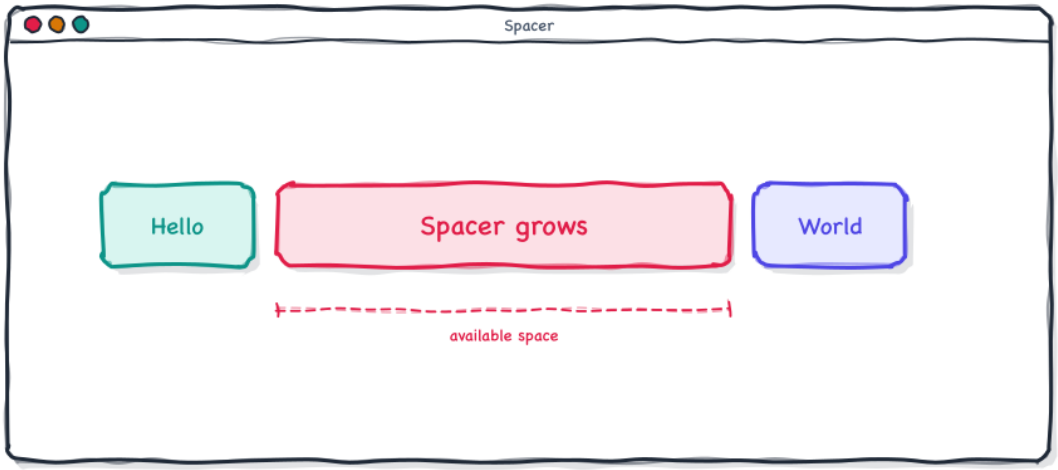
Spacer

Developer Manual

GemsFX • com.dlsc.gemsfx • JavaFX Custom Controls



A generated cartoon overview of Spacer.

Spacer is a small Region subclass that grows in HBox or VBox and can collapse itself by binding visible and managed to a styleable active property.

Table of Contents

1. Introduction

Spacer is a tiny `Region` that sets HBox and VBox grow priority to ALWAYS. Its styleable active property controls whether it is visible and managed.

1.1 Key features

- Style class `spacer`.
- active defaults to true.
- visible is bound to active.
- managed is bound to visible.
- HBox and VBox grow priorities are ALWAYS.

1.2 Maven dependency

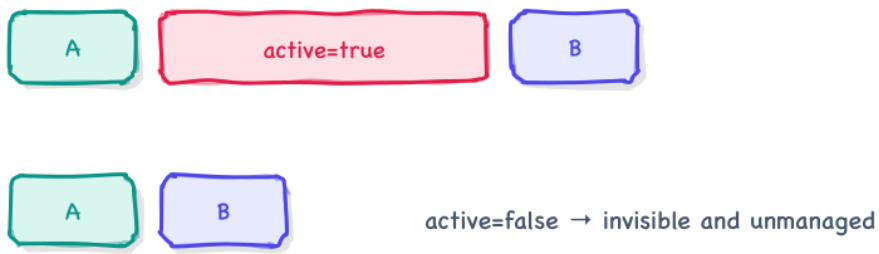
```
<dependency>
    <groupId>com.dlsc.gemsfx</groupId>
    <artifactId>gemsfx</artifactId>
    <version>4.4.1</version>
</dependency>
```

Maven coordinates for the GemsFX control library.

2. Getting started

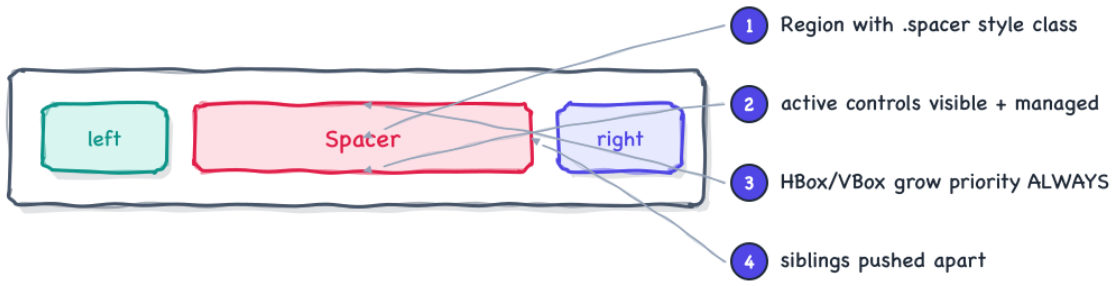
```
HBox row = new HBox(8, new Label("Hello"), new Spacer(), new Label("World"));
VBox column = new VBox(new Label("Top"), new Spacer(), new Label("Bottom"));
```

Spacer in horizontal and vertical boxes.



Active and inactive spacer states.

3. Anatomy

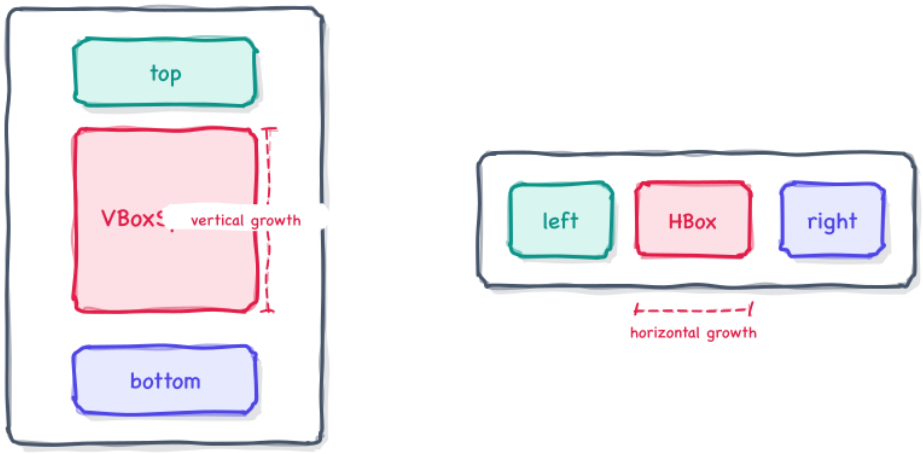


A Spacer between two sibling nodes.

Property	Type	Default	Description
active	BooleanProperty	true	Controls visible and managed state. Styleable with -fx-active.

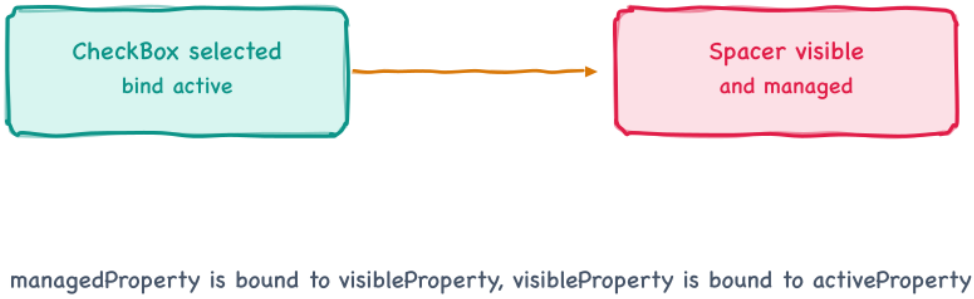
Constructor action	Effect
style class spacer	Application selector.
managed.bind(visible)	Collapsed spacer reserves no space.
visible.bind(active)	active drives visibility.
HBox/VBox grow ALWAYS	Consumes extra space in common box containers.

4. Layout behaviour



The same class grows horizontally in HBox and vertically in VBox.

Spacer does not inspect its parent. It simply sets both grow constraints; the parent layout that understands one of them uses it.



Binding active controls visibility and layout management.

5. Styling

There is no user agent stylesheet. The CSS metadata exposes `-fx-active`.

CSS property	Type	Default in metadata / property
`-fx-active`	boolean	metadata false, property true

```
.debug .spacer {
    -fx-background-color: rgba(255, 0, 128, .15);
}
.spacer {
    -fx-active: true;
}
```

Note. The CSS metadata default is false, while the property instance starts true. A stylesheet value wins when CSS is applied.

6. Recipes

6.1 Toggle a group of spacers

```
CheckBox active = new CheckBox("Spacer active");
spacer.activeProperty().bind(active.selectedProperty());
```

6.2 Toolbar right alignment

```
toolbar.getChildren().addAll(leftButtons, new Spacer(), rightButtons);
```

6.3 Checklist

1. Use Spacer primarily in HBox and VBox.
2. Bind active rather than manually changing managed and visible.
3. Use temporary backgrounds for layout debugging.
4. Do not use Spacer when fixed padding is the real requirement.

7. See also

- Demo application: `com.dlsc.gemsfx.demo.SpacerApp` (run with `mvn javafx:run -f gemsfx-demo/pom.xml -Dmain.class=com.dlsc.gemsfx.demo.SpacerApp`)
- `ThreeItemsPane` - semantic three-slot alternative.
- JavaFX `Region` - Spacer superclass.
- API documentation: https://dlsc-software-consulting-gmbh.github.io/GemsFX/api/