

UniPie

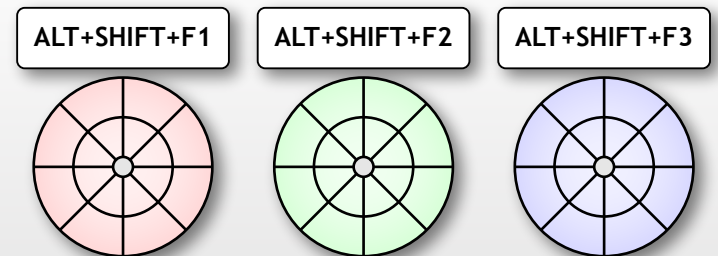
A universal pie menu system

UniPie is a Windows 7 application running in the background until invoked by a hotkey (or mouse button). When that happens a pie menu shows up where the user can pick one of the slices.

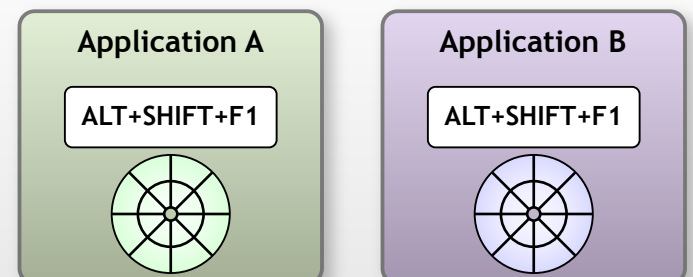
Each slice represents an action: sending one or more virtual button presses or starting an application.



Any number of pie menus can be defined, each with its own hotkey.

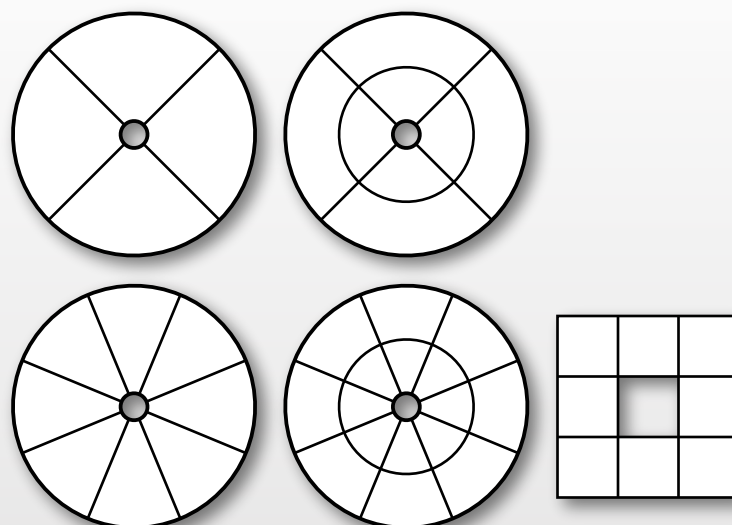


Pie menus can be system wide or application specific. In the latter case the same hotkey can invoke different pie menus depending on the application currently in focus.



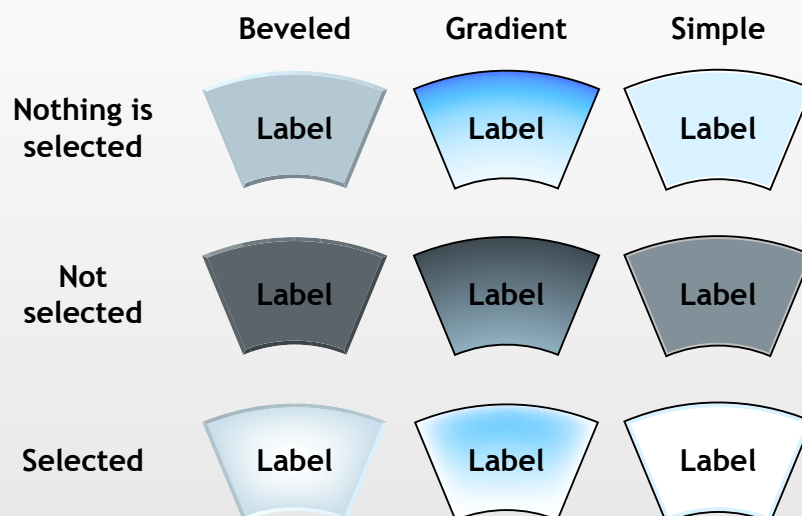
There are several menu layouts to choose from:

- Radial 4
- Radial 4+4
- Radial 8
- Radial 8+8
- Rectangular 3x3

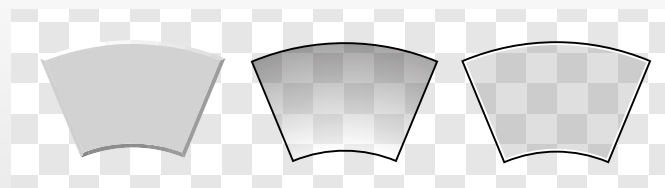


The following visual styles are available: *Beveled*, *Gradient* and *Simple*.

If nothing is selected (the mouse is not over the pie menu) then the slices are neither dimmed nor highlighted. If a slice is selected, the others are dimmed.



Blank slices show up differently with different visual styles.



This is a special menu, the Numeric Input Pie (NIP), which can be called from inside numeric input fields.

Although it was directly “inspired” by annoyances in Adobe Illustrator, the problems it solves are present in many other applications as well.

NIP provides a way to comfortably and precisely adjust a numeric value, unlike the opacity slider in Illustrator which is so tiny that it’s a pain to grab and precisely adjust.

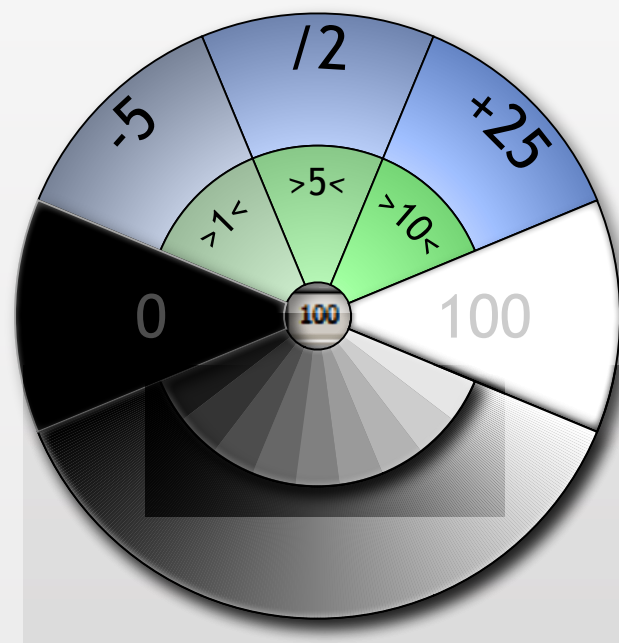
To fix that, NIP has gradients which represent values between the two extremes (0 and 100 in this example, but it can be customized per input field). You could think of it as a color picker for numbers.

The longer, smooth gradient allows fine adjustment while the shorter one has distinct steps for quantized values. (The number of steps can be customized.) The black and white slices show the low and high values and make easy to zero/max out the numeric field.

The other problem I often encounter is the failure of grid snapping when I end up with coordinates slightly off grid (198.74 instead of 200.0 and so on).

To help with that, NIP can round the existing value using configurable rounding targets (green slices).

Finally basic math operations can be performed on the number present in the input field.



For each slice the user can define the following properties:

- Label or icon.
- Hue and saturation.
- Keyboard shortcut(s) to produce or application to start.

The labels are automatically aligned and rotated according to the slice they are in.

When multiple shortcuts (a macro) are defined, a delay can be set before each key.

When an application is to be started, the tasklist is checked: if an instance of that application is already running then it is brought to focus instead of starting it again.

The pie menu always pops up under the cursor.
(Multiple monitors are supported.)

By default, a pie can be cancelled in two ways:

- The user releases the pie caller hotkey while the cursor is outside the pie. The pie goes away, the hotkey never arrives to the focused application.
- The user releases the pie caller hotkey while the cursor is inside the center hole. The menu disappears but the hotkey is passed on to the application in focus.

A pie menu has several properties for fine tuning its behavior:

- *Infinite outer slices:*

If this flag is true then outer slices extend to infinity (logically, not visually) so one of them will be selected even if the cursor is off the pie menu. (The center hole of the menu still deselects everything.)

- *Release hotkey to show menu:*

When this flag is on then releasing the menu hotkey will display the pie, pushing and releasing again will hide it. If the flag is turned off then holding down the hot key keeps the menu shown.

- *Click selects slice:*

If this flag is set then a slice will be executed only when clicking on it with the left mouse button. Otherwise hiding the menu with a slice highlighted will run it.

UniPie also features an input remapper: certain keystrokes or mouse buttons can be translated into others. For example the middle mouse button could act as the space bar in Photoshop for easier panning.

UniPie is developed in C#, uses WPF controls for configuration UI and vector graphics for drawing the pie menus.

It can monitor key strokes and mouse events and catch them if necessary before they reach the application currently in focus.

Key strokes and mouse events can be sent by UniPie to the target application.

UniPie is minimized to the systray, its configuration UI can be opened by clicking on the icon there.

Configuration is stored in a text file in JSON format.

Windows 7 is the target platform, earlier windows versions are not supported.