

Okular tile-based engine

Mailson D. Lira Menezes

About me

Mailson Menezes

Google Summer of Code 2012

Openbossa - INdT (Instituto Nokia de Tecnologia)

Recife - Brazil

blog.mailson.org

github.com/omailson

mailson@gmail.com

The problem

400% zoom limit

Bug 148527 - I want more zoom with okular

Status RESOLVED FIXED

Product	i okular	Reported	2007-08-05 09:38 UTC by rodz
Component	i general	Modified	2013-05-05 12:10 UTC (History)
Version	0.5.90	CC List	<input checked="" type="checkbox"/> Add me to CC list 15 users (edit)
Platform	unspecified Linux	See Also	i
Importance	NOR wishlist with 273 votes (vote)	Latest Commit	i
Target Milestone	---	Version Fixed In	i
Assigned To	Okular developers		

The problem

Entire page at once



Okular basics

Internals

Generator

PDF

Receive a pixmap request

Pixmaps can be stored in cache

Tile-based rendering

The Tile Manager

Separate the logic

Internal tiles structure

Okular talks to the Tile Manager

Start using tiles

Initially don't use tiles

- Requests are for the entire page

If the request is too big

- Create a tile manager

- Intercept the request

 - Change it to the visible region

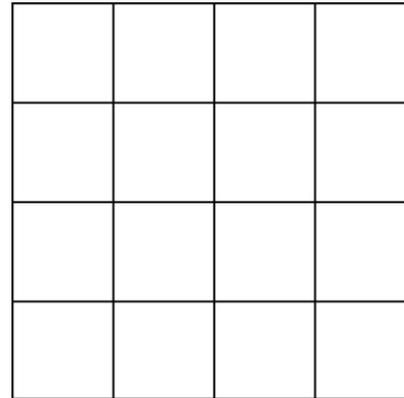
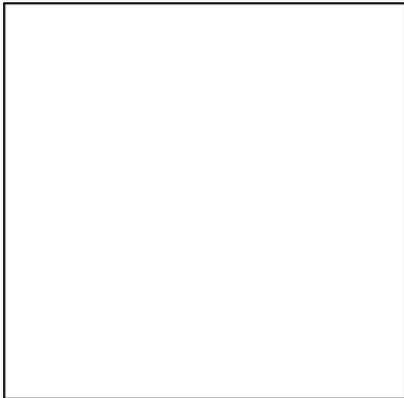
- Store the cached page on the tile manager

Tile representation

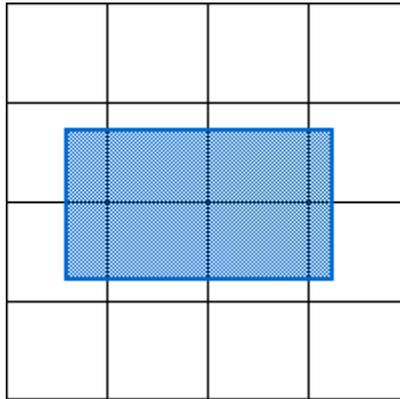
Page initially divided

Tile not in a fixed size

Tile associated with the same region



Start using tiles

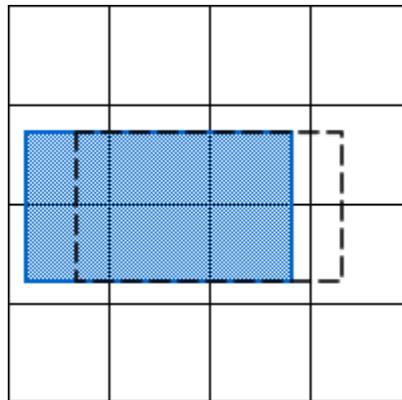


Requesting tiles

Check for tiles in the viewport

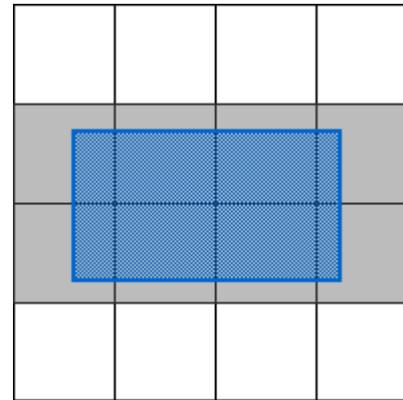
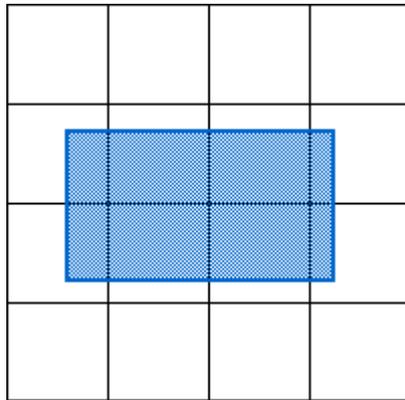
Does it have pixmaps?

Are they updated?



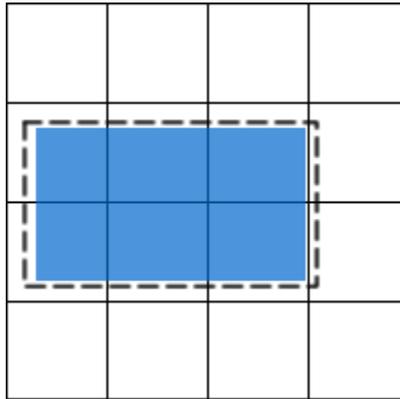
Requesting tiles

Create a tiled pixmap request
Tile align the visible region



Preload tiles

Expand the viewport



Render tiled request

Send request to the generator

Check if it's a tiled request

Render a cropped region

Small change in the code

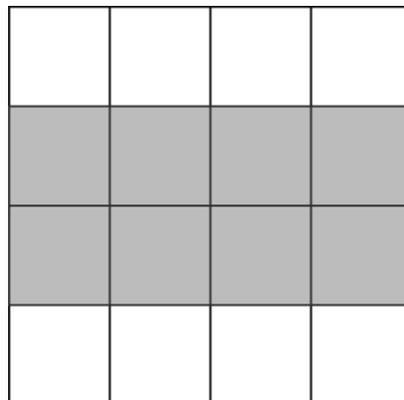
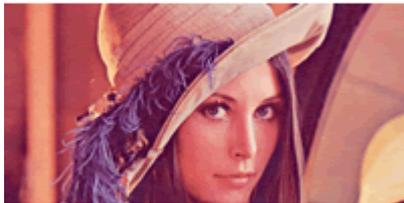
```
QImage img;
if (p)
{
+   if ( request->isTile() )
+   {
+       QRect rect = request->normalizedRect().geometry( request->width(), request->height() );
+       img = p->renderToImage( fakeDpiX, fakeDpiY, rect.x(), rect.y(), rect.width(), rect.height(), Poppler::Page::Rotate0 );
+   }
+   else
+   {
+       img = p->renderToImage(fakeDpiX, fakeDpiY, -1, -1, -1, -1, Poppler::Page::Rotate0 );
+   }
}
```

Store rendered pixmap

Let the tile manager handle it

The pixmap may represent more than one tile

It will crop the pixmap accordingly



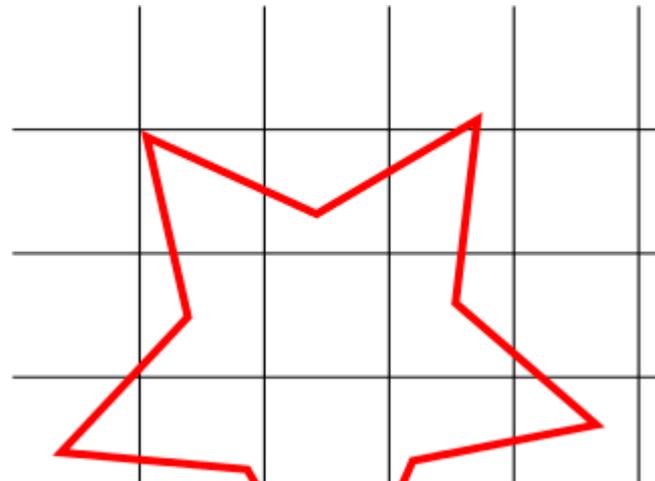
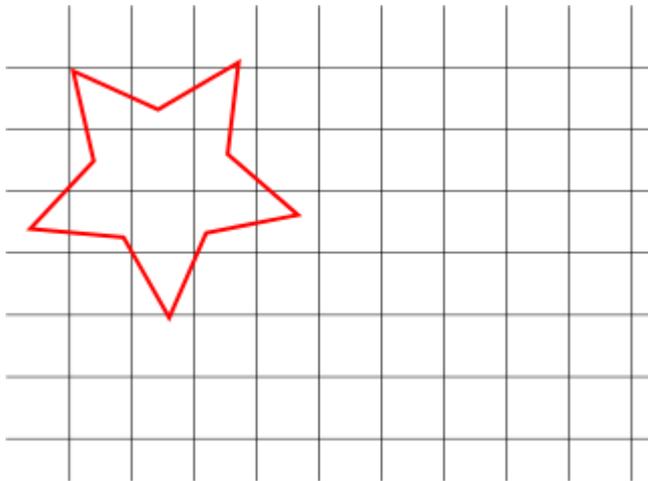
Zoom changes

Mark tiles as dirty

Keep cached pixmaps

Tiles still represent the same region

Cached pixmap can be rescaled

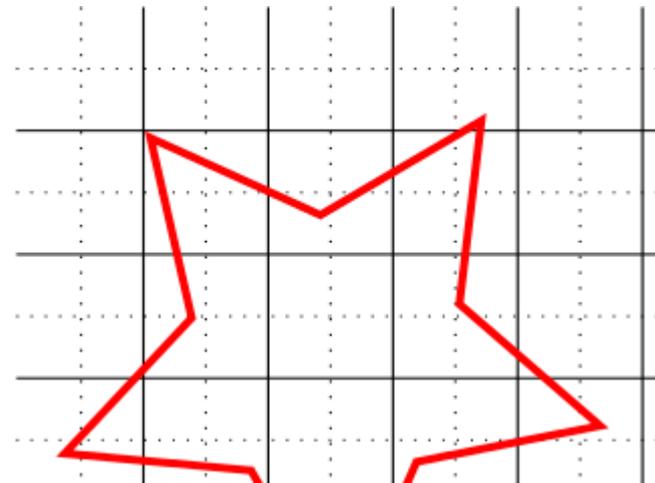
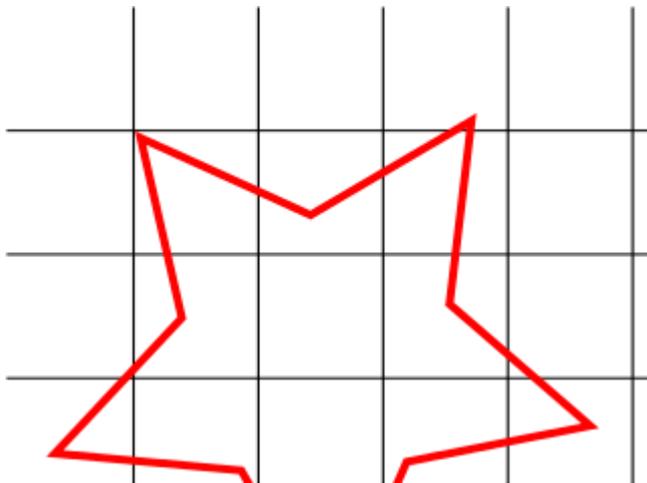


When the tile becomes big

It happens when we zoom even further

Split the tile

Only the requested region



When the tile becomes big

Tree structure

No cropping (yet)



Tile representation

Tree of tiles



...and when tiles are too small

Much easier

Requested pixmap is too small

Don't need to crop

Delete children tiles



Memory management

Okular already deletes old cached pages
Old tiles should be evicted as well

Problem: which one to delete first?

Memory management

Rank tiles

- By dirty state

- By distance from the viewport

Keep visible tiles

Page painting

Fetch the tiles of a region to paint
Get each piece and build the page
Tiles may be outdated
 Paint old tiles rescaled

Stop using tiles

When the page is not big enough

Remove the tiles manager

Request an entire page

What's next?

Cache pixmap in different zoom levels

Use the rendered time in the ranking algorithm

Parallel rendering

Export to an external lib :)

Thanks!

Mailson D. Lira Menezes

blog.mailson.org

github.com/omailson