

The Wayback Machine - <https://web.archive.org/web/20110721001737/http://jonoscript.wordpress.com:80/20...>

- [Home](#)
- [These things I believe.](#)

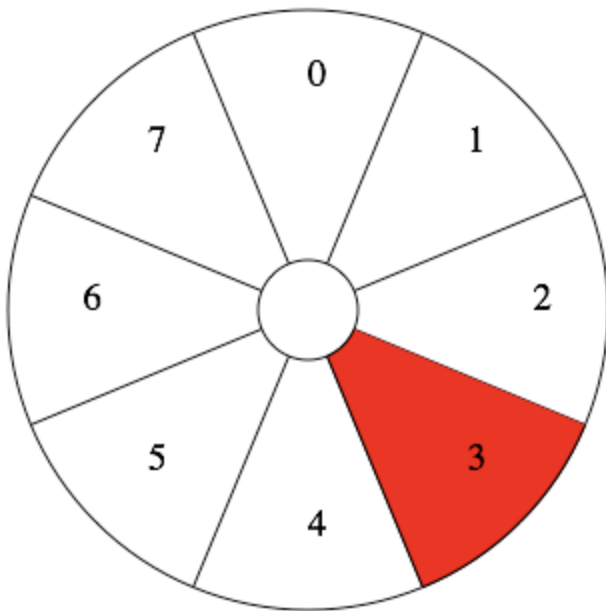
Not The User's Fault

Jono at Mozilla Labs

October 28, 2008

Pie In The Sky

Posted by jonoscript under [Uncategorized](#) | Tags: [Fitts' Law](#), [interactive demos](#), [pie menus](#), [ubiquity](#), [UI math](#) | [\[143\] Comments](#)



This blog post is a hands-on experiment. If you have Firefox 3, before you read any further, please go to [this page and try out the demo](#).

(Unfortunately, the demo will only work in Firefox 3.0 and later, because it uses some cutting-edge features of the Canvas object that aren't supported in Internet Explorer, Safari, or older versions of Firefox. I believe strongly in presenting web content in a cross-browser compatible way, but sometimes for feature demos, requiring Firefox is unavoidable. This is *Mozilla* labs, after all.)

Anyway, the demo presented two different styles of pop-up menus and asked you to repeatedly select the same menu item.

Which menu style made the task easier? Why do you think that was?

A Brief History of Pie Menus

Apple Inc. v. Smith Interface Techs., LLC
IPR2024-01083 | Smith EX2009

Pie menus are nothing new. [Don Hopkins](#) has been evangelizing them for decades. Mozilla has [experimented with pie menus before](#), although (as far as I know) the experimentation hasn't led to any core browser features.

Also, here's [Radial Context](#), an extension that replaces the Firefox context menu with a pie menu; unfortunately, it's no longer actively maintained and is only compatible with Firefox versions 2.0 and older.

[UPDATE: [Here is an updated version of the Radial Context extension](#) that works with Firefox 3.]

Do you know of any good pie menu implementations for leading browsers? If so, please leave a link in the comments. For example, I know that Opera has mouse gestures, but does it have pie menus?

Fitts' Law

The reason why pie menus are good has to do with [Fitts' Law](#), one of the most reliable theories in all of HCI, which has been empirically verified time and time again. Sadly, a lot of people who do UI design, even professionals, don't know about this simple and useful formula:

$$T = a + b \log_2 \left(\frac{D}{W} + 1 \right)$$

where:

- **T** is the time taken to move a pointing device to a target. It's a function of:
- **a** and **b** are constants, which depend on the hardware and the individual person using it, so they're generally outside the control of UI designers.
- **D** is the distance from the starting point to the center of the target.
- **W** is the width of the target, in the direction of motion.

In other words: bigger things are easier to hit than smaller things, and closer things easier to hit than those farther away. (No duh, Sherlock!)



In a pie menu, **D** is very low, because the pie slices are all adjacent to the starting point. And **W** (the width of the target) is correspondingly large, since each pie slice occupies the largest possible area. (In some implementations, the pie slice area effectively extends all the way to the edge of the screen, making **W** very large indeed.) Therefore Fitts' Law predicts that the time to acquire the target in a pie menu should be minimal.

Habituable Mouse Gestures

But that's not all. Besides Fitts' Law, pie menus are good because of the habits they teach.

What are you doing with your hands when you select from a menu? You're making a *mouse gesture* while holding down the mouse button.

We usually think of the menu that you see on the screen as a UI element, but here's a different way to think about it: *the mouse gesture IS the command*. The menu on the screen is just visual feedback to help you find the right mouse gesture.

With a traditional list-style menu, the mouse gesture consists of moving downward by a certain distance. Choosing a different item means moving a different distance downward.

With a pie menu, the gesture consists of moving a certain direction. Choosing a different item means moving a different direction.

Gestures that vary by *direction of movement* are easier to distinguish, kinesthetically, than gestures that vary only by *distance of movement*. That means that the gestures involved in selecting from a pie menu are easier to perform without looking. And it's always easier to form a habit around a UI interaction when it can be performed without looking.

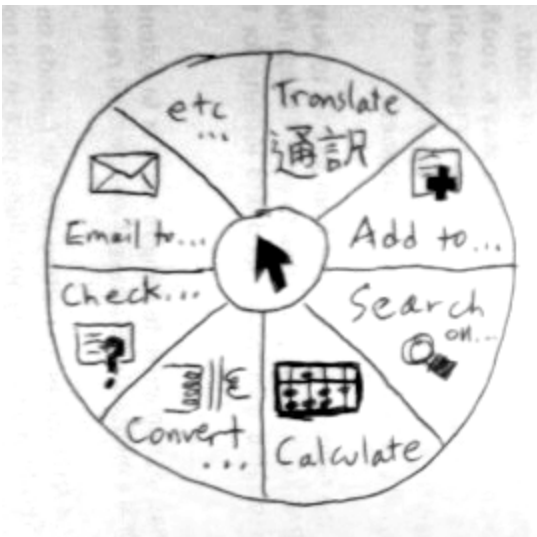
A good user-interface is one that naturally teaches the user effective habits. And pie menus are an effortless way of learning efficient mouse gestures.

Pie Menus in Ubiquity

For a UI widget with so many advantages, pie menus are sorely underused. They're used in computer games ([The Sims](#) being the best-selling current example), and they're used in (shameless plug) [Songza](#) — click on the title of a song to see a pie menu.

But pie menus are conspicuously absent from most popular UI libraries and toolkits. They're not in the Human Interface Guidelines of Mac, Windows, Gnome and KDE. Many UI designers don't even know they exist. Sure, round things are a little harder to program than square things, since they tend to involve polar coordinate math, but it's not *that* much extra effort.

I'm particularly interested in the possibility of using pie menus to replace the Ubiquity context menu. (Wait, Ubiquity has a context menu?) Yes, it does, but it's not well-documented or widely-used. It's also not very useful, since it's currently a big, unwieldy, unsorted, list of random commands which may or may not be applicable. It would be cool if right-clicking on some selected text — or an image or link, for that matter — resulted in a menu like this:



Not a Panacea

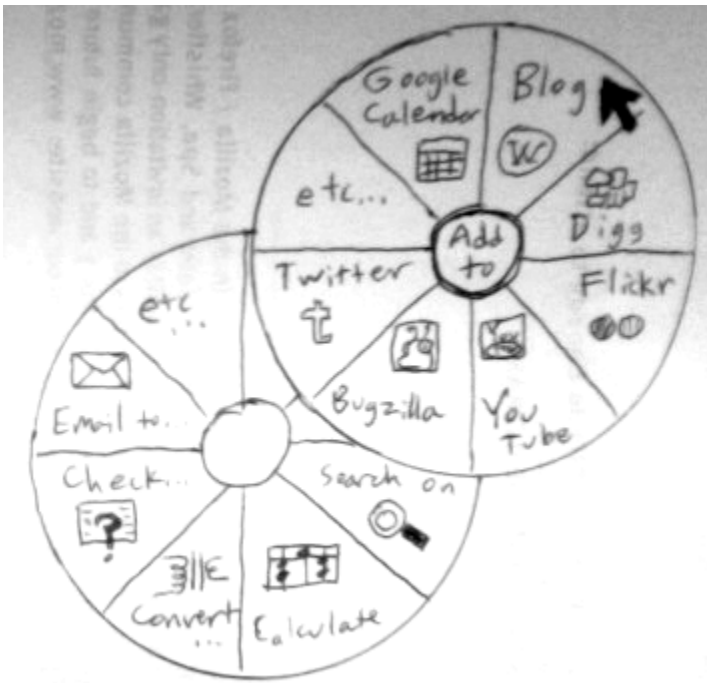
Just replacing a rectangular context menu with a pie menu doesn't magically make everything better, though. In fact, it introduces a new set of problems that need to be solved.

One of the problems that we'd immediately face is the large number of Ubiquity commands available. At my count, there are no fewer than **twenty** built-in Ubiquity commands which can take arbitrary text for one or more of their arguments. That means that when we bring up a Ubiquity context menu on a text selection, we start at a *baseline* of twenty commands, *before* adding in third-party command subscriptions or specialized commands that can act on the particular data-type of the selection.

Even if we increase the specificity of commands and aggressively filter non-applicable commands out of the context menu, we're still looking at a large and ever-growing set of commands.

Time for another demo. Try out [the pie menu on this page](#) in order to understand the usability problems that arise when there are too many items in a pie menu. (Basically, when the wedges get too small, you lose the benefit of being able to select one without looking at it.)

One approach to this problem is to allow a hierarchy of pie menus. For example, if we took the previous illustration as a starting point, then if the user mouses over the "Add to..." pie wedge, they might see a new subsidiary pie menu pop up with the list of "Add to..." subcommands. Like this:



(I don't have an interactive demo for this yet, unfortunately. It involves a lot more programming than I wanted to do for this blog post.)

I think hierarchical pie menus will be a big step forward. But another problem still remains. To understand what it is, [here is the third and last interactive demo](#). Give it a try.

Annoying, isn't it?

If the number of items in the pie menu can change, then a given item will not always be in the same place. And if it's not always in the same place, then you have to visually double-check the menu every time you use it. That means the habit-forming benefits of pie menus are lost.

And here's the rub: the number of items in the Ubiquity context menu is *guaranteed* to change. Not only does the list of applicable commands vary depending on the type of content you have selected, but the total number of commands that Ubiquity knows can grow without bound as the user subscribes to third-party commands. There's simply no way of fixing the number of items in the context menu.

I don't have a good solution to this yet. I have some vague ideas about keeping the most-used items in the main pie menu, which has a fixed number of slices (perhaps eight?), while shunting the variable number of lesser-used commands off to a variable number of hierarchical sub-menus. But it's very much an open question. As always, I would love to hear any suggestions and feedback you have.

Like Be the first to like this post.

143 Responses to "Pie In The Sky"



1. *Lindsey Says:*

[October 28, 2008 at 1:18 am](#)

Interesting article.

Quick question, the pie menu javascript in the demos – did you create it, or was it from somewhere?



2. [Gen Kanai](#) Says:

[October 28, 2008 at 1:34 am](#)

Love the pie menu picker. The vertical list was much more challenging.



3. [Marien](#) Says:

[October 28, 2008 at 1:39 am](#)

Glad to see you spotted RadialContext! Unfortunately you missed my fork that works in Firefox 3 (<https://addons.mozilla.org/en-US/firefox/addon/5739> , shameless plug and all that).

Some random thoughts based on the way I use RadialContext:

- Hierarchical menus work.

- Context-sensitive menus work. One thing that might be worth playing with is some obvious visual clue (menu color?) to tell you which context you got (RadialContext does not do this, but perhaps it should. Would require a bit of graphical redesign to allow me to do this though.)

- Having 8 directions works: moving up, down, left, right or diagonally “between” those basic directions is easy.

(In your drawings you have the separators between areas pointing straight up/down/left/right. I think that’s not quite as easy but I’m very used to how RadialContext works so I could be wrong). I think always having 8 directions but sometimes simply having no action mapped to a direction is better than changing the number of directions (RadialContext usually has 4 or 8 actions per menu, but occasionally some directions are “blank”).

- Having the choices change position automatically is very very bad.

The way I use RadialContext (and the way I think most people who use it for longer than a few minutes do) is as gestures with a ui: for a few common actions I really just push the mouse button, drag the mouse around, and release the mouse button without ever looking at the menu at all. If I need an option I’m not familiar with but that I know is there I start looking at the menu (and if I use that option more than a couple of times I start using it like a mouse gesture too). If options change around by themselves I start doing random scary things I did not want to do (like closing tabs instead of opening tabs).

One thing I’ve seen another implementation (I think it was easyGestures) do is track and show you statistics of which options you use a lot (which you can use to rearrange the menu). I think that’s a better idea than actually having it rearrange the menu for you.



4. [Ryan Scott Scheel](#) Says:

[October 28, 2008 at 2:16 am](#)

I have used (and hated) the radial context extension, when suggested to it by its author. Maybe I just need to change my perception of context menus. Then again, maybe it was just because I couldn’t control the

commands very well in the extension. Make any excess context menus take a single click to open.

So, how about that problem of multiple items crowds pie menu along with the changing direction problem. The constant of 8 per pie menu is a really good idea, with having an ect. piece. See how usable it'd be to have a rectangular context menu come out when you click it (create a demo first) and it'd be nice to test.



5. [*jonoscript*](#) Says:

[October 28, 2008 at 2:48 am](#)

Lindsey: I created the javascript pie menu implementation from scratch. If you want to use it for any purpose, you have my blessing. (I should go add an official MPL.)

Marien: Thank you so much for providing the link to the updated RadialContext. I've updated the article with your link. I just installed the extension and am trying it out now. I might do a full blog post about it sometime in the future.

Thanks also for your helpful feedback on my design problem.



6. [*Justin Dolske*](#) Says:

[October 28, 2008 at 4:58 am](#)

I like pie.



7. [*Blair McBride*](#) Says:

[October 28, 2008 at 5:51 am](#)

Here's one possible way of doing it:

http://farm4.static.flickr.com/3171/2980743126_f67ea260f2_o.png

I really like the idea of hierarchical menus. Although going too deep could be a problem. However, it would also makes sense to have the best suggestions available from the initial menu. So In this mockup, I've combined both.

There's likely going to be instances where there will be more than 8 suggestions in a menu (so as to avoid having a too deep hierarchy). So I'd like to experiment with filtering the entries similar to the way the normal Ubiquity popup does.



8. [*Vladimir Dzhuvinov*](#) Says:

[October 28, 2008 at 6:09 am](#)

I find pie menus psychologically irritating 😬

When I pie menu pops up on the screen I get struck with two challenges:

First, where to start looking from?

Then, in which direction shall I move my eyes?

Rectangular menus feel more comfortable to me because they have a beginning and an end, and scanning them proceeds linearly. I suppose people's brains are mostly wired for linear visual scanning and having to move the eyes in a circular fashion would feel a bit unnatural 😊

From what I understand Fitt's law is only about the purely mechanistic efficiency of movement of the cursor across the screen by the user. Usability, however, is also a function of human cognition.

PS: Thumbs up for the well structured article 😊



9. [Myk Melez](#) Says:

[October 28, 2008 at 6:41 am](#)

I love pie menus, but I think they're more appropriate for a limited set of commands that doesn't change much or at all.

For a larger set of commands that changes regularly (although perhaps not frequently), I would instead investigate a command cloud that appears in a tab (or translucent overlay, for context clouds) and fills the space with command targets, with more frequent/appropriate ones appearing larger and closer to the context point (or the center of the page).

The cloud shares some of the positive characteristics of pie menus while accommodating many more targets. And if changes are gradual enough, users may have time to accustom themselves to them (although that's a mighty big if, this cries out for validation).



10. [Fred Jendrzewski](#) Says:

[October 28, 2008 at 9:35 am](#)

In this case I clearly preferred the pie as the mouse paths are much shorter 😊.



11. [Neil Rashbrook \(SeaMonkey UI Tsar/Czar\)](#) Says:

[October 28, 2008 at 10:15 am](#)

I don't like click-hold menus either way 😊



12. [Michael Lefevre](#) Says:

[October 28, 2008 at 10:38 am](#)

Having tried them in a couple of apps (including the add-on for Firefox), I've not liked pie menus.

I don't usually choose the same command repeatedly from a small, unchanging menu. For frequently used functions, I generally use toolbar buttons or shortcut keys. If I'm using menus, I'm generally hunting for a lesser used command, and having to read around in a circle is harder.



13. *Jon* Says:

[October 28, 2008 at 12:53 pm](#)

I love these Pie Menu's.

Must be so careful to get your items correct though.

If you regularly changed them your users would never begin to remember the gesture.



14. *Albert* Says:

[October 28, 2008 at 12:54 pm](#)

A normal context menu has the advantage of presenting choices in an easy top-to-bottom reading manner. I totally agree that a pie makes 'reading' harder, especially because you have to read 'twice' (once left-down, once right-up or down, or whatever your feeling tells you). Could a solution be something between a context menu and a pie? Like a quarter or a third pie to the left of the mouse cursor, that has 4-6 slices that can be 'read' from top to bottom once, AND have the advantage of being clicked quickly?

Btw. The demo makes it unnecessary hard to select a number in the 'normal' context menu, normally you would have extra horizontal lines, length of text and other visual cues to select the item you want. When I switched Firefox from English to my native language, I noticed that I wasn't reading the text in the context menu but was using the length of the text to determine which item on the list I should be clicking – and these were different of course in another language, making me notice my subconscious behavior 😊



15. *Albert* Says:

[October 28, 2008 at 12:57 pm](#)

Rereading my text: "Like a [...] pie to the left of the mouse cursor" – I should have written: to the right, because that's just like where the normal context menu would appear.



16. *Marien* Says:

[October 28, 2008 at 1:25 pm](#)

Yes, pie menus really do not make as much sense when they change. I think the people considering dynamic menus should play with a modified version of the demo from the blog post that puts something fancier (larger and/or harder to recognize) than 8 sequential numbers in the menu. Scanning a circle is much harder (slower) than scanning from top to bottom. And putting lots of text in a menu like this does not really work (it works fine for the left/right actions, but there is no convenient room for the top/bottom actions unless you make the menu huge. RadialContext puts 4 lines of text to the left and right of the menu, but this makes it hard to match a line of text to a direction quickly (visually)).

Example of actions from RadialContext that I use quite a bit: history back/forward (mapped to left and right), opening and closing tabs and switching left/right (mapped to "northeast" for tab actions followed by up, down, left, right respectively), downloading links (mapped to down on a link or image), "view

source” (“southeast” followed by left). There are others but I think that covers the ones I most frequently use.

When I know the gesture this is really much faster than hitting some other target with the mouse. For example for “close tab” I really just swing the mouse around in something like a half-circle without aiming at anything or having to look at the screen or mouse pointer. I just have to get the general direction right. Having to click a button or tab (given the number of tabs I usually have open) requires much more aiming. Shortcut keys are more similar.

When I do not know the gesture yet I start actually looking at the menu. Like I said before I use them as gestures with a UI: when I know there is a gesture for an action I can look at the menu to find the gesture (instead of having to look through the configuration or a cheat sheet or something). For actions I do not intend to do through the radial menu again I use the regular context menu or other UI.

I think (but it has been a considerable time since I used this so I could be wrong) that <http://vsxu.com/> contains a fancy and surprisingly usable hierarchical cloud menu (it supports arbitrarily deeply nested menus and a surprisingly large number of entries per menu without becoming completely unusable, although finding the right needle in the haystack does become a bit slow). Might be worth looking at for inspiration too.



17. [Ian Thomas \(thelem\)](#) Says:

[October 28, 2008 at 2:05 pm](#)

Sorry – I wasn’t paying attention. I think put up a post with my thoughts and the above link. If I did and it’s just awaiting moderation then please delete this post, if not here is exactly what Marien asked for!



18. [Mark](#) Says:

[October 28, 2008 at 3:12 pm](#)

Click + hold menus are a bit annoying. I prefer click/release + click.



19. [Wilcox](#) Says:

[October 28, 2008 at 3:14 pm](#)

Pies use too much space compared to linear menus.



20. [Sean Phelan](#) Says:

[October 28, 2008 at 3:37 pm](#)

Interesting article. I think your last problem is the worst and hardest to overcome. Plus also the english language is not going to help you here as you have to make the text big enough to read but long enough to understand even in context. Maybe Kanji based languages are better placed for pie menus



21. [Benedict](#) Says:

[October 28, 2008 at 3:38 pm](#)

There are a few problems with pie menu.

1. They obscure the data that is being manipulated.
2. The radial design makes it harder to scan text.

I've only used pie menus a bit (in Firefox 2). I think they probably work well for changing the mode of the cursor. For example, changing tools in graphics or audio editing app.

It's when pie menus are used for issuing commands when the problems arise.



22. [eight](#) Says:

[October 28, 2008 at 3:41 pm](#)

What if the radial context menu consists of more than 7 entries? In Firefox for example I got 14 context menu entries; would that work just as good?



23. [Marien](#) Says:

[October 28, 2008 at 3:49 pm](#)

Ian: Thanks!

Some thoughts: in practice the menu will not be quite as random as this, but there will often be at least a handful of different contexts, so you do occasionally need to scan through the menu. And (at least on this system) something like "Seamonkey" does not actually fit in the menu...

RadialContext uses what it (arguably incorrectly) calls a "tooltip", which is the yellow thing in the second screenshot on <http://www.radialthinking.de/radialcontext/screenshots.html>

As you can see this is easier to scan but it's a bit harder to match the text to a direction once you have found the text you were looking for. If you install RadialContext-mz you can play around with this (there are some options in tools -> add-ons -> extensions -> radialcontext-mz -> options that control when that text shows up and goes away). But perhaps a (much) larger menu with more distinguishable icons would actually work better...



24. [Orvar Orädd](#) Says:

[October 28, 2008 at 3:55 pm](#)

At least with the rectangular menu in example one, you had the option of choosing 8 or null, neither of which were possible to choose in the pie menu.



25. [bharath](#) Says:

[October 28, 2008 at 3:56 pm](#)

Yeah, pretty interesting concept!

I will try to implement it in the next software i develop.

Thanks, and the demo itself is clearly convincing of the concept.



26. [David Griffin](#) Says:

[October 28, 2008 at 3:58 pm](#)

I think you're right about pie menus being under utilised. but i think your problem is not the best example of where they might be useful, like you said they are about muscle memory. In your case i think It's a solution in search of a problem.

you could make the wedges different sizes depending on their frequency of use, this could also help with the changing location problem.



27. [bharath](#) Says:

[October 28, 2008 at 4:00 pm](#)

For albert:

Yes, context menus are easy to read when in usual way(top to down).

But, who needs top read to remember a menu in Firefox or IE or in Windows desktop?

I have most of the menu items memorized.

This pie menu concept reduces the time taken to scroll to the bottom choices, and also looks and feels comfortable.

I have experienced such concept before,but cant quite remember where !



28. [Nathan Gilder](#) Says:

[October 28, 2008 at 4:09 pm](#)

A friend linked me to your article this morning and it immediately reminded me of using Autodesk Maya (3d animation application). Here are a couple images of their pie menus (called the "Hot Box") in action:

<http://img504.imageshack.us/img504/2810/polygonquickmenunothingrx6.jpg>

<http://caad.arch.ethz.ch/info/maya/manual/UserGuide/Overview/images/WorkingMaya.fm.anc1.gif>

These menus can be fully customized by the user – and different pie menus can appear depending on what key you're pressing on the keyboard while clicking the mouse.

This allows me to have a different pie menu while pressing the "k" key than appears while pressing "g", for example.

There are other modifiers too, such as ctrl, alt, etc.

Now, this isn't a replacement for hotkeys, but in applications where things are easily buried under multiple menus, it's a tremendous blessing.

Noting sub-menus on pie menus: Your example of the number of choices being an annoyance was well demonstrated. To confront this issue, what Maya allows is traditional dropdown menus under the top choice on the pie menu.

<http://caad.arch.ethz.ch/info/maya/manual/UserGuide/Overview/images/WorkingMaya.fm.anc7.gif>

This combination is an effective hybrid, in my opinion, compared to pie menus inside of other pie menus.

To me, the best option is to allow users to fully customize the layout of their menus if they choose to. This goes beyond allowing us to set hotkeys.

For example, if you find yourself using a specific menu item frequently, it should be up to you, the user, to move it closer to the menu header rather than digging three layers deep each time. This can require additional development, but in the case of Maya, the developers simply created a framework and give users access to it.

Applications that are more straight forward could benefit greatly by integrating these tested and proven UI abilities.

Thanks for bringing up the formula and, again, the great examples.

Regards
Nathan



29. *Alexmark* Says:

[October 28, 2008 at 4:15 pm](#)

10 bucks says Microsoft steals this idea and puts it in the next version of Office!



30. *me* Says:

[October 28, 2008 at 4:22 pm](#)

Your test is false, it chooses one of the two most accessible locations (left and right of the center) in the pie menu, and uses the middle one on the normal menu.

Also remember you can't use the middle space, or people will accidentally choose it when they click and don't move.

31. *m_eiman* Says:

[October 28, 2008 at 4:28 pm](#)

What about screen edges (or window edges in the case of in-web content)? If you pop up a menu near the edge of where you can draw, you'll need to either move the menu (which kills muscle memory actions) or

lets parts of it be invisible (which kind of makes it hard to read).

A square popup has the same problem, of course, but they're not meant to be muscle memory compatible (althoug some of them are, epecially ones that always appear in the same spot on the screen such as taskbar icon menus).



32. *Jim Gagnon Says:*

[October 28, 2008 at 5:02 pm](#)

Example cases are contrived; by deliberately making the rectangular menu a non-standard size, the user's previous menu experience is negated. Use a system standard rectangular menu and you'll find the user's accuracy practically the same as the pie menu.

Biggest problem with pie menus is that they obscure the target of their actions, making context sensitive choice difficult. Rectangular menus, as they generally appear to the right and bottom of the mouse click allow the user to position the menu so that the object of the action can still be seen. Also must deal with the issue of a pie menu for a mouse click too close to the edge of the screen to be displayed centered on the mouse. Having the computer move the mouse position is bad form — what does a scrolling pie look like?



33. *William Morgan Says:*

[October 28, 2008 at 5:05 pm](#)

Another important problem with pie menus becomes obvious when you must invoke a context menu that reaches the outer boundary of the displayable menu space (usually the side or top edge of the screen) where the menu would be obscured. Rectangular menus can be extended from the left, right, top, or bottom of the cursor as needed. To try moving the pie menu to the side removes all the gestural advantages you mentioned.



34. *atc Says:*

[October 28, 2008 at 5:07 pm](#)

Some interesting points, nicely put too. I think the demo could take some work – mouse over events weren't triggering well enough for it to be a truly representative example of pie menus. Hovering over the 6 meant I had to 'wiggle' the mouse pointer to make it highlight my selection.

Otherwise, an informed and interesting post – thanks!



35. *Caustic Says:*

[October 28, 2008 at 5:08 pm](#)

Well written article, but I'm wondering about how well the radial motion translates to typical motor control.

Mouse gestures are typically wrist motions and I don't believe a circular motion maps all that efficiently or easily. Additionally, the average user doesn't really have the hardware, dexterity and "training" to have fine control of the mouse necessary to make it efficient.

Granted, hierarchical, vertical menus are probably even worse.

Glad to see sharp minds questioning the status quo, though!!

36. [Pie Based Navigation | Sean Walther's Blog](#) Says:

[October 28, 2008 at 5:10 pm](#)

[...] A blog posting from Not The User's Fault, with regards to software navigation and leveraging Pie navigation, Pie In The Sky. [...]



37. [Tim](#) Says:

[October 28, 2008 at 5:28 pm](#)

I was first introduced to pie menus in the game Neverwinter Nights, and it did everything you said. There were eight slices with submenus, and the options changed their meaning depending on what you clicked on. To solve the "too many options" problem they had one option be "more options", I think it was always placed in slice 8. I was a big fan of this menu and I look forward to seeing some use of it in mozilla.

By the way, I got half way through this post when I noticed your name, I'm a ChiPy member too. I hope things are going well for you with Mozilla!

38. [Nathan Peretic Dot Com » Blog Archive » pie menus](#) Says:

[October 28, 2008 at 5:30 pm](#)

[...] "Pie Menus and Fitts' Law," Not the User's Fault [...]



39. [Ian Calvert](#) Says:

[October 28, 2008 at 5:34 pm](#)

I find pie menus very difficult to use, because I can't read them quickly. I can read at least 3 lines of a menu at a glance, usually more (depends on the spacing), but I can only read one item of a pie menu at a time. I don't have to build up deep familiarity with a normal menu to be productive, and when I have I'm not noticeably faster with pie menus because I'm pretty accurate and fast with a mouse.

This all makes them feel like treacle to use.

Also, I can't use keys to navigate them as obviously. Up? Which direction is that? Clockwise, or dependent on position?

To me they only solve a problem when you have a small number of large icons, and the list doesn't change regularly. I can work with large amounts of text in small spaces much more easily, and cope with changes (move mouse down becomes move mouse down more, not move mouse down becoming move mouse left).

I disagree with Fitts law, too, since it ignores the aspect of expectation. If I know that a regular menu will appear, I can already start moving my mouse to the centre before I've parsed any of the text. I can start moving it to the top for an obvious action, or the bottom for a more technical/less commonly used one. By the time I've parsed the text, I can already have my cursor over the "properties" section, for instance.



40. *Jeremy* Says:

[October 28, 2008 at 5:40 pm](#)

A "fix" for the changing items would be to simply make it a priority queue, the often-used items stay in the same places, and the less-used get bumped to the sub-menus. This would also make sense since the lesser-used items probably won't be in the users' spatial memory, so they'd have to be looking for it anyway.



41. *H.A.M.* Says:

[October 28, 2008 at 5:44 pm](#)

Mmmmmm, I like Pie.



42. *Shahid* Says:

[October 28, 2008 at 6:07 pm](#)

It might be worthwhile playing a bit of Neverwinter Nights. That had a pie implementation, but because the menu options changed so much, it didn't really work out. However, I have used pies in the past, and I love them as a concept. For common commands they're awesome – I would consider them more guides for simple mouse gestures than anything else. (Mouse gestures also work great for a limited and commonly repeated set of commands, like the browser back button.)



43. *icandothat* Says:

[October 28, 2008 at 6:43 pm](#)

Is it just me? This doesn't seem to work at all in either FF or IE... I don't get it.



44. *icandothat* Says:

[October 28, 2008 at 6:46 pm](#)

OIC FF3 only. hmm



45. *atlas* Says:

[October 28, 2008 at 6:48 pm](#)

while yes it is easier to navigate using the pie wheel of awesomeness thingy – to me at least – its easier to absorb more information more efficiently from the regular rectangular one. more info can take up less space, communicating it to the user more efficiently.

the pie wheel is great for things that don't need any description or can be represented using only a small graphic, but it far from making the rectangle thing obsolete.



46. *TMcD* Says:

[October 28, 2008 at 6:52 pm](#)

I very much like this. I wanted to point out that doing various things such as quickly mousing away from the pie menu screen and then coming back and clicking an item will result in a Null return, however doing this on the list menu will result in a return of 0. Just thought I would throw that out here. I want to reiterate how much I like this.

47. [joe preston](#) Says:

[October 28, 2008 at 6:58 pm](#)

See I wouldn't bother with showing the whole pie. I think one quarter of the pie tucked into the bottom left corner of screen or app would work just as well, it could be controlled by arrow keys or the scroll wheel to rotate through the available options. In fact if you only show a section of the pie, then you are not truly confined to ANY space restrictions on what goes in the pie. If you only show a section of the pie, then any breakout menus, whether a heirarchical pie (although I prefer the "exploding" pie chart for a submenu here, but the submenus appear in a fixed position every time that never obscures the original menu. Interesting article



48. *chmod777* Says:

[October 28, 2008 at 7:07 pm](#)

3dsmax has used a menu system like this for its' context menu for years. after using max, and then heading over to photoshop, i find that i have forgotten how to use 'normal' right click menuss

49. [Eric](#) Says:

[October 28, 2008 at 7:20 pm](#)

It would be interesting to see peoples response times for each menu.

50. [AlphaQuant](#) Says:

[October 28, 2008 at 7:35 pm](#)

What about keeping a constant number of slices in the pie but having the slices that you see be part of a "staircase", so basically each slice is like a step of a staircase that you are looking down on from above.

You then need a way of walking up or down the staircase and the way I would see this is that you have a very narrow slice that if you move across it in a clockwise or counterclockwise fashion would move you up or down the staircase respectively. (For those of you that have taken a course in Complex Analysis another way to look at this is like branch cuts for representations of a Riemann Surface and every time you move across the branch cut you move onto another part of the surface [sorry if I messed up the terms here, it's been a while].) This should also give you some fairly standardised mouse gestures which would be a plus.

I wouldn't see this as replacing hierarchical menus, but rather as a way to deal with more than the standard number of items at each level of the hierarchy in your menu.



51. [Mark Esler](#) Says:

[October 28, 2008 at 7:38 pm](#)

Ok.. so when you put this on your site wouldnt the overlay of the right click go over all other links? So if you had a link in your content or logo it wouldnt work right?

Another issue is how would you get the user to use it? For the techno savvy its not an issue but a lot of people dont like new things and are used to normality (not saying that we need to be confined to the old, but how would you make them transition)

this would be sweet for a minimalist design... imagine just this big white glossy and slightly beveled box in the center of the browser. It could have such a minimalistic clean look if it were easy to implement.

props



52. [James](#) Says:

[October 28, 2008 at 7:39 pm](#)

This is a good idea.

It would be very hard to implement in certain cases though. You need to start off with a very strong ui design.

53. [Miramon](#) Says:

[October 28, 2008 at 7:48 pm](#)

There have been many radial menus used in a wide variety of PC games over many years. Bioware does seem to like them, but so do various other studios.

For the most part (with a few admitted exceptions) they have been pretty crappy. This is not necessarily an indictment of the concept, but of the implementation. The game industry has more not-invented-here syndrome than any other, and a lot of mediocre UI and controls designers.

54. [glen](#) Says:

[October 28, 2008 at 7:56 pm](#)

just like second life

55. [O'Maolchathaigh](#) Says:

[October 28, 2008 at 8:01 pm](#)

Well, as a long-time computer user, but not a programmer (unless you count Basic and Fortran), I like the pie menu much more than drop down or other linear menus. I think, obviously, that the number of slices or cells must remain constant and in the same order. Because the number of commands varies, the idea of keeping a fixed number of slices (8 is good) makes perfect sense. There then has to be a way, as through the center, to access sub menus, or even know that there are sub menus or other choices. Perhaps only the left click produces the main menu, and sub menus are called up by right clicks? Or vice versa of course. There's that center button too for PCs. But, I don't have the educational or experiential knowledge to make a really viable suggestion. Just thought I'd chime in because this was interesting.

56. [yoospi](#) Says:

[October 28, 2008 at 8:43 pm](#)

I found vertical/rectangular much easier to use than the pie. Perhaps because my mouse chose to keep skipping over certain pies in the pie chart. But for me, it was much easier to hold and drag down.

57. [UX buzzspree » Blog Archive » Pie anyone?](#) Says:

[October 28, 2008 at 8:53 pm](#)

[...] Pie In The Sky « Not The User's Fault has an interesting article on pie menus. They capitalize on Fitts law (which basically says bigger, closer targets are better for usability). These seem especially good for manipulation of objects, such as a photo, that the user might click on in the course of workflow. The "pie" visual shape is not a forgone conclusion. The idea of controls/choices appearing around a central touch point is. [...]

58. [Dave](#) Says:

[October 28, 2008 at 8:53 pm](#)

Just a couple of thoughts.

If you put the text for the currently highlighted section of pie under the menu then you'd be able to fit more items in before things get crazy. And if you made the cursor graphic in the middle a "back" button you could do hierarchical menus by just drawing a new menu in the same position.

Not sure how that works with any HCI theories, peoples experiences or whatnot.

59. [Stan](#) Says:

[October 28, 2008 at 8:54 pm](#)

The rectangular menu was much easier to select the same number multiple times in a row. The pie menu wouldn't select the same number twice without sliding to a different number, then coming back to the number you really wanted to select which made it very frustrating.

60. [Braydon Fuller](#) Says:

[October 28, 2008 at 8:58 pm](#)

Using pie menus is a good way for Enso/Ubiquity to stay quasimodal. It also makes knowing what is available so much easier. 😊

61. [Pie menus - Tom on HCI](#) Says:

[October 28, 2008 at 9:00 pm](#)

[...] spoke in class the other day about pie menus, so I thought I'd share this article I found about [...]

62. [Roy](#) Says:

[October 28, 2008 at 9:01 pm](#)

I have Firefox 3.0.3. With the circular selector, when I went directly to '6' it would not select. I had to go to '5' or '7' and then move to '6' to get it to select.

The stack worked as it should.

63. [Mike](#) Says:

[October 28, 2008 at 9:07 pm](#)

I am from a time when computers were powered by coal. If you guys knew a software called TIPS, from Crystal Graphics, you will see a kind of pie menu into action. TIPS used floating squared menus, like a box full of icons that appeared where you clicked and had sub boxes for some functions. The speed of operation of TIPS, a graphics package, never were matched by any other software on the planet. It seems people forgot how to do a software. Crystal had other two packages called RIO and TOPAS, the best vector and 3D packages ever made for DOS. Creating a 3D Bevel structure on TOPAS was a piece of cake. You created it on the fly, live. TIPS, RIO and TOPAS, specially the first two, had superb interfaces. This thing of a fixed menu that shows vertical boxes sucks to infinity + 1

64. [rickcannon](#) Says:

[October 28, 2008 at 9:19 pm](#)

I didn't really read any of this post. I just saw "if you're using Firefox 3, please..." so I tried it.

I have to say, I prefer the circle pie menus. But these menus are pretty awesome, regardless.

65. [*Rob Farrow*](#) Says:

[October 28, 2008 at 9:31 pm](#)

Great Stuff!

66. *anon* Says:

[October 28, 2008 at 9:34 pm](#)

But you can't use them without paying a lot of money. Pie menus are patented:
<http://www.freepatentsonline.com/6549219.html>

67. *james* Says:

[October 28, 2008 at 9:35 pm](#)

MYTH.

The secret to success in software has NOTHING to do with usability. It has everything to do with marketing and market penetration. Microsoft is not successful because of quality software, they are successful because they know how to leverage product recognition and branding. Apple is not successful because their products are more usable, they are successful because their products are more fashionable.

68. *Jacob Beauregard* Says:

[October 28, 2008 at 9:47 pm](#)

“There's simply no way of fixing the number of items in the context menu.”

Two sorely obvious solutions, though I guess if they were obvious someone would have found them by now:

1. Having different sets of pies that you can rotate through
2. A method of rotating (a slice of the pie, clicking the middle, or mouse wheel)

Then you're only stuck with figuring out how many items is the best number of items (I'd pick the number of people an average manager manages if I had no other basis)

Positioning against screen boundaries would be a bit of a harder problem imo, particularly with the hierarchical pie menu concept. If only computers were spherical, right?

69. *quazy* Says:

[October 28, 2008 at 9:51 pm](#)

The real answer is a shitload of intuitive quick-keys.

Circles are ugly inside a rectangular frame. I'm amused though.

70. [PIE MENUS - Why pie menus should be in every user interface - The WebZappR](#) Says:

[October 28, 2008 at 11:12 pm](#)

[...] MENUS – Why pie menus should be in every user interface submitted by jonw to programming [link]
[247 [...]

71. *ehskoro* Says:

[October 28, 2008 at 11:13 pm](#)

Ooo Wow. I really liked the circle. =D

72. *Missus X* Says:

[October 28, 2008 at 11:21 pm](#)

I actually found the dropdown menu quicker, but if it was greater than the number 6 I would have been in trouble..

73. *stevenet* Says:

[October 28, 2008 at 11:22 pm](#)

I found the pie chart more difficult to use, but I suspect that is just because of a familiarity with the vertical menu format. In theory, the pie should be easier, so with more usage, it might be.

74. [orchidellia](#) Says:

[October 28, 2008 at 11:27 pm](#)

I find that when using a trackpad, the pie menu is more difficult to use. The rectangular menu is easier with Apple's gesture trackpad.

But, when using a mouse, the opposite is true. I guess preference not only depends on a person's personal preference, but what input device they are using.

75. *Chris Hills* Says:

[October 28, 2008 at 11:43 pm](#)

I think that you could apply the Dasher input method to the pie menu – starting from the center and working outwards to navigate not just a single menu but many sub-menus as well, with a minimum of effort.

76. [*Diego Méndez*](#) Says:

[October 28, 2008 at 11:45 pm](#)

Pie is nice, but to me it wasn't possible to chose "6" again just after chosing it once. If I wanted to chose it again I had to move over "7" or "5" and then into "6" to make it work.

In rectangular menus, I was forced to pass over all choices to get to 6, so chosing 6 again was not a problem.

I'm sure it's possible to modify the script so you can chose 6 two or more times in a row (directly, without passing over other choices first)... in such case, and only in such case: Pie would be nicer!

77. [*Ibrahima*](#) Says:

[October 29, 2008 at 12:06 am](#)

Interesting read. Microsoft Research has experimented with pie menus too, with the InkSeine app for tablet PCs. <http://research.microsoft.com/inkseine/>

Unfortunately I think it's a bit buggy and hasn't been updated in a while, but I think with the move to touchscreens pie menus might become more prominent.

78. [*K*](#) Says:

[October 29, 2008 at 12:08 am](#)

Yeah, but where is the cross-browser implementation?

I am completely boggled why I haven't seen a comment on this, but maybe I just scanned over it.

All you have is an example for FF3. If there isn't a real world example of this working on across browsers (including IE6 *shudder*) then why are you surprised that it hasn't been implemented more widely?

79. [*Jason Freeman*](#) Says:

[October 29, 2008 at 12:25 am](#)

Hierarchical menus could be supported by presenting concentric rings which appear when an item is selected. The inner ring would remain highlighted as the user cruises around the outer ring.

Also, don't overlook the use of the mouse-wheel either. A menu could be presented with the "up" item selected, then, rather than making the user select his choice through hand motion (yes, gestures are nice) the wheel spins with the mouse-wheel.

Coupling these features (concentric rings, and the mouse-wheel) could make complex hierarchical context menu selection fast, visually interesting and possibly fun.

80. *Sanz Says:*

[October 29, 2008 at 12:33 am](#)

Maya has had pie menus for well over 10 years and they work quite well. I've always wondered why they didn't pick up more mainstream use.

81. *Braydon Fuller Says:*

[October 29, 2008 at 12:35 am](#)

Here is a JavaScript implementation of Pie Menus:

http://git.braydon.com/gitweb.cgi?p=js_radial_menu;a=summary

Here is a demo:

http://git.braydon.com/gitweb.cgi?p=js_radial_menu;a=blog_plain;f=index.html;hb=master

82. *Matt Says:*

[October 29, 2008 at 12:47 am](#)

Either Fitt's law is wrong or you explained it incorrectly. Based on your diagram W and D would be the same regardless of how many options there are in a menu. You also said that a and b are outside the programmers control. Therefore, from what you've said it should be as easy to select one out of 6 choices as to select one out of 60 choices. This pretty clearly isn't the case.

83. *The Ready Set Robot » Blog Archive » Fitts' Law and the Pie Says:*

[October 29, 2008 at 1:37 am](#)

[...] Jono at Mozilla Labs has a wonderful blog post on pie based menus, But pie menus are conspicuously absent from most popular UI libraries and toolkits. They're not in the Human Interface Guidelines of Mac, Windows, Gnome and KDE. [...]

84. *Robert Prije Says:*

[October 29, 2008 at 1:54 am](#)

Regarding the problem of too many items in the menu:

How about doing something similar to how many combo-boxes work?

In a combo box, you click on the box and, say, 8 countries come up with a scrollbar to browse through the rest.

You could imagine something similar for pie menus. A pie menu with 8 countries in 8 wedges with a 9th wedge enabling clockwise and anti-clockwise scrolling.

85. *b mcmillan* Says:

[October 29, 2008 at 2:39 am](#)

Wow – we started implementing a pie menu control just this past friday for a large enterprise application. The timing of your blog is great 😊

My design choice was deliberate. 1) There will only be a few (3-5) choices, 2) they will not change, 3) speed is a major concern, 4) habits are bound to be formed due to the heavy usage expected (500-600 people, 8 hours / day / person, everyday for 1-10 years), and 5) it will not be hierarchical. About the last point, however — we are implementing the control in a standard controls library, and as such will be supporting hierarchy. We are researching using pie sub menus (like RadialContext), plus reserving one slice of the sub menu, in the direction of the parent menu, for going back “up” to the parent’s menu, with proper animations.

The screen edge problem, however, is interesting. Thanks for pointing that out.

One interesting problem, however, is that we’re doing this in WPF. 😊 Wish us luck ... good luck with your implementations!

Thanks for the article.

86. *Jon* Says:

[October 29, 2008 at 3:29 am](#)

Very cool. The one problem that isn’t addressed in any of the demos is that of fitting the text onto the pieslices. Your’s don’t need to occupy that much space since they’re only single and double digit numbers.

Love the idea though, I hope it gets adopted more.

87. *Junior Holanda* Says:

[October 29, 2008 at 7:07 am](#)

Poderia traduzir para o português?!!!!

88. *Menus radiales en Javascript con Canvas | aNieto2K* Says:

[October 29, 2008 at 7:44 am](#)

[...] por del.icio.us, he caído en Not The User’s Fault y más concretamente en un experimento de menú radial desarrollado sobre canvas con javascript que nos permite testear la posibilidad de implementar este tipo de menus* en [...]

89. *Swen* Says:

[October 29, 2008 at 8:40 am](#)

I do like the idea of a pie context menu, but what would happen if the context menu was invoked on the extreme sides of the screen. In case of the rectangular menu it would just render to the other side.

If that was to be used for the pie then the pointer wouldn't be in the center anymore and with that goes the advantage of knowing the direction of the menu item.

90. *AsA* Says:

[October 29, 2008 at 9:07 am](#)

I havent red all comments, perhaps somebody said it before. Also i wont explain much, just want to say: I would put the first entry of the menu on the left or on the upper left, not at the top.

91. *celebsip* Says:

[October 29, 2008 at 10:00 am](#)

See the demo for IE here – <http://reviewk.techwatch.com>
Enjoy!

92. *damien* Says:

[October 29, 2008 at 11:40 am](#)

Pie menus have been known forever, but they have never quite taken off.

My take on why they have'nt taken off comes down to a mismatch between text items in a menu, and the radial nature of the pie.

For example, my Google Chrome browser has the following context menu (which is relatively short compared to the one that Firefox offers):

- Back
- Forward
- Save As...
- Print...
- View Page Source
- View Page Info
- Inspect Element

Note the different lengths of the options – how would you display these differing sizes of options in a pie menu?. Your examples are great until you try to integrate text into them.

Where I have seen pie menus work is where there are a fixed small number of options represented by icons.

A standard menu, however, can easily adapt to a changing set of items of differing rectangular sizes.

93. [Vic \(Ms. Place\)](#) Says:

[October 29, 2008 at 11:46 am](#)

Once you figure out how to do it, duh, the pie is easier.

94. *Steve Bennett* Says:

[October 29, 2008 at 11:59 am](#)

Maybe I'm doing something wrong, but it was much easier with the straight menu. For some reason, with the radial one, I had to move the mouse off the 6, then back onto it for it to light up. No such problems with the straight one. FF 3.03.

Fwiw, what I don't like about radial menus: there's no clear order to look for choices in. It's easy to remember rules like "Properties is always at the bottom". 1 dimension is sometimes better than 2.

95. *eskimokaka » links for 2008-10-29* Says:

[October 29, 2008 at 12:04 pm](#)

[...] Pie In The Sky « Not The User's Fault Pie menus are cool, very cool. (tags: ui, usability, design, programming) Nog geen reacties. Dit is uw kans om de eerste en misschien wel de enigste te zijn. Go champ go! [...]

96. *Gary* Says:

[October 29, 2008 at 12:26 pm](#)

A choice of 8 should be enough, it should be standardized, then if 9 comes along it would have to take the place of 1 of the 8.

97. *jasper1201u* Says:

[October 29, 2008 at 12:35 pm](#)

love the pie menu! /so/ much easier than the list menu.

98. *mz* Says:

[October 29, 2008 at 1:18 pm](#)

Thanks for introducing me to pie menus — it seems like an interesting concept. I haven't had time to read through all the comments, so I apologize if someone has already posited the following solution (for a variable number of menu entries).

Why not subdivide the menu into discrete chunks (quadrants, maybe?), and then put multiple entries in those quadrants (still displaying their names in the base menu). If a new menu item becomes available, add it to the quadrant that has the least number of entries or use some predetermined order in the case of a tie. The key to this idea is that moving to a particular quadrant brings up a new pie menu with just the entries

from that quadrant in it. Repeat until there are 4 or less entries in the menu. Then each entry can bring up a submenu, if it has one, and start the whole process again.

I claim that this method creates a repeatable mouse gesture for any variable number of elements. The only issues are running out of space to fit the type and hitting a screen edge.

I guess you could have a “...” entry or something if you run out of space to fit the words. A screen edge seems like it would (assuming you keep the menu on the screen) shoot you all the way down to a “leaf” of the menu. So maybe you could make the pies smaller and smaller — limiting to size 0 as one approaches the edge of the screen? I’m not sure about that part as it might make the gesture a little more dependent on the starting point of the menu, but the directions at each point in the gesture would still be unique. It might also make things really illegible if you start close to a screen edge. I guess jumping the menu (and mouse) to the middle of the screen might alleviate things?

Anyway, that’s my \$0.02...

99. *eleventyurple* Says:

[October 29, 2008 at 1:58 pm](#)

This is great, BTW. One issue I had with the rectangular option was that using the particular machine I’m on right now with a lower resolution, a few times the 6 and higher options were off-screen and my selection was merely a guess unless I started my selection by clicking higher in the selection window. The pie option eliminated this issue.

100. *perimotocross* Says:

[October 29, 2008 at 2:16 pm](#)

metros en mi blog <http://www.perimotocross.wordpress.co>

101. *perimotocross* Says:

[October 29, 2008 at 2:17 pm](#)

metros en mi blog <http://www.perimotocross.wordpress.com> sorry

102. *Violet* Says:

[October 29, 2008 at 2:20 pm](#)

pie
used both my regular mouse and the touch pad on notebook.

103. *Razee Marikar* Says:

[October 29, 2008 at 2:23 pm](#)

Hey, I just got an idea... Pie menus could be used for something like the most used features / most recently used features and limited to 8 (without an etc option). This should be available on right click + drag (ie like a gesture, just like you've implemented in the example). But on release without moving the cursor over a menu item, show the standard, rectangular menu. Additionally, there could be a delay (of say, 0.5 sec) before showing the pie menu; but would be displayed as soon as a drag of > 2px occurs (encouraging the use of 'gestures' as soon as people start trying out pie menus). I think this will enable a smooth transition from conventional menu to pie menus, and always leaving an option to use the conventional menu.

104. [*Habitual Thinking, An Obstacle*](#) Says:

[October 29, 2008 at 3:01 pm](#)

[...] Jono's brilliant post, I wasn't even aware of the existence of pie menus. Pie menus are more convenient when there [...]

105. *mz* Says:

[October 29, 2008 at 3:36 pm](#)

So I think I solved the other issues I was stumbling on in my last post:

Instead of opening a new pie at the new mouse position, jump the mouse back to the center of the pie, and then move (animate?) the entries in that quadrant so that they expand to fill the pie. If you only expand them so that they take up 3/4 of the pie, the quadrant in the opposite direction from the one that just got selected can represent a "back" movement. That way, the user can negate a selection by just moving the mouse in the opposite direction.

I think there's also a subtle hitch in that you can't just tack on new entries to the end of a quadrant. You'd have to recursively cycle through the quadrants "within" the quadrants to make sure you keep unique mouse gestures, if that makes any sense. Slightly difficult to describe, but it doesn't seem like it would be too terrible to implement.

106. *theworkshop* Says:

[October 29, 2008 at 3:56 pm](#)

pie menu simply rocks. i wonder when developers will get serious about them

107. *dianne1018* Says:

[October 29, 2008 at 3:58 pm](#)

nicely done!

108. [*Stephen Rees*](#) Says:

[October 29, 2008 at 4:05 pm](#)

In am using Firefox 3.0.3 in Ubuntu – I cannot see any charts on that web page

109. [*Caoticamente, pieces of me » Blog Archive » La legge di Fitts*](#) Says:

[October 29, 2008 at 4:08 pm](#)

[...] articolo invece è un esempio dell'esatto contrario, ovvero l'applicazione di un noto principio [...]

110. *Ron* Says:

[October 29, 2008 at 4:29 pm](#)

Very interesting article! Thanks. (And thanks to codepo8 for the twitter to catch it.)

Some alternatives/extensions worth considering might include a 'dartboard' menu where you can add bulleye bands to the pie ... or a much simple to code 'calendar' style menu.

I doubt I would want 31 commands in a menu, but if I really really needed 12 then a 3 by 4 calendar grid might be a reasonably 'Fitt' solution, eh?

111. *MLR* Says:

[October 29, 2008 at 4:39 pm](#)

While the author of this blog is enthusiastically applauded for wanting to make life better for end users, much of what I've read in the blog and replies is focused on the software or hardware, not the person! I think one or two people mentioned that gestures were done with the wrist, and several talked about scroll wheels or other buttons. But what needs to be clarified is that not all mouse motions are done with the wrist. Some are with the wrist, such as short side-to-side motions, but others are done by extending or curling the fingers (such as short forward or backward motions). Then there's medium-distance gestures that require rotating from the elbow, which allows for most of this plus diagonal movements but is limited within the fixed arc of the forearm length. For forward (or up-screen) motions beyond what the fingers can do by extension, as well as long-distance movements, the whole arm must move from the shoulder joint. Consider also that not everyone consistently uses proper mousing posture, so some rest their wrists and compensate in other ways, some "float" the hand over the mouse while others make full palm contact, mice are shaped very differently which impacts gestures, etc.

I also want to ask just how much time are we talking about saving? If it's milliseconds or microseconds per action, and over the course of a normal 8-hour workday we're talking the difference of maybe a minute, is it worth it? Then, considering the savings is almost guaranteed to be offset by a learning curve, decremented readability, errors, etc. I can understand why this hasn't been utilized as widely as one might hope. I find the idea very intriguing, so I recommend designing a controlled empirical study that can help you determine what types of applications can benefit from this line of thinking and what types cannot. Then, uncover what it is about that subgroup that lends itself to these menu types, and choose a development path based on those needs.

Remember, the needs of the people drive successful development, not the other way around! Good luck to you.

112. *David* Says:

[October 29, 2008 at 6:46 pm](#)

It's important to note that pie menus act like mouse gestures because the menu appears as soon as the mouse button is clicked, allowing one to choose a slice of the pie before letting go of the mouse button. Windows' context menus, on the other hand, appear only after one has completed the click. In other words, your pie menu is quasimodal, while the Windows context menu is modal. (I don't know what OS X does, but GNOME's context menus function both quasimodally and modally.)

Being only quasimodal means that it's fairly easy to dismiss the menu by un-clicking, either at the centre of the pie or at the edge of the screen, both of which would be fairly easy to reach, due to Fitts' Law. Consequently, you can make the pie even bigger (as long as it doesn't touch the edges of the screen), without worrying that it covers up information unnecessarily (especially if the menu is a little transparent, *a la* Ubiquity). This, in turns, allows you to overcome the problem of too-many-thin-slices by increasing the size of each slice (and, thus, the value of *W*). Alternatively, more entries could be added around the pie, forming concentric circles. Alternatively, the "slices" could be re-positioned as segments of a spiral (a 'Spiral Menu', if you will).

I especially like my spiral menu idea, as it allows "slices" to be quite large when there the menu is sparsely populated, like the pie menu, while allowing more items to be added fairly easily by tightening the spiral. This also creates a unique ordering for menu items, solving the readability problem mentioned in other comments. Finally, this allows more frequently-used items to be placed closer to the centre of the spiral, again taking advantage of Fitts' Law.

As for making the menu fit Ubiquity's needs better, someone mentioned [Dasher](#)'s text-input mechanism. I hope you will explore that, as it seems quite promising. There's also Quicksilver's [Constellation](#) mode, which seems like an obvious precedent for the type of thing you're trying to do.

I like the buffer zone in the centre of the pie, which is obviously meant to prevent accidentally landing on an undesired slice when clicking (similar to the problem of Double Dysclicksia). However, since the closest point is always exactly where the pointer is, it would be useful to add another entry to the centre of the pie instead of a buffer zone—but only when it makes sense to have a default action. For example, clicking on a link would show the pie menu, with whatever possible actions, including 'Follow this link' in the central circle, instead of the buffer zone. This way, one could easily activate the default action for things like links and buttons (and anything verb-like), while being able to choose alternate actions easily. Essentially, left-click and right-click would be merged into one. Standard left-clicking would be unaffected, whilst context menus would be more discoverable and accessible. Right-click would, consequently, be freed up from its undiscoverable role as hiding place for any and all hidden functionality. It could then be usurped for a more humane usage, namely, 'Select'—just as Jef Raskin wanted!

Of course, this makes sense only if Ubiquity completely takes over Firefox's context menus. Hence, my question: *Is the goal here to create a separate menu for Ubiquity, or to replace Firefox's context menus wholly with Ubiquitized pie menus?*

113. [Ronnie Brown » Blog Archive » Are Pie Charts the Future of Web Navigation?](#) Says:

[October 29, 2008 at 7:42 pm](#)

[...] from Mozilla Labs has a cool experiment going on over on his blog. As he says, try out the demo (if you're using Firefox 3.0 and above) and see which is [...]

114. [Olivier G.](#) Says:

[October 30, 2008 at 9:14 am](#)

To handle the “too many items” problem, you could implement multi-layered pies, like in http://www.neoformix.com/2006/MLPC_p3.gif

115. [jenamoney](#) Says:

[October 30, 2008 at 1:21 pm](#)

COOL..... JUST KIDDING NOBODY CARE

116. [James Cole](#) Says:

[October 30, 2008 at 1:34 pm](#)

Reading this (and some of the comments) made me think of a kind of middle-ground between pie menus and ordinary menus that involves a linear menu with a technique for radial selection.

I have no idea whether these “Fan menus” have been suggested before, but anyway I’ve described them (with some pictures) here – <http://explorerstreet.blogspot.com/2008/10/fan-menus-for-user-interfaces.html>

117. [Leo Horie](#) Says:

[October 30, 2008 at 3:33 pm](#)

I take that you’re starting from the assumption that the user can read vertical text and/or decipher the meaning of icons and/or they are comfortable reading things in a clockwise manner, when they visually scan a pie-style menu. I’m not sure that’s a great assumption.

Personally, I find scanning menu options downwards easier than clockwise... some circular UIs that come to mind are LucasArts’ Full Throttle action menu that was difficult to recognize at first (I couldn’t even tell how many menu items there were), and Trickster Online’s social menu, which I find incredibly non-intuitive (I haven’t memorized it to this day). Maya has a multi-level circular menu, but I found myself looking for the keyboard shortcuts after a while.

All in all, my experience with circular menus haven’t been very positive. On top of that, people are normally fairly wary of changes to fundamental UI elements, because of muscle memory, familiarity/consistency with other applications, etc.

I feel pie menus works well when there are 4 or less options, but once you need to organize a ton of things in hierarchies, the circular approach starts to look clumsy and I find myself fighting against mouse precision.

118. [David](#) Says:

[October 30, 2008 at 4:45 pm](#)

I prefer keyboard shortcuts for *all* commands. I memorize them and I’m a lot more productive that way. If at all possible I avoid the mouse entirely. That said, I guess I don’t care what type of context menu is used as long as each and every command has a keyboard shortcut.

119. [JeremiahTolbert.com » Blog Archive » links for 2008-10-30](#) Says:

[October 30, 2008 at 6:09 pm](#)

[...] Pie In The Sky « Not The User's Fault (tags: webdesign userinterface ui userexperience) [...]

120. [James Cole](#) Says:

[October 31, 2008 at 1:33 am](#)

(i posted this comment yesterday but it doesn't seem to be showing)

Reading this (and some of the comments) made me think of a possible middle-ground b/ween linear and pie menus, that presents a linear menu but a radial means of selecting items within it. I've got no idea whether anything like this has been suggested before, but anyway I've written it up (with pictures) here:

<http://explorerstreet.blogspot.com/2008/10/fan-menus-for-user-interfaces.html>

121. [Sergey Shelukhin](#) Says:

[November 1, 2008 at 8:17 am](#)

I evaluated pie menu extension a while ago, for a week.

The problems are

- 1) Realistic count of items is 8.
- 2) Navigating submenus (both rectangular and pie) just sucks, hence 1 is crippling.
- 3) There's no way to represent text
- 4) Navigating dynamically changed menus is PITA. With rectangular, your knee-jerk item selection may be 1-2 items off, with pie, you have to move mouse in different direction – if not to different submenu.

122. [chrisjf](#) Says:

[November 1, 2008 at 10:47 pm](#)

First off, thanks for an awesome blog post!

Second, I had originally started writing a comment but it became extremely unruly and since it has morphed into a full-on essay complete with mockup! Head on over here to read it:

<http://chrisjf.blogspot.com/2008/11/hybrid-radial-context-menu-for-firefox.html>

123. [Gene](#) Says:

[November 3, 2008 at 1:58 am](#)

I was gonna send this privately, but I can't find your contact info.

Your article inspired me to try a different kind of navigation for my website. I've often felt the vertical list was too cluttered and didn't draw people in to click.

I've created a brief mock-up of a new navigation for my page using a radial... it's a design I'm just not finding really being used much on the web (and there may be a reason for that).

I can already tell you I'll rotate the wheel 1 "notch" counter-clockwise to make "Listen" stand out better, and I also want to make the purples better match the purple glow of my logo.

Nevertheless, I'd love to get your feedback, good bad or ugly, to this:

<http://www.blacklightradio.com/radial.html>

Thanks for the inspiring article!

124. [Patrick Keenan](#) Says:

[November 4, 2008 at 11:02 pm](#)

Why not have it adjust as the user moves the mouse. Closer areas expand further areas contract. Also why not use different corners as containers for different pie menus. The bottom left corner of the browserr could be recent, the top right could be custom, the top right new stuff, etc

125. [A Digital Outrigger / Library Usability Links 11/7/08](#) Says:

[November 7, 2008 at 7:44 pm](#)

[...] Very interesting write-up on Radial Pie Menues [...]

126. [Radial Menu Tests in Firefox | Konigi](#) Says:

[November 7, 2008 at 11:01 pm](#)

[...] Menu Tests in Firefox Testing of proof of concept prototypes of radial menus. URL jonoscript.wordpress.com/2008/... [...]

127. [A few announcements. Articulate Announcements. « Not The User's Fault](#) Says:

[November 8, 2008 at 1:11 am](#)

[...] Articulate Announcements. November 8, 2008 1. If you tried out the pie menu demo in my previous post, you might (depending on which day you tried it) have run into a bug where you couldn't [...]

128. [Nathan](#) Says:

[November 9, 2008 at 5:01 am](#)

The video game Sacrifice also made use of Pie Menus for selecting spells. I really liked it.

One solution to the problem of habituation is to have fixed positions for categories. Say the top half of any given Pie Menu is made up of selectable items, and the bottom half is made up of categories. Intelligently design the categories and the top half of each menu can stay fixed, while the bottom half just branches into ever-increasing complexity.

129. [Pie Menu evolves into Grid Menu! « Not The User's Fault](#) Says:

[November 9, 2008 at 6:07 am](#)

[...] Menu evolves into Grid Menu! November 9, 2008 Response to my post about pie menus was overwhelming! There were way too many comments for me to reply to each one individually, so [...]

130. [Formatvorlage» Blogarchiv » Pie Menus und Baby Enten](#) Says:

[November 9, 2008 at 1:53 pm](#)

[...] Menüitems statt einer vertikalen Anordnung. Das Not The User's Fault Blog hat dazu einen sehr interessanten Artikel mit einigen Beispielen und einer ausführlichen Diskussion zum Thema. Die mangelnde Verbreitung lässt sich – neben den im [...]

131. [Nowości 9. - 10.11.2008 at livio's way](#) Says:

[November 10, 2008 at 10:59 pm](#)

[...] do niedawna obstawała przy menu kołowym. Podpierała się prawem Fittsa, przedstawiała szkice, różne demo, a ludzie się [...]

132. [November Labs Night, Thunderbird Awesomeness at Toolness](#) Says:

[November 14, 2008 at 4:00 am](#)

[...] kicked off the lightning talks by presenting his explorations in the land of pie menus. This was followed by a presentation by Alex Peake on his new [...]

133. [New MinimalComp: WheelNav | BIT-101 Blog](#) Says:

[November 19, 2008 at 2:53 am](#)

[...] <http://jonoscript.wordpress.com/2008/10/28/pie-in-the-sky/> [...]

134. [Fitts' Law and the Pie](#) Says:

[November 25, 2008 at 12:33 am](#)

[...] Jono at Mozilla Labs has a wonderful blog post on pie based menus, But pie menus are conspicuously absent from most popular UI libraries and toolkits. They're not in the Human Interface Guidelines of Mac, Windows, Gnome and KDE. [...]

135. [Anjoschu](#) Says:

[November 28, 2008 at 7:08 am](#)

Another tackle on pie menus on the web, with a simplistic prototype and some thoughts about proper use of pie menus in general:

<http://www.schuderer.net/experiments.shtml#piemenus>

136. [New MinimalComp: WheelNav | Verge](#) Says:

[December 10, 2008 at 2:47 am](#)

[...] <http://jonoscript.wordpress.com/2008/10/28/pie-in-the-sky/> [...]

137. [Fitts' Law and the Pie | L.A.R.S.](#) Says:

[December 18, 2008 at 6:39 pm](#)

[...] Jono at Mozilla Labs has a wonderful blog post on pie based menus, But pie menus are conspicuously absent from most popular UI libraries and toolkits. They're not in the Human Interface Guidelines of Mac, Windows, Gnome and KDE. [...]

138. [Bryan Seigneur](#) Says:

[December 26, 2008 at 9:03 pm](#)

This is fun and all if you've never used the menu before, it might help.

BUT, if your brain is used to storing stuff sequentially (up->down, or left->right (right->left if your mother language is Semitic)) this could slow you down in the long run.

Now, if you're used to this (you're Ferengi, or something), no problem. Even then, though, it would be better if the choices are all at 60 degree angles.

The more text is crammed what are usually non-texty containers, the more the widget in question will remain in the lab or ivory tower.

139. [MFA in Interaction Design - Wolf's Little Store](#) Says:

[February 11, 2009 at 9:08 pm](#)

[...] Fitt's Law Formula, door Jono. [...]

140. [RaiulBaztepo](#) Says:

[March 28, 2009 at 8:53 pm](#)

Hello!

Very Interesting post! Thank you for such interesting resource!

PS: Sorry for my bad english, I've just started to learn this language

See you!

Your, Raiul Baztepo

141. [Parnell Springmeyer](#) Says:

[August 26, 2009 at 4:54 pm](#)

Hello! I came across your Javascript implementation and wanted to adapt it for use in my own web application.

I've already tweaked it considerably and applied it for what I wanted, but, I do want your blessing and permission to use your code.

– Parnell

142. [Interesting reading... – The Blogs at HowStuffWorks](#) Says:

[January 11, 2010 at 5:22 pm](#)

[...] Pie In The Sky – “Anyway, the demo presented two different styles of pop-up menus and asked you to repeatedly select the same menu item. Which menu style made the task easier? Why do you think that was?” [...]

143. [Microsoft Courier – brauchbar als Notizbuchersatz?](#) | [caroco.de](#) Says:

[March 9, 2010 at 12:30 pm](#)

[...] leider noch nichts bekannt aber ich habe mich gefreut, einen alten Bekannten wiederzutreffen: Das Pie-Menu... Category: Allgemein, Interaction Tags: creative, inspiration, interaction paradigm, [...]

Leave a Reply

Enter your comment here...

Guest

Log In

Log In

Log In

Email (required)

(Not published)

Name (required)

Website

☐ Notify me of follow-up comments via email.

☐ Notify me of new posts via email.

Post Comment

Archived Entry

- **Post Date :**
- October 28, 2008 at 12:34 am
- **Category :**
- [Uncategorized](#)
- Tags: [Fitts' Law](#), [interactive demos](#), [pie menus](#), [ubiquity](#), [UI math](#)
- **Do More :**
- You can [leave a response](#), or [trackback](#) from your own site.

[Blog at WordPress.com.](#) — Theme: [Connections](#) by [www.vanillamist.com](#).