Menu User Experience Documentation

Authors: Jan Rojcek, Chip Alexander

1. Introduction

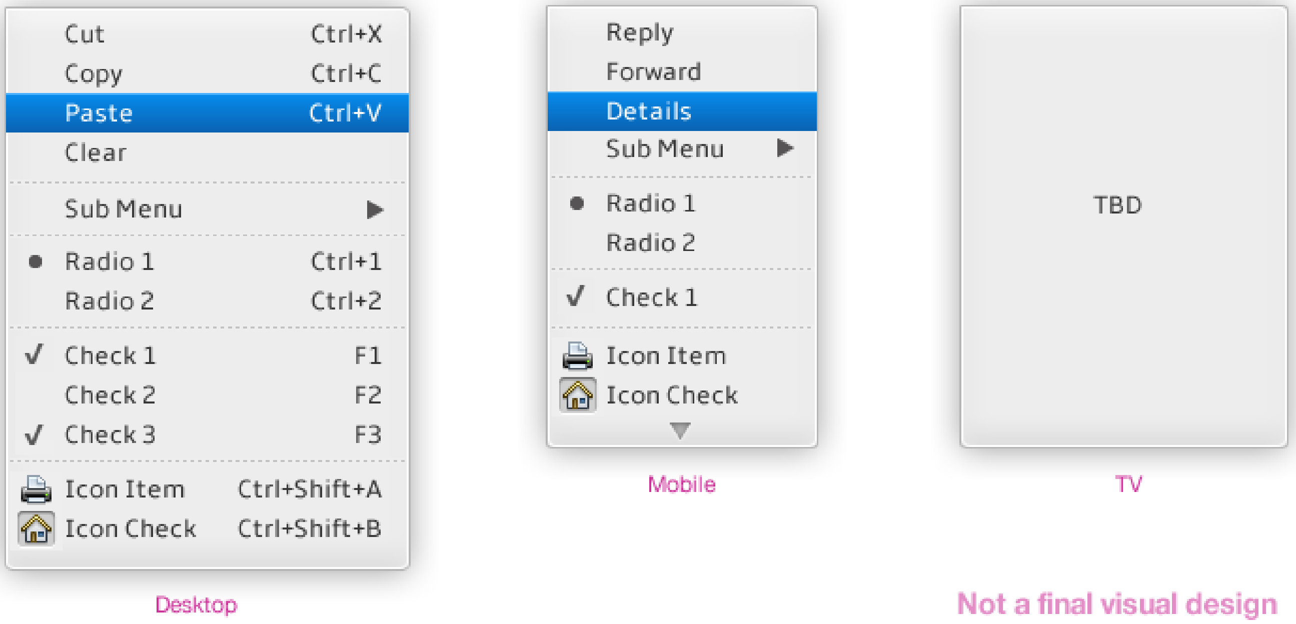
This document is the user experience specification for the JavaFX Menu control.

2. Menu Overview

The Menu is a control used for choosing one item from a list of predefined menu items. Menu items are typically actions (*Open File...*), options (*Full Screen* - a check box or radio menu item) or objects (*www.java.com* - a bookmark). A Menu normal state is closed. To use a Menu, the user needs to open it and select an item which automatically closes the menu. Menus can be open from various places: Menu Bar, Menu Button, Split Menu Button, various custom controls and as a contextual menu using a secondary pointer button on Desktop platform.

Menu shares some common characteristics and behaviors with Choice Button's Popup List (appearance, scrolling, hover behavior, selection behavior, etc.). Menu does not have a concept of current value which is present in Choice buttons. On the other hand, Menus can have shortcuts and submenus which are not available in Choice Buttons.

Figure 1 Menu on Desktop, Mobile and TV platform



Desktop

Mobile

TV

Not a final visual design

Menu items

Menu can contain the following menu items:

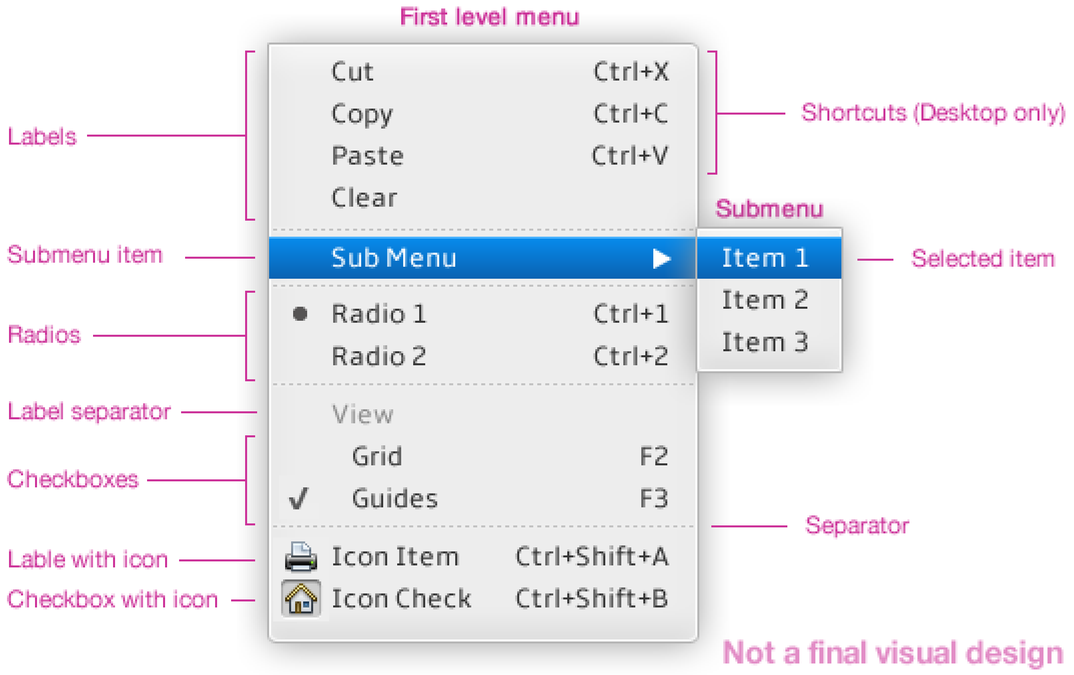
- Label
- Label with icon
- Icon only
- Check box
- Check box with icon
- Radio
- Submenu
- Separator
- Labeled separator
- *Custom menu item*

Separator and labeled separator are non-selectable.

All items except separators and submenu items can be triggered - the associated action gets invoked and menu closes.

On Desktop platform, all items except Separators and Submenus can have shortcuts. *Note: We may consider permitting shortcuts also for Submenus which is sometimes used in complex applications with strong focus on enduser productivity with keyboard (typically IDEs - integrated development environments).*

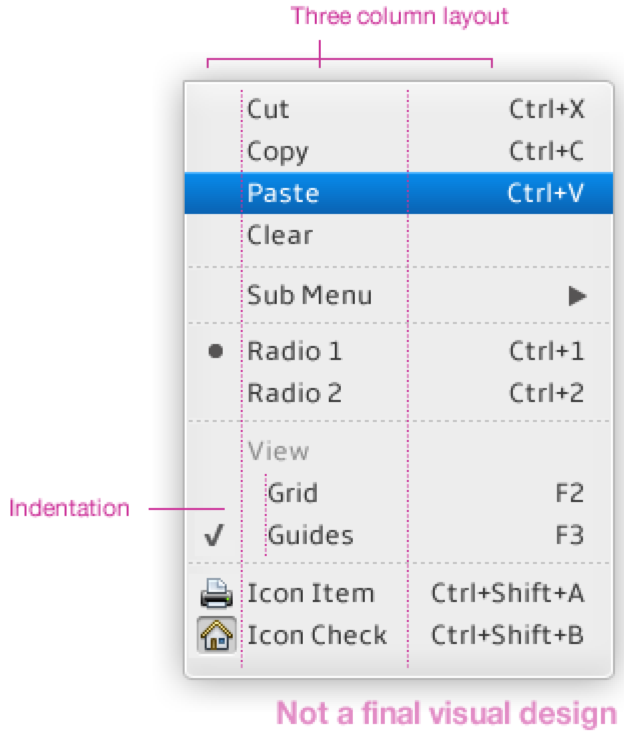
Figure 2 Menu items



Layout

Menu items are typically laid in a three column layout with possible indentation of items. As developers can add custom menu items, the layout needs to be as flexible as possible to permit adding vertical visual separators, search boxes, progress bars, icon items in a grid, icons on the right hand side, etc.

Figure 3 Typical menu layout



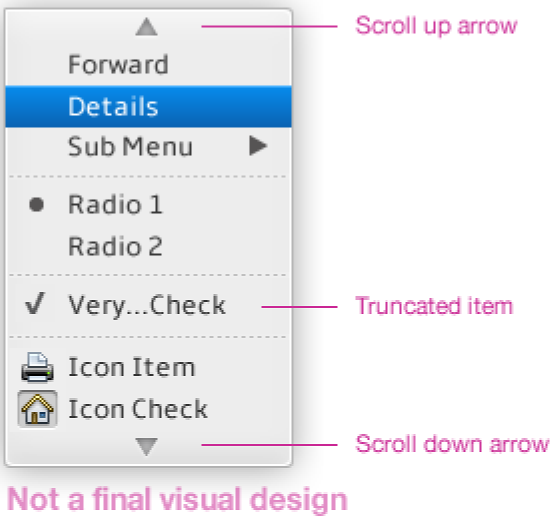
Sizing and Scrolling

The Menu width is set by the size of the largest item in the list. Its height is set to make all items visible without the need to scroll menu items on Desktop platform. On Mobile and TV platform where screen space is limited, developers can decrease the default menu size so that it does not overlap all controls on the screen. In such case and in case when the menu size is automatically limited to fit the available screen size, the menu items scroll vertically and they are truncated horizontally.

When scrolling is needed to reveal all menu items, Scroll Arrow Buttons are added to the top and bottom of the menu. The arrow indicator graphics inside each of top and bottom buttons are displayed only when hidden list items are present in the direction of the arrow.

On the desktop and touch platforms, the scrolling arrows can be used for scrolling the menu items. On the 2D traversal platforms, they serve only as indicators that other items exist which are not shown. On the desktop platform, the Mouse Wheel works for scrolling items as well.

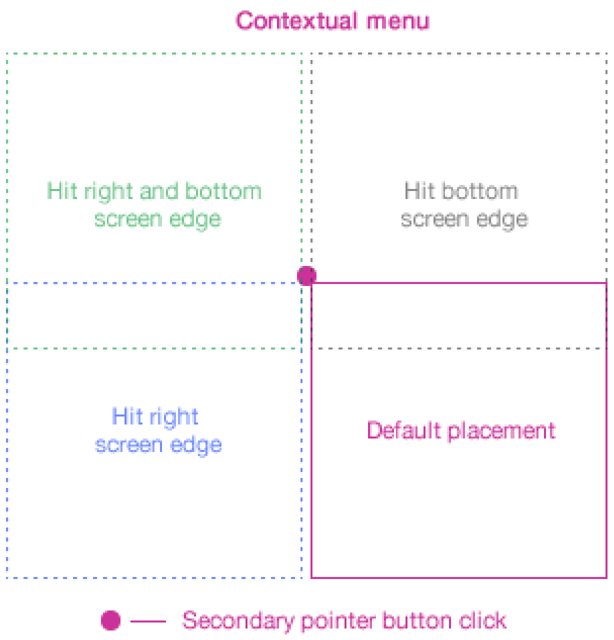
Figure 4 Menu with smaller than needed size

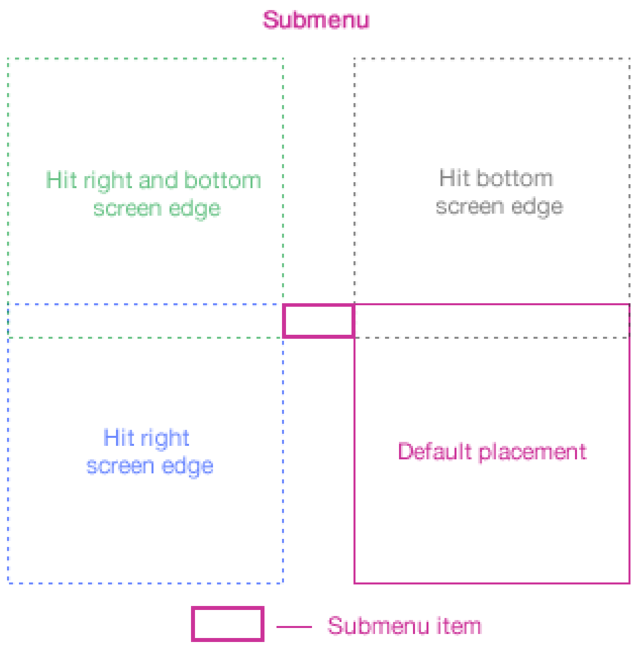


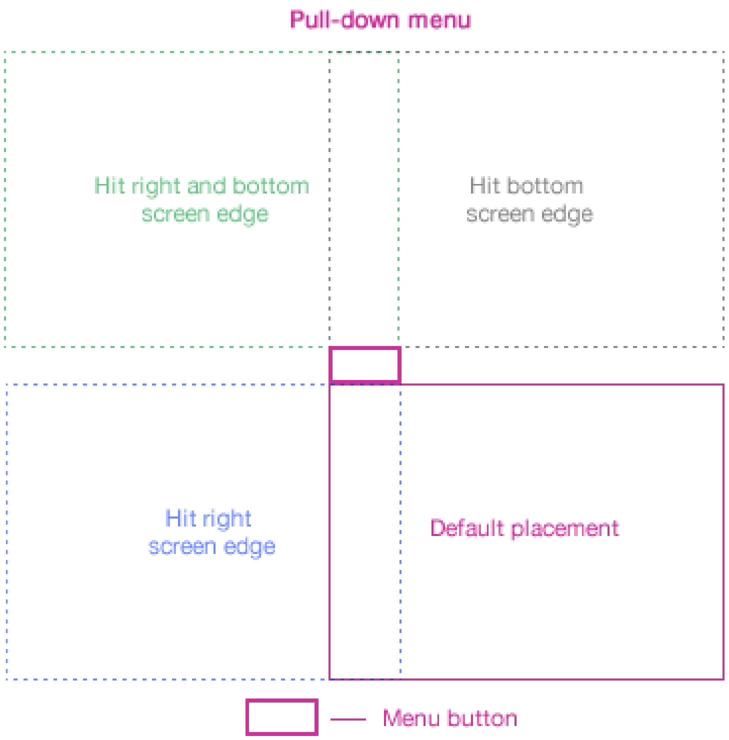
Placement

Menu opens on a position relative to the control or place which initiated menu opening. There are multiple possible positions depending on whether there is enough space to open the menu. See the following figure. If the menu is too big to fit on the screen in either position, it adjust its position within the corresponding axis. If it still does not fit in, it opens with scrolling arrows.

Figure 5 Possible menu opening positions







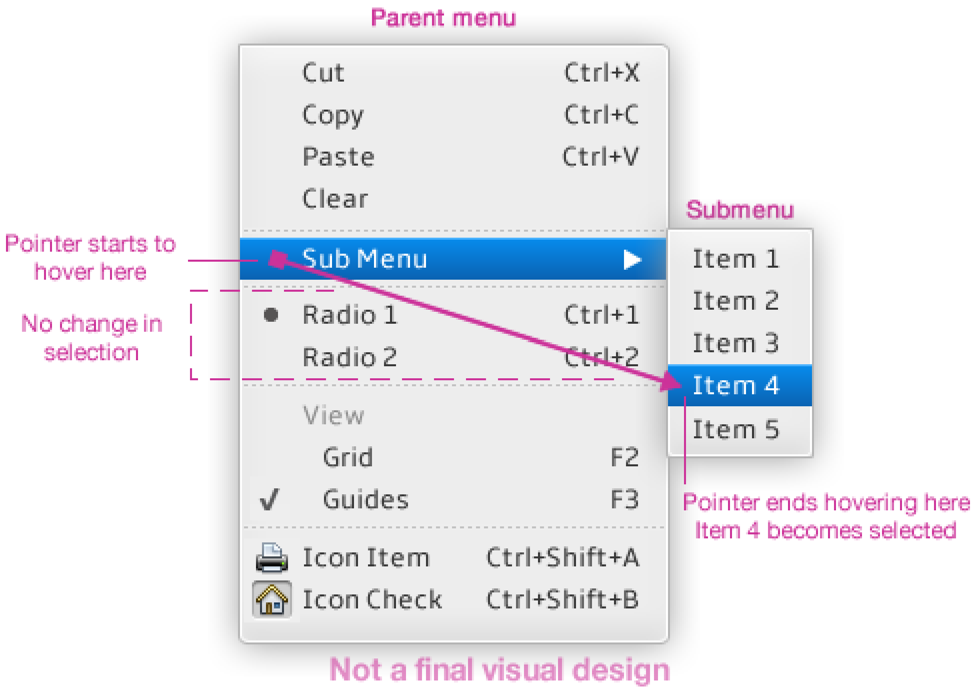
3. Detailed Behaviors

Menus are typically controlled with a pointing device. Hovering over menu selects the menu item underneath the pointer. Pressing the pointing device button above the menu area performs no user visible action. Releasing the button triggers the menu item underneath the pointer if it can be triggered (not separator or disabled item). If an item gets triggered, it closes the menu.

Menu can be used in a so called One Step Selection approach described in detail in Choice Button UE spec. The user presses a pointing device button on a Menu button, which opens the Menu, and while holding the pointing device button, the user selects a desired menu item and releases the pointing device button. This way the user can perform a menu item selection with one "long" click.

Menu also employs a technique for easier use of submenus. If a submenu is open and the user moves pointer fast enough from the parent menu item to the submenu while crossing other menu items, it does not select the hovered-over parent items. On **Figure 6**, items "Radio 1" and "Radio 2" do not get selected even if the pointer hovers over on the way from the parent item "Sub Menu" to Submenu's "Item 4". This behavior makes it easier for users to move pointer to submenu without closing it accidentally. It works only if the pointer moves fast enough and under right angle. If the pointer moves slowly or it does not aim towards submenu items, it select a parent menu item and closes the submenu.

Figure 6 Special-case hover behavior



Pointer starts to hover here

No change in selection

Pointer ends hovering here Item 4 becomes selected

Not a final visual design

Table 1 Pointer gestures

Pointer Gesture	Target	Behavior
Hover Pointer	Menu item	Selects the menu item if it can be selected. Selected submenu item opens its menu (after a short delay). Separator does nothing.
	Scroll arrow button	Scrolls the menu items in the corresponding direction.
	Outside of Menu	Clears selection if selected item was not a submenu item. Submenu item stays selected and the submenu stays open.
Primary Button Press	Menu item	Nothing.
	Scroll arrow button	Scrolls the menu items in the corresponding direction.
	Outside of Menu	Closes the menu.
Primary Button Release	Menu item	Triggers the menu item if it can be triggered. If it is a submenu item, it opens the submenu without a delay. Does nothing on separators and disabled menu items.
	Outside of Menu	Nothing.
Mouse wheel scroll	Menu area	Scrolls the menu if items are hidden in the corresponding direction.

Table 2 Menu states

State	Description
Menu	

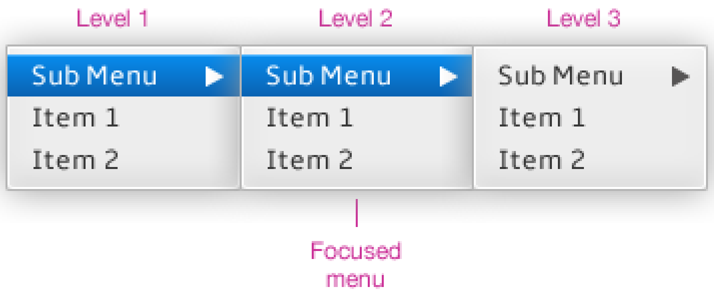
Open	Menu is visible on the screen.
Closed	Default state. Menu is not visible on the screen.
Menu item	
Normal /Enabled	Default state.
Selected/Hover	If it is a regular menu item then the selected state indicates it can be triggered. If it is a submenu item, it indicates the submenu is open (the submenu opens after a short delay after the submenu item becomes selected). Separators cannot be selected.
Disabled	Menu item can be selected but cannot be triggered.
Scroll arrow button	
Normal /Enabled	It indicates there are hidden items in the corresponding direction.
Disabled	It indicates there are no hidden items in the corresponding direction. Clicking it does nothing.
Hover	If it is enabled, it scrolls menu items in the corresponding direction.
Pressed	If it is enabled, it scrolls menu items in the corresponding direction.

4. Keyboard Support

Keyboard support provides 2D focus traversal, mnemonic and special typing access to menu items. Menu items can also be triggered with keyboard shortcuts (e.g. Cltr+C), but as it does not result in any user visible change in Menus, it is not covered in this spec.

As multiple menu items can be selected in the hierarchy of open submenus, the lower-level submenu containing selection is the one holding focus and as such reacting to key events.

Figure 7 Focus in menu hierarchy



2D focus traversal

If a first level menu is open with no item selected yet and the user presses the down (up) key, it selects the first (last) menu item. Pressing up/down keys changes selection. It cycles around, meaning that if the first (last) item is selected and the user presses the up (down) key, the last (first) item becomes selected.

If a submenu item is selected, it opens the submenu after a short delay. Or if the user presses the right key, it opens the submenu immediately (if not already open) and selects its first item. This menu becomes focused. Pressing left key on any item in a submenu clears the selection from the submenu and moves focus to the parent menu.

Mnemonic access

TBD - Windows-style Alt+Character access to menu items. Better for productivity.

Typing access

TBD - Mac-style access to menu items by typing a first few characters of a menu item. No need to care about it by developers.

Table 3 Keyboard support

Key	Behavior
Up	In a first level menu with no selection yet, it selects the last menu item. Otherwise moves selection one item up (and scrolls one item if needed). If the first item is selected, it selects the last item.

Down	In a first level menu with no selection yet, it selects the first menu item. Otherwise moves selection one item down (and scrolls one item if needed). If the last item is selected, it selects the first item.
Right	If a submenu item is selected, it selects the first item in the submenu and gives it keyboard focus. If a regular item is selected, it does nothing (except if the menu is part of a menu bar - see the Menu Bar UE Spec for details).
Left	If focus is in a submenu, it clears selection in the submenu and moves focus to the parent menu. If a regular item is selected, it does nothing (except if the menu is part of a menu bar - see the Menu Bar UE Spec for details).
Enter, Space	Triggers the selected item if it can be triggered (needs to be enabled). If the selected item is a submenu item, it selects the first item in the submenu and gives it keyboard focus.
Esc	Closes the menu. If it is a submenu, it moves focus to parent menu.