

igraph-help[\[Top\]](#) [\[All Lists\]](#) Search[Advanced](#)[\[Date Prev\]](#) [\[Date Next\]](#) [\[Thread Prev\]](#) [\[Thread Next\]](#) [\[Date Index\]](#) [\[Thread Index\]](#)**Re: [igraph] vertex.frame.width? vertex.frame.size?****From:** Gábor Csárdi**Subject:** Re: [igraph] vertex.frame.width? vertex.frame.size?**Date:** Fri, 15 Mar 2013 10:27:00 -0400

Hi Yannick,

indeed, this does not work any more, but the good news is that there is now an API to define new vertex shapes. Here is an example:

library(igraph)

```
mycircle <- function(coords, v=NULL, params) {
  vertex.color <- params("vertex", "color")
  if (length(vertex.color) != 1 && !is.null(v)) {
    vertex.color <- vertex.color[v]
  }
  vertex.size <- 1/200 * params("vertex", "size")
  if (length(vertex.size) != 1 && !is.null(v)) {
    vertex.size <- vertex.size[v]
  }
  vertex.frame.color <- params("vertex", "frame.color")
  if (length(vertex.frame.color) != 1 && !is.null(v)) {
    vertex.frame.color <- vertex.frame.color[v]
  }
  vertex.frame.width <- params("vertex", "frame.width")
  if (length(vertex.frame.width) != 1 && !is.null(v)) {
    vertex.frame.width <- vertex.frame.width[v]
  }

  mapply(coords[,1], coords[,2], vertex.color, vertex.frame.color,
          vertex.size, vertex.frame.width,
          FUN=function(x, y, bg, fg, size, lwd) {
            symbols(x=x, y=y, bg=bg, fg=fg, lwd=lwd,
                  circles=size, add=TRUE, inches=FALSE)
          })
}
```

```
add.vertex.shape("fcircle", clip=igraph.shape.noclip,
  plot=mycircle, parameters=list(vertex.frame.color=1,
    vertex.frame.width=1))

plot(graph.ring(10), vertex.shape="fcircle", vertex.frame.color=rainbow(10),
  vertex.frame.width=1:10/2)
```

The new API is not the best, and it is likely to change again in the next major version (0.7), so watch out!

Best,
Gabor

On Fri, Mar 15, 2013 at 7:21 AM, Yannick Rochat <[address@hidden](#)> wrote:

Dear Gabor,

I've kept using your great option "vertex.frame.width" until today (see message I'm replying to). But with R 2.15.3 & igraph 0.6.5-1 I've just updated, it doesn't work anymore :-)

For example, I cannot reproduce the example at the end of your post. I get the following error (in french)

Erreur dans .igraph.shapes[[shape[1]]]\$clip :
objet de type 'closure' non indiquable

Thanks in advance for your help !

Best,

Yannick

2009/9/10 Gábor Csárdi <[address@hidden](#)>

Yannick,

the logo was not actually made with igraph. :(But anyway, you can change the edge width by giving 'edge.width' to plot, or setting the 'width' edge attribute. You cannot currently change the width of the frame, at least not easily, you need to define a new vertex shape for this, slightly modifying the current 'circle' shape.

```
.igraph.shape.circle2 <- function(coords, el=NULL, v=NULL,
  mode=c("clip", "plot"),
    params, end=c("both", "from", "to")) {

  mode=match.arg(mode)
  end =match.arg(end)
```

```
#####
## clipping mode

if (mode=="clip") {
  if (length(coords)==0) { return (coords) }

  vertex.size <- 1/200 * params("vertex", "size")

  if (end=="from") {
    phi <- atan2(coords[,4] - coords[,2], coords[,3] - coords[,1])
    vsize.from <- if (length(vertex.size)==1) {
      vertex.size
    } else {
      vertex.size[ el[,1]+1 ]
    }
    res <- cbind(coords[,1] + vsize.from*cos(phi),
                  coords[,2] + vsize.from*sin(phi) )
  } else if (end=="to") {
    phi <- atan2(coords[,4] - coords[,2], coords[,3] - coords[,1])
    r <- sqrt( (coords[,3] - coords[,1])^2 + (coords[,4] - coords[,2])^2 )
    vsize.to <- if (length(vertex.size)==1) {
      vertex.size
    } else {
      vertex.size[ el[,2]+1 ]
    }
    res <- cbind(coords[,1] + (r-vsize.to)*cos(phi),
                  coords[,2] + (r-vsize.to)*sin(phi) )
  } else if (end=="both") {
    phi <- atan2(coords[,4] - coords[,2], coords[,3] - coords[,1])
    r <- sqrt( (coords[,3] - coords[,1])^2 + (coords[,4] - coords[,2])^2 )
    vsize.from <- if (length(vertex.size)==1) {
      vertex.size
    } else {
      vertex.size[ el[,1]+1 ]
    }
    vsize.to <- if (length(vertex.size)==1) {
      vertex.size
    } else {
      vertex.size[ el[,2]+1 ]
    }
    res <- cbind(coords[,1] + vsize.from*cos(phi),
                  coords[,2] + vsize.from*sin(phi),
                  coords[,1] + (r-vsize.to)*cos(phi),
                  coords[,2] + (r-vsize.to)*sin(phi) )
  }

  res
}
```

```
#####
## plotting mode

} else if (mode=="plot") {
  vertex.color    <- params("vertex", "color")
  if (length(vertex.color) != 1 && !is.null(v)) {
    vertex.color <- vertex.color[v+1]
  }
  vertex.frame.color <- params("vertex", "frame.color")
  if (length(vertex.frame.color) != 1 && !is.null(v)) {
    vertex.frame.color <- vertex.frame.color[v+1]
  }
  vertex.size      <- 1/200 * params("vertex", "size")
  if (length(vertex.size) != 1 && !is.null(v)) {
    vertex.size <- vertex.size[v+1]
  }
  vertex.size <- rep(vertex.size, length=nrow(coords))

  vertex.frame.width <- params("vertex", "frame.width")
  vertex.frame.width <- rep(vertex.frame.width, length=nrow(coords))

  vertex.color <- rep(vertex.color, length=nrow(coords))
  vertex.frame.color <- rep(vertex.frame.color, length=nrow(coords))
  vertex.size <- rep(vertex.size, length=nrow(coords))

  for (i in seq_len(nrow(coords))) {
    symbols(x=coords[i,1], y=coords[i,2], bg=vertex.color[i],
fg=vertex.frame.color[i],
           circles=vertex.size[i], add=TRUE, inches=FALSE,
lwd=vertex.frame.width[i])
  }
}

.igraph.shapes <- get( ".igraph.shapes", asNamespace("igraph"))
.igraph.shapes[["circle2"]] <- .igraph.shape.circle2
unlockBinding(".igraph.shapes", asNamespace("igraph"))
assign(".igraph.shapes", .igraph.shapes, envir=asNamespace("igraph"))

i.default.values <- get("i.default.values", asNamespace("igraph"))
i.default.values$vertex$frame.width=1
unlockBinding("i.default.values", asNamespace("igraph"))
assign("i.default.values", i.default.values, envir=asNamespace("igraph"))
```

and then you can do something like this:

```
plot(graph.ring(10), layout=layout.circle, vertex.shape="circle2",  
vertex.frame.width=5, edge.width=1:10)
```

We should have an API for adding new vertex shapes soon.....

Best,
Gabor

On Thu, Sep 10, 2009 at 2:09 PM, Yannick Rochat

<address@hidden> wrote:

> Dear Gábor and Tamás,

>

> I'm plotting some graphs for a poster and I would like to increase the width
> of the edges and the vertices' frames in order to have it equal everywhere.

> How can I do that? It seems possible to do it with igraph:

> <http://igraph.sourceforge.net/images/igraph2.png> if that was produced with
> igraph :)

>

> Thanks!

>

>

> Yannick Rochat - IMA - Université de Lausanne

> <http://www.unil.ch/unisciences/YannickRochat>

>

>

> igraph-help mailing list

> address@hidden

> <http://lists.nongnu.org/mailman/listinfo/igraph-help>

>

>

--

Gabor Csardi <address@hidden> UNIL DGM

igraph-help mailing list

address@hidden

<http://lists.nongnu.org/mailman/listinfo/igraph-help>

igraph-help mailing list

address@hidden

<https://lists.nongnu.org/mailman/listinfo/igraph-help>

--
Gabor Csardi <address@hidden> MTA KFKI RMKI

reply via email to

[Gábor Csárdi](#)

[\[Prev in Thread\]](#)

Current Thread

[\[Next in Thread\]](#)

- **[Re: \[igraph\] vertex.frame.width? vertex.frame.size?](#)**, *Yannick Rochat*, 2013/03/15
 - **[Re: \[igraph\] vertex.frame.width? vertex.frame.size?](#)**, *Gábor Csárdi* <=>

-
- Prev by Date: [\[igraph\] Help for transitivity vs clustering](#)
 - Next by Date: [Re: \[igraph\] Help for transitivity vs clustering](#)
 - Previous by thread: [Re: \[igraph\] vertex.frame.width? vertex.frame.size?](#)
 - Next by thread: [\[igraph\] Help for transitivity vs clustering](#)
 - Index(es):
 - [Date](#)
 - [Thread](#)