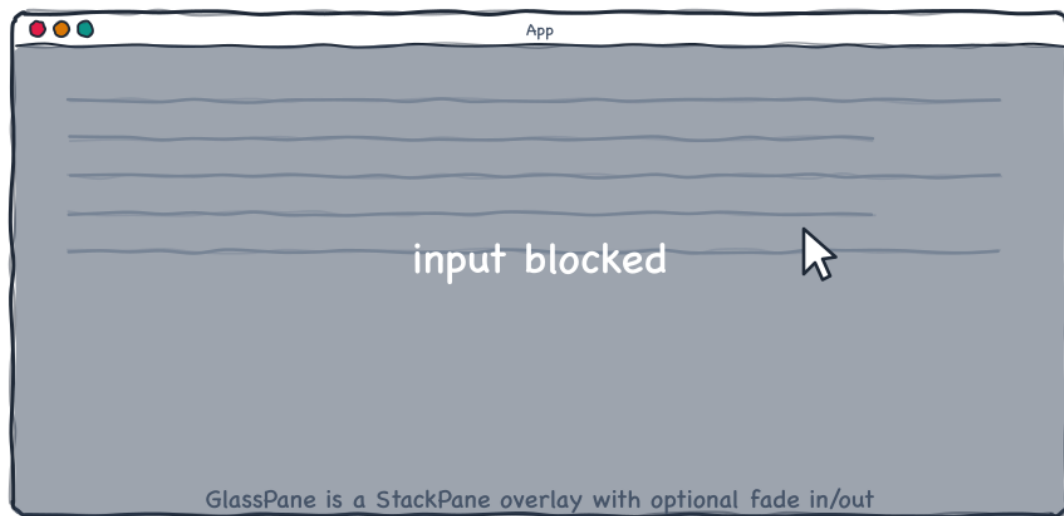


GlassPane

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



GlassPane darkens the UI and remains mouse-opaque while visible.

GlassPane is a StackPane overlay with a black CSS background, a separate blockingOpacity property and optional fade in/out animation. It is used by DrawerStackPane and can also be layered directly over application content.

Table of Contents

1. Introduction	2
2. Getting started	2
3. Anatomy	3
4. Control API	3
5. Behaviour	4
6. Layout and sizing	4
7. Styling with CSS	5
8. Interaction patterns	6
9. Recipes	7
9.1 Bind to a running property	7
9.2 Use without animation	7
9.3 Animated modal backdrop	7
10. Troubleshooting	8

1. Introduction

GlassPane extends **StackPane**. It is a lightweight overlay that blocks mouse input to content below it and optionally fades between hidden and shown states.

- Root style class is `glass-pane`.
- The constructor sets `mouseTransparent` to `false` and `visible` to `false`.
- The `hide` property controls showing and hiding.
- A separate `blockingOpacity` lets fade animation use opacity without losing the configured target opacity.

```
StackPane root = new StackPane(content);
GlassPane glass = new GlassPane();
root.getChildren().add(glass);

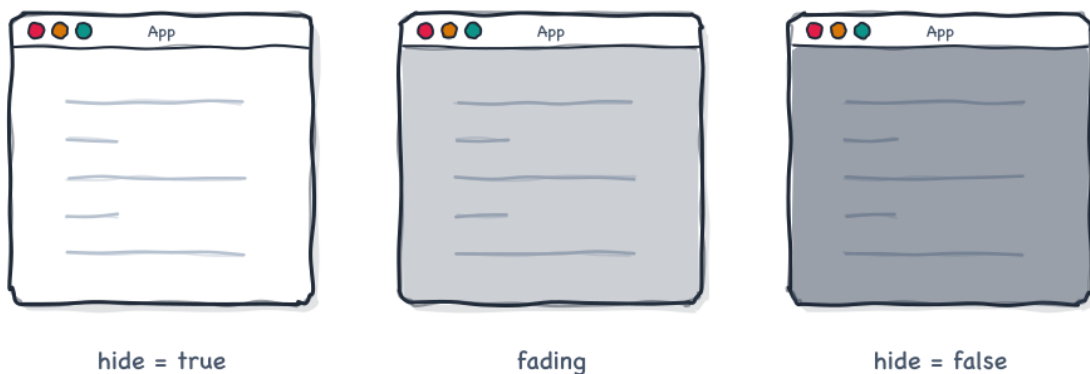
// show the overlay
glass.setHide(false);
```

2. Getting started

Add the glass pane as the last child of a **StackPane** or another layout where it covers the content. Since the CSS sets max width and height to Infinity, common panes can stretch it to fill the available area.

```
GlassPane glassPane = new GlassPane();
glassPane.setBlockingOpacity(0.45);
glassPane.setFadeInOut(true);
glassPane.setFadeInOutDuration(Duration.millis(150));

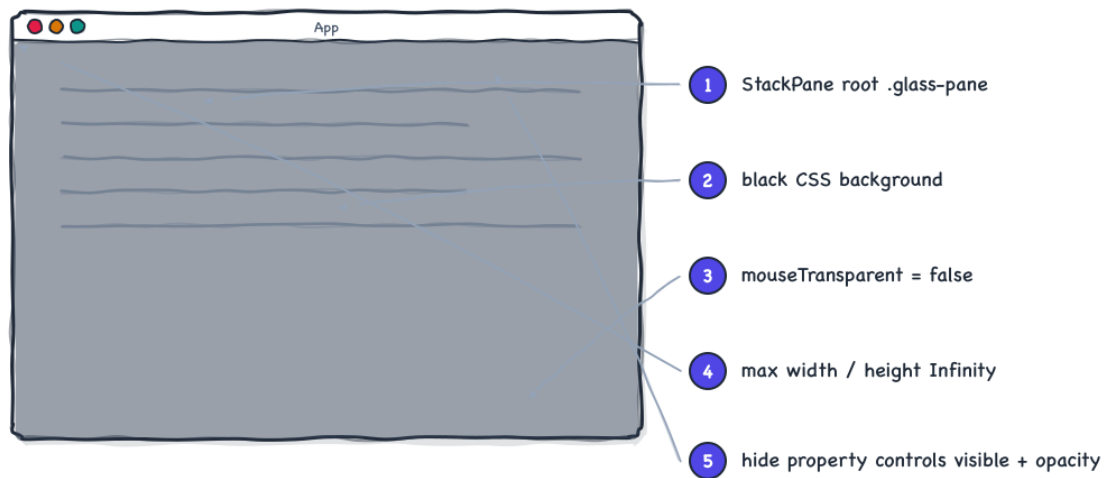
glassPane.setOnMouseClicked(evt -> glassPane.setHide(true));
```



fadeOut chooses animation or immediate visible/opacity updates

The hide property switches the pane between hidden, fading and shown.

3. Anatomy



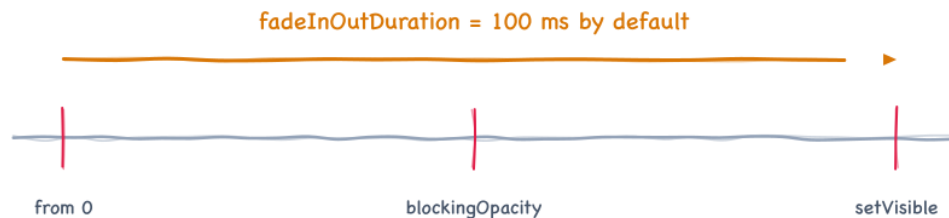
GlassPane is a StackPane plus styleable animation properties.

Part	Source	Description
Root node	StackPane	The overlay container and event blocker.
Background	glass-pane.css	Black background color.
Opacity	blockingOpacity	Target opacity while shown.
Fade transition	FadeTransition	Bound to <code>fadeInOutDuration</code> and installed on the glass pane.
Visibility	hide listener	Visible is true while shown and during fade-out.

4. Control API

Property	Type	Default	Description
<code>blockingOpacity</code>	<code>DoubleProperty</code>	0.5	Target opacity while the overlay blocks input. Values outside 0..1 are rejected by restoring the old value. Styleable.
<code>fadeOut</code>	<code>BooleanProperty</code>	false	Whether show/hide transitions animate opacity. Styleable.
<code>fadeOutDuration</code>	<code>ObjectProperty<Duration></code>	100 ms	Duration of the fade transition. Styleable.
<code>hide</code>	<code>BooleanProperty</code>	true	True means hidden; false means shown. This property is not styleable.

5. Behaviour



Fade animation uses blockingOpacity as its target value.

When hide changes	fadeOut false	fadeOut true
false	Opacity becomes blockingOpacity and visible becomes true.	Visible becomes true; transition runs from 0 to blockingOpacity.
true	Opacity becomes 0 and visible becomes false.	Transition runs from blockingOpacity to 0; visible becomes false when finished.

If a fade transition is already running when hide changes again, the current transition is stopped before the new one starts.

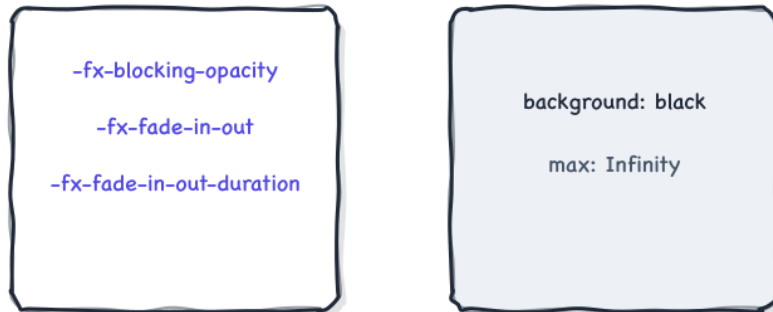
6. Layout and sizing

GlassPane itself does not override layout. It relies on StackPane sizing and its user agent CSS:

```
.glass-pane {
  -fx-max-width: Infinity;
  -fx-max-height: Infinity;
  -fx-background-color: black;
}
```

Tip. Place the glass pane after the content in the parent children list so it is painted and hit-tested above the content.

7. Styling with CSS



CSS sets the color; properties set the opacity and animation

CSS controls the fill color; styleable properties control opacity and timing.

CSS property	Type	Default
<code>-fx-blocking-opacity</code>	number	0.5
<code>-fx-fade-in-out</code>	boolean	false
<code>-fx-fade-in-out-duration</code>	Duration	100ms

```
.glass-pane {  
    -fx-background-color: #111827;  
    -fx-blocking-opacity: 0.62;  
    -fx-fade-in-out: true;  
    -fx-fade-in-out-duration: 180ms;  
}
```

8. Interaction patterns

The class only supplies the overlay mechanics. Applications decide what a click means. DrawerStackPane uses its glass pane to auto-hide the drawer when the user clicks outside it.

```
glassPane.addEventHandler(MouseEvent.MOUSE_CLICKED, evt -> {  
    if (evt.getButton() == MouseButton.PRIMARY) {  
        glassPane.setHide(true);  
    }  
});
```

- 1 Create the glass pane once.
- 2 Keep it as the top child of the layered parent.
- 3 Set hide to false before starting work that must block input.
- 4 Set hide to true after the blocking operation or dialog completes.

9. Recipes

9.1 Bind to a running property

```
glassPane.hideProperty().bind(service.runningProperty().not());
```

9.2 Use without animation

```
glassPane.setFadeInOut(false);  
glassPane.setHide(false);
```

9.3 Animated modal backdrop

```
glassPane.setFadeInOut(true);  
glassPane.setBlockingOpacity(0.35);  
glassPane.setFadeInOutDuration(Duration.millis(120));
```


10. Troubleshooting

- Remember that `hide=false` means the pane is shown.
- Setting `blockingOpacity` outside 0..1 is ignored by reverting to the previous value.
- The class does not set an `AccessibleRole` or accessible text in the source.
- If clicks reach controls below, verify the glass pane is the top child and `mouseTransparent` is still false.