This Way

ConTEXt magazine #10 MKIV
March 2005

Good looking shapes
Hans Hagen
PRAGMA ADE

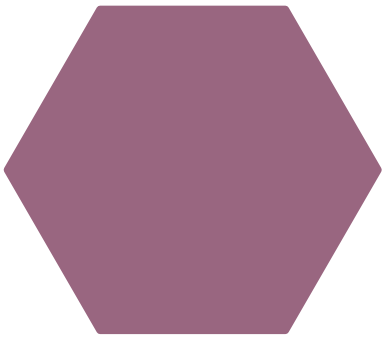
Just as it takes while to get an understanding what TEX is about, it takes a couple of listening loops to get a general picture about Tori Amos' Beekeeper. While browsing the rather nicely designed booklet I got puzzled —as usual when seeing such nice book(let)s— why everything looked okay except the text. High end design combined with rather low end typography. Don't get me wrong, apart from the typesetting it's a pretty good product! Tori being one of my favourite artists, you can imagine that I wrote quite some ConTEXt code listening to her music.

Now I will not argue that TEX (or ConTEXt) is the proper system for making CD covers, but since most of such a booklet is a matter of pasting graphics components together, I can imagine that one should ask someone to typeset the text snippets using a proper engine. Anyway, most buyers (fans) won't notice it, but anyone familiar with TEX will immediate get distracted by the strange intercharacter and interline spacing.

Typesetting in a fixed shape is non–trivial. First of all lines should break in a pleasing way. If possible, hyphenation should be avoided. The gaps between characters must not become to large and the last line should not be too short. Doing this in TEX is non trivial either, not so much because TEX cannot do such things, but because one needs to control several mechanisms at once. On the other hand, one should know what one's dealing with anyway.

Because the size of the shape is fixed, we can manipulate the number of lines and/or the line length and scale afterwards to the desired size. The font size is not fixed. This permits us to implement a semi–automated solution. The difference between the first version of the solution and current one is that we take into account an odd/even number of lines. Also, finding the best exit condition took some experiments. The final solution is not that complex and also shows a couple of tricks.

The shape we are dealing with looks as follows:



We will will later put such a shape behind the text for which we define an overlay:

```
\definecolor[BeeColorA][r=.4,g=.5,b=.6]
\definecolor[BeeColorB][r=.5,g=.6,b=.4]
\definecolor[BeeColorC][r=.6,g=.4,b=.5]

\definecolor[BeeColor] [BeeColorA]
```

```
\defineoverlay
  [beecell]
  [\uniqueMPgraphic{beecell}{offset=3mm,color=BeeColor}]

\startuniqueMPgraphic{beecell}{offset,color}
  fill
    for i = 1 upto 6 : (0,OverlayHeight/2)
      rotatedaround (center OverlayBox,i*60) --
    endfor cycle
    withpen pencircle scaled \MPvar{offset}
    withcolor \MPvar{color} ;
\stopuniqueMPgraphic
```

Normally one will not put a shape behind the text, but in our case it illustrates the idea. We use an offset in order to get a more pleasing look.

We will use the following two sample texts. The original linebreaks are visible in the source:

```
\startbuffer[parasol]
\title {PARASOL} when I come to
terms to terms with this when
I come to terms with this when I
come to terms to terms with this my
world will change for me I haven't moved
since the call came since the call came I
haven't moved I stare at the wall knowing on the
other side the storm that waits for me then the
Seated Woman with a Parasol may be the only one you
can't Betray if I'm the Seated Women with a Parasol I will
be safe in my frame I have no need for a sea view for a sea
view I have no need I have my little pleasures this wall
being one of these when I come to terms to terms
with this when I come to terms with this when I
come to terms with this whip lash of Silk on
wool embroidery then the Seated Woman
with a Parasol may be the only one you
can't betray if I'm the Seated Woman
with a Parasol I will be safe in my
frame I will be safe in my frame
in your House in your frame
\stopbuffer

\startbuffer[beekeeper]
\title {THE BEEKEEPER} Flaxen hair
blowing in the breeze It is time
for the geese to head south I have
```

```
come with my mustard seed I cannot
accept that she will be taken from me
``Do you know who I am'' she said ``I'm the
one who taps you on the shoulder when it's
your time Don't be afraid I promise that she
will awake Tomorrow Somewhere Tomorrow
Somewhere'' --- wrap yourself around the Tree of
Life and the Dance of the Infinity of the Hive --- take
this message to Michael I will comb myself into chains In
between the tap dance clan and your ballerina gang I have
come for the Beekeeper I know you want my You want
my Queen --- Anything but this Can you use me instead?
In your gown with your breathing mask Plugged into
a heart machine As if you ever needed one I must
see the Beekeeper I must see if she'll keep her
alive Call Engine 49 I have come with my
mustard seed Maybe I'm passing you by
On my way On my way I'm just passing
you by But don't be confused
One day I'll be coming for you \unknown\space
I must see the Beekeeper
I must see the Beekeeper
\stopbuffer
```

We will call these buffers indirectly (using setups is a convenient way to collect commands and definitions).

```
\startsetups [beetext]
  \getbuffer[parasol]
\stopsetups
```

Now comes the dirty code. We assume that you know a bit of ConTeXt. First of all we choose a font, in our case a Termes for the running text. We will use Hermann Zapf optimization, which is way more acceptable that intercharacter spacing and gives quite good results here.

```
\definefontfeature[hzdefault][default][hz=quality]
\definefont[BeeFont][file:texgyre-termes*hzdefault]
```

The core of the code is a loop wherein we try to figure out what the best width is. In principle this method can be used for similar shapes. Beforehand we define a few variables.

```
\cldcontext{math.cosd(60)}
\cldcontext{math.sind(60)}

\newdimen\BeeEdge
\newdimen\BeeLine
```

```
\newdimen\BeeSize

\newbox  \BeeBox

\def\BeeLines{17}   % choose optimum odd/even
\def\BeeStart{2cm}  % set automatically
\def\BeeStep {.5mm} % accurate enough
```

0.5 0.86602540378444

The loop starts with a rather small width and with increasing steps tries to find the solution where the number of used lines equals the asked number of lines. We could have used low level TEX primitives, but using a few ConTEXt wrappers makes more sense because that way struts and alike are set as well. In the end we stretch the interline spacing to match the height of the cell.

```
\startsetups beeloop

\def\title##1%
  {{\ss\bf\kerncharacters[0.25]##1}%
    \hskip.5em plus .5em minus .25em\relax
    \ignorespaces}

\setbox\scratchbox=\hbox{\setups[beetext]}

\edef\BeeStart
  {\the\dimexpr.5\wd\scratchbox/\BeeLines\relax}

\def\BeeMax
  {10000}

\def\BeeShapeA
  {\scratchdimen\numexpr\recurselevel-1\relax
     \dimexpr\BeeEdge/\BeeLast\relax
   \appendetoks
     \the\dimexpr\BeeEdge- \scratchdimen\relax\space
     \the\dimexpr\hsize  +2\scratchdimen\relax\space
   \to\scratchtoks}

\def\BeeShapeB
  {\appendetoks
     \zeropoint\space
     \the\dimexpr\hsize+2\BeeEdge\relax\space
   \to\scratchtoks}

\doloop
  {\bgroup
```

```
\forgetall
\dontcomplain
\edef\BeeLast
  {\the\numexpr(\BeeLines\ifodd\BeeLines-1\fi)/2\relax}%
\hsize\dimexpr\BeeStart+\recurselevel\dimexpr\BeeStep\relax\relax
\BeeEdge=\cldcontext{math.cosd(60)}\hsize
\BeeSize=\cldcontext{math.sind(60)}\hsize
\BeeLine=\dimexpr2\BeeSize/\numexpr2*\BeeLast+1\relax\relax
\setupinterlinespace[line=\BeeLine,stretch=.5]%
\setuptolerance[verytolerant]%
\setupalign[hz]%
\parfillskip\zeropoint
\scratchtoks\emptytoks
\ifodd\BeeLines
  \dostepwiserecurse{1}{\BeeLast}{+1}{\BeeShapeA}%
                                     \BeeShapeB
  \dostepwiserecurse{\BeeLast}{1}{-1}{\BeeShapeA}%
  \rightskip\zeropoint
\else
  % we want to stay inside the shape, so we need
  % to compensate the right side
  \advance\hsize +\dimexpr\BeeEdge/\BeeLast\relax
  \dostepwiserecurse{1}{\BeeLast}{+1}{\BeeShapeA}%
  \dostepwiserecurse{\BeeLast}{1}{-1}{\BeeShapeA}%
  \advance\hsize -\dimexpr\BeeEdge/\BeeLast\relax
  \rightskip\dimexpr\BeeEdge/\BeeLast\relax
\fi
\setbox\scratchbox\vbox \bgroup
  % we set it like this in case grid is turned on
  \baselineskip=1\baselineskip plus 20pt minus 20pt
  \parshape\numexpr\BeeLines\relax\the\scratchtoks
  \begstrut
  \ignorespaces\setups[beetext]\removeunwantedspaces
  \endstrut
  \endgraf
  \xdef\BeeTotal{\number\prevgraf}%
  \xdef\BeeRate {\number\badness }%
\egroup
\writestatus
  {beestate}
  {     run: \recurselevel\space
     target: \BeeLines    \space
      lines: \BeeTotal    \space
    badness: \BeeRate}%
\CheckBeeLines % sets 'done'
\ifdone
```

```
      \vbox to 2\BeeSize
        {\unvbox\ifvoid\BeeBox\scratchbox\else\BeeBox\fi}%
      \egroup
      \exitloop
    \else
      \egroup
    \fi}

\stopsetups
```

The end criterium is determined by:

```
\def\CheckBeeLines
  {\ifnum\BeeTotal>\BeeLines\relax
     \donefalse
   \else
     \donetrue
   \fi}
```

This solution is rather safe and, at the cost of the ugly saving of the number of lines as registered in `\prevgraf`, works better than measuring the height of the box.

We could build the loop out of more isolated pieces of code like this but the reason why we do it for the checker is that we now can redefine it. At the cost of a few more tests, the following checker is better, because it goes on for a while and keeps looking for better solutions. If you have no idea what badness is, just skip the following code snippet.

```
\def\CheckBeeLines
  {\ifnum\BeeTotal>\BeeLines\relax
     \donefalse
   \else\ifnum\BeeTotal=\BeeLines\relax
     \ifnum\BeeRate=\zerocount
       \global\setbox\BeeBox=\box\scratchbox
       \donetrue
     \else\ifnum\BeeRate<\BeeMax\relax
       \global\let\BeeMax\BeeRate
       \global\setbox\BeeBox=\box\scratchbox
       \donefalse
     \else
       \donefalse
     \fi\fi
   \else
     \donetrue
   \fi\fi}
```

Well, this is not the kind of code you want a designer to enter, but providing it as feature in a desk top publishing application is also non–trivial because each case

differs and turning many knobs to get things done is not easy either, so basically it comes down to manual work (neglectable to the total amount of work involved in getting such a musical product done). Of course one can ask someone to typeset the text in TEX and provide it as image, but that would make coordination the production more complex.

The criterium (here .5mm) can be made smaller when you encounter problems. If we set it to 1mm, we get one case where the amount of lines jumps 2 and the loop is exit unexpected. Of course one can catch such cases but it does not make much sense in such a one–shot macro.

The previous setup is applied as follows:

```
\startsetups beeloner
  \framed
    [offset=overlay,
     frame=off,
     background=beecell,
     foregroundstyle=\BeeFont]
    {\setups[beeloop]}
\stopsetups
```

We will now put several variants alongside. For this we use a layer:

```
\startsetups beesample

\definelayer
  [beekeeper]
  [width=13cm,
   height=9cm]

\setlayer
  [beekeeper]
  [preset=lefttop]
  {\scale[width=5cm]{\def\BeeLines{16}\setups[beeloner]}}

\setlayer
  [beekeeper]
  [preset=leftbottom]
  {\scale[width=5cm]{\def\BeeLines{17}\setups[beeloner]}}

\setlayer
  [beekeeper]
  [preset=righttop]
  {\scale[width=5cm]{\def\BeeLines{18}\setups[beeloner]}}

\setlayer
  [beekeeper]
```

```
  [preset=rightbottom]
  {\scale[width=5cm]{\def\BeeLines{19}\setups[beeloner]}}

\setlayer
  [beekeeper]
  [preset=middle]
  {\scale[width=5cm]{\def\BeeLines{20}\setups[beeloner]}}

\tightlayer[beekeeper]

\stopsetups
```

The first samples, shown in figure 2, will be typeset using:

```
\startsetups [beetext]
  \getbuffer[parasol]
\stopsetups

\definecolor[BeeColor][BeeColorA] \setups[beesample]
```

The second example, shown in figure 3, is done in a similar way. We redefine the `beetext` setup.

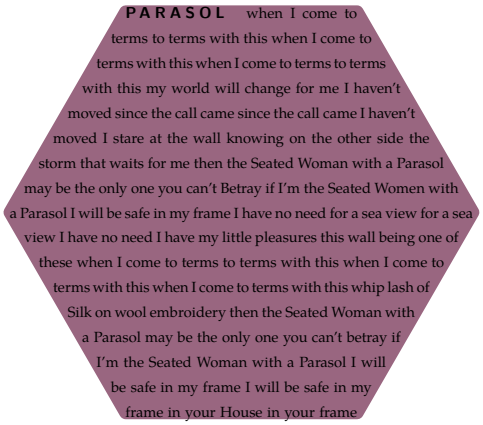
```
\startsetups [beetext]
  \getbuffer[beekeeper]
\stopsetups

\definecolor[BeeColor][BeeColorB] \setups[beesample]
```

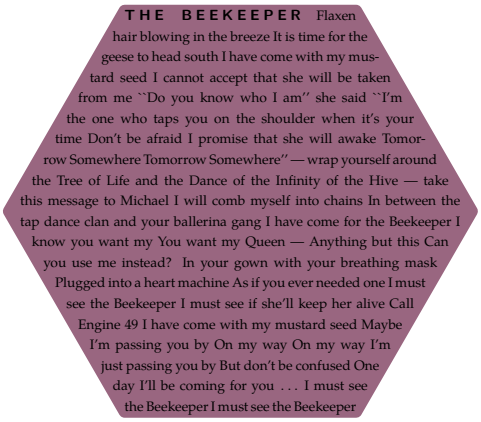
You can zoom in on cells using your viewer. An enlarged example is shown in figure 1.

```
\definecolor[BeeColor][BeeColorC]%
\startcombination
  {\scale
     [width=.475\textwidth]
     {\startsetups[beetext]\getbuffer[parasol]\stopsetups
      \def\BeeLines{17}\setups[beeloner]}}
  {Parasol}
  {\scale
     [width=.475\textwidth]
     {\startsetups[beetext]\getbuffer[beekeeper]\stopsetups
      \def\BeeLines{20}\setups[beeloner]}}
  {The Beekeeper}
\stopcombination
```

Choosing the best alternative is a matter of taste. If you ever get a change to see the CD (a good buy anyway) you will note the difference. It is possible to improve the spacing at the top and bottom but we leave this as an exercise.



Parasol



The Beekeeper

Figure 1 An few enlarged examples.

Figure 2 Parasol

Figure 3 The Beekeeper

The downside of this exercise was that in the process my laptop suddenly made some funny noises and made me end up with a cracked CD. So in the end the message may be not to bother too much about badly typeset paragraphs in CD booklets.

PARASOL when I come to
terms to terms with this when I come to
terms with this when I come to terms to terms
with this my world will change for me I haven't
moved since the call came since the call came I haven't
moved I stare at the wall knowing on the other side the
storm that waits for me then the Seated Woman with a Parasol
may be the only one you can't Betray if I'm the Seated Women with
a Parasol I will be safe in my frame I have no need for a sea view for a sea
view I have no need I have my little pleasures this wall being one of
these when I come to terms to terms with this when I come to
terms with this when I come to terms with this whip lash of
Silk on wool embroidery then the Seated Woman with
a Parasol may be the only one you can't betray if
I'm the Seated Woman with a Parasol I will
be safe in my frame I will be safe in my
frame in your House in your frame

THE BEEKEEPER Flaxen
hair blowing in the breeze It is time for the
geese to head south I have come with my mus-
tard seed I cannot accept that she will be taken
from me ``Do you know who I am'' she said ``I'm
the one who taps you on the shoulder when it's your
time Don't be afraid I promise that she will awake Tomor-
row Somewhere Tomorrow Somewhere'' — wrap yourself around
the Tree of Life and the Dance of the Infinity of the Hive — take
this message to Michael I will comb myself into chains In between the
tap dance clan and your ballerina gang I have come for the Beekeeper I
know you want my You want my Queen — Anything but this Can
you use me instead? In your gown with your breathing mask
Plugged into a heart machine As if you ever needed one I must
see the Beekeeper I must see if she'll keep her alive Call
Engine 49 I have come with my mustard seed Maybe
I'm passing you by On my way On my way I'm
just passing you by But don't be confused One
day I'll be coming for you . . . I must see
the Beekeeper I must see the Beekeeper

source code of this document

```
% language=uk

% author    : Hans Hagen
% copyright : PRAGMA ADE & ConTeXt Development Team
% license   : Creative Commons Attribution ShareAlike 4.0 International
% reference : pragma-ade.nl | contextgarden.net | texlive (related) distributions
% origin    : the ConTeXt distribution
%
% comment   : Because this manual is distributed with TeX distributions it comes with a rather
%             liberal license. We try to adapt these documents to upgrades in the (sub)systems
%             that they describe. Using parts of the content otherwise can therefore conflict
%             with existing functionality and we cannot be held responsible for that. Many of
%             the manuals contain characteristic graphics and personal notes or examples that
%             make no sense when used out-of-context.

\usemodule[mag-01,abr-02]

\startbuffer[abstract]
    The content of tenth magazine was written while listening to Tori Amos'
    latest album, The Beekeeper. In the (nice) booklet the text flows in shapes
    and here I will demonstrate that \TEX\ can do something similar. It's also a
    nice example of applying \HZ\ optimization.
\stopbuffer

\startdocument
  [title={Good looking shapes},
   author=Hans Hagen,
   affiliation=PRAGMA ADE,
   date=March 2005,
   number=10 \MKIV]

Just as it takes while to get an understanding what \TEX\ is about, it takes a
couple of listening loops to get a general picture about Tori Amos' Beekeeper.
While browsing the rather nicely designed booklet I got puzzled |<|as usual when
seeing such nice book(let)s|>| why everything looked okay except the text. High
end design combined with rather low end typography. Don't get me wrong, apart
from the typesetting it's a pretty good product! Tori being one of my favourite
artists, you can imagine that I wrote quite some \CONTEXT\ code listening to her
music.

Now I will not argue that \TEX\ (or \CONTEXT) is the proper system for making
\CD\ covers, but since most of such a booklet is a matter of pasting graphics
components together, I can imagine that one should ask someone to typeset the
text snippets using a proper engine. Anyway, most buyers (fans) won't notice it,
but anyone familiar with \TEX\ will immediate get distracted by the strange
intercharacter and interline spacing.

Typesetting in a fixed shape is non||trivial. First of all lines should break in
a pleasing way. If possible, hyphenation should be avoided. The gaps between
characters must not become to large and the last line should not be too short.
Doing this in \TEX\ is non trivial either, not so much because \TEX\ cannot do
such things, but because one needs to control several mechanisms at once. On the
other hand, one should know what one's dealing with anyway.

Because the size of the shape is fixed, we can manipulate the number of lines
and/or the line length and scale afterwards to the desired size. The font size is
not fixed. This permits us to implement a semi||automated solution. The
difference between the first version of the solution and current one is that we
take into account an odd|/|even number of lines. Also, finding the best exit
condition took some experiments. The final solution is not that complex and also
```

source code of this document

shows a couple of tricks.

```
\startbuffer
\definecolor[BeeColorA][r=.4,g=.5,b=.6]
\definecolor[BeeColorB][r=.5,g=.6,b=.4]
\definecolor[BeeColorC][r=.6,g=.4,b=.5]

\definecolor[BeeColor] [BeeColorA]

\defineoverlay
  [beecell]
  [\uniqueMPgraphic{beecell}{offset=3mm,color=BeeColor}]

\startuniqueMPgraphic{beecell}{offset,color}
  fill
    for i = 1 upto 6 : (0,OverlayHeight/2)
      rotatedaround (center OverlayBox,i*60) --
    endfor cycle
    withpen pencircle scaled \MPvar{offset}
    withcolor \MPvar{color} ;
\stopuniqueMPgraphic
\stopbuffer

\getbuffer
```

The shape we are dealing with looks as follows:

```
\startlinecorrection
\startMPcode
  fill
    for i = 1 upto 6 : (5cm,0)
      rotatedaround(origin,i*60) --
    endfor cycle
    withpen pencircle scaled 2mm
    withcolor \MPcolor{BeeColorC} ;
  currentpicture := currentpicture xsized(5cm) ;
\stopMPcode
\stoplinecorrection
```

We will will later put such a shape behind the text for which we define an overlay:

```
\typebuffer
```

Normally one will not put a shape behind the text, but in our case it illustrates the idea. We use an offset in order to get a more pleasing look.

We will use the following two sample texts. The original linebreaks are visible in the source:

```
\startbuffer
\startbuffer[parasol]
\title {PARASOL} when I come to
terms to terms with this when
I come to terms with this when I
come to terms to terms with this my
world will change for me I haven't moved
since the call came since the call came I
haven't moved I stare at the wall knowing on the
other side the storm that waits for me then the
Seated Woman with a Parasol may be the only one you
```

source code of this document

```
can't Betray if I'm the Seated Women with a Parasol I will
be safe in my frame I have no need for a sea view for a sea
view I have no need I have my little pleasures this wall
being one of these when I come to terms to terms
with this when I come to terms with this when I
come to terms with this whip lash of Silk on
wool embroidery then the Seated Woman
with a Parasol may be the only one you
can't betray if I'm the Seated Woman
with a Parasol I will be safe in my
frame I will be safe in my frame
in your House in your frame
\stopbuffer

\startbuffer[beekeeper]
\title {THE BEEKEEPER} Flaxen hair
blowing in the breeze It is time
for the geese to head south I have
come with my mustard seed I cannot
accept that she will be taken from me
``Do you know who I am'' she said ``I'm the
one who taps you on the shoulder when it's
your time Don't be afraid I promise that she
will awake Tomorrow Somewhere Tomorrow
Somewhere'' --- wrap yourself around the Tree of
Life and the Dance of the Infinity of the Hive --- take
this message to Michael I will comb myself into chains In
between the tap dance clan and your ballerina gang I have
come for the Beekeeper I know you want my You want
my Queen --- Anything but this Can you use me instead?
In your gown with your breathing mask Plugged into
a heart machine As if you ever needed one I must
see the Beekeeper I must see if she'll keep her
alive Call Engine 49 I have come with my
mustard seed Maybe I'm passing you by
On my way On my way I'm just passing
you by But don't be confused
One day I'll be coming for you \unknown\space
I must see the Beekeeper
I must see the Beekeeper
\stopbuffer
\stopbuffer

\typebuffer \getbuffer

We will call these buffers indirectly (using setups is a convenient way to
collect commands and definitions).

\startbuffer
\startsetups [beetext]
  \getbuffer[parasol]
\stopsetups
\stopbuffer

\typebuffer \getbuffer

Now comes the dirty code. We assume that you know a bit of \CONTEXT. First of all
we choose a font, in our case a Termes for the running text. We will use
Hermann Zapf optimization, which is way more acceptable that intercharacter
```

source code of this document

spacing and gives quite good results here.

```
\startbuffer
\definefontfeature[hzdefault][default][hz=quality]
\definefont[BeeFont][file:texgyre-termes*hzdefault]
\stopbuffer

\typebuffer \getbuffer
```

The core of the code is a loop wherein we try to figure out what the best width is. In principle this method can be used for similar shapes. Beforehand we define a few variables.

```
\startbuffer
\cldcontext{math.cosd(60)}
\cldcontext{math.sind(60)}

\newdimen\BeeEdge
\newdimen\BeeLine
\newdimen\BeeSize

\newbox  \BeeBox

\def\BeeLines{17}   % choose optimum odd/even
\def\BeeStart{2cm}  % set automatically
\def\BeeStep {.5mm} % accurate enough
\stopbuffer

\typebuffer \getbuffer
```

The loop starts with a rather small width and with increasing steps tries to find the solution where the number of used lines equals the asked number of lines. We could have used low level \TEX\ primitives, but using a few \CONTEXT\ wrappers makes more sense because that way struts and alike are set as well. In the end we stretch the interline spacing to match the height of the cell.

```
\startbuffer

\startsetups beeloop

\def\title##1%
  {{\ss\bf\kerncharacters[0.25]##1}%
    \hskip.5em plus .5em minus .25em\relax
    \ignorespaces}

\setbox\scratchbox=\hbox{\setups[beetext]}

\edef\BeeStart
  {\the\dimexpr.5\wd\scratchbox/\BeeLines\relax}

\def\BeeMax
  {10000}

\def\BeeShapeA
  {\scratchdimen\numexpr\recurselevel-1\relax
     \dimexpr\BeeEdge/\BeeLast\relax
   \appendetoks
     \the\dimexpr\BeeEdge- \scratchdimen\relax\space
     \the\dimexpr\hsize  +2\scratchdimen\relax\space
   \to\scratchtoks}

\def\BeeShapeB
```

source code of this document

```
  {\appendetoks
     \zeropoint\space
     \the\dimexpr\hsize+2\BeeEdge\relax\space
   \to\scratchtoks}

\doloop
  {\bgroup
   \forgetall
   \dontcomplain
   \edef\BeeLast
     {\the\numexpr(\BeeLines\ifodd\BeeLines-1\fi)/2\relax}%
   \hsize\dimexpr\BeeStart+\recurselevel\dimexpr\BeeStep\relax\relax
   \BeeEdge=\cldcontext{math.cosd(60)}\hsize
   \BeeSize=\cldcontext{math.sind(60)}\hsize
   \BeeLine=\dimexpr2\BeeSize/\numexpr2*\BeeLast+1\relax\relax
   \setupinterlinespace[line=\BeeLine,stretch=.5]%
   \setuptolerance[verytolerant]%
   \setupalign[hz]%
   \parfillskip\zeropoint
   \scratchtoks\emptytoks
   \ifodd\BeeLines
     \dostepwiserecurse{1}{\BeeLast}{+1}{\BeeShapeA}%
                                        \BeeShapeB
     \dostepwiserecurse{\BeeLast}{1}{-1}{\BeeShapeA}%
     \rightskip\zeropoint
   \else
     % we want to stay inside the shape, so we need
     % to compensate the right side
     \advance\hsize +\dimexpr\BeeEdge/\BeeLast\relax
     \dostepwiserecurse{1}{\BeeLast}{+1}{\BeeShapeA}%
     \dostepwiserecurse{\BeeLast}{1}{-1}{\BeeShapeA}%
     \advance\hsize -\dimexpr\BeeEdge/\BeeLast\relax
     \rightskip\dimexpr\BeeEdge/\BeeLast\relax
   \fi
   \setbox\scratchbox\vbox \bgroup
     % we set it like this in case grid is turned on
     \baselineskip=1\baselineskip plus 20pt minus 20pt
     \parshape\numexpr\BeeLines\relax\the\scratchtoks
     \begstrut
     \ignorespaces\setups[beetext]\removeunwantedspaces
     \endstrut
     \endgraf
     \xdef\BeeTotal{\number\prevgraf}%
     \xdef\BeeRate {\number\badness }%
   \egroup
   \writestatus
     {beestate}
     {     run: \recurselevel\space
        target: \BeeLines    \space
         lines: \BeeTotal    \space
       badness: \BeeRate}%
   \CheckBeeLines % sets 'done'
   \ifdone
     \vbox to 2\BeeSize
       {\unvbox\ifvoid\BeeBox\scratchbox\else\BeeBox\fi}%
     \egroup
     \exitloop
   \else
```

source code of this document

```
    \egroup
  \fi}

\stopsetups
\stopbuffer

\getbuffer \typebuffer
```

The end criterium is determined by:

```
\startbuffer
\def\CheckBeeLines
  {\ifnum\BeeTotal>\BeeLines\relax
     \donefalse
   \else
     \donetrue
   \fi}
\stopbuffer

\getbuffer \typebuffer
```

This solution is rather safe and, at the cost of the ugly saving of the number of lines as registered in \type {\prevgraf}, works better than measuring the height of the box.

We could build the loop out of more isolated pieces of code like this but the reason why we do it for the checker is that we now can redefine it. At the cost of a few more tests, the following checker is better, because it goes on for a while and keeps looking for better solutions. If you have no idea what badness is, just skip the following code snippet.

```
\startbuffer
\def\CheckBeeLines
  {\ifnum\BeeTotal>\BeeLines\relax
     \donefalse
   \else\ifnum\BeeTotal=\BeeLines\relax
     \ifnum\BeeRate=\zerocount
       \global\setbox\BeeBox=\box\scratchbox
       \donetrue
     \else\ifnum\BeeRate<\BeeMax\relax
       \global\let\BeeMax\BeeRate
       \global\setbox\BeeBox=\box\scratchbox
       \donefalse
     \else
       \donefalse
     \fi\fi
   \else
     \donetrue
   \fi\fi}
\stopbuffer

\getbuffer \typebuffer
```

Well, this is not the kind of code you want a designer to enter, but providing it as feature in a desk top publishing application is also non||trivial because each case differs and turning many knobs to get things done is not easy either, so basically it comes down to manual work (neglectable to the total amount of work involved in getting such a musical product done). Of course one can ask someone to typeset the text in \TEX\ and provide it as image, but that would make coordination the production more complex.

source code of this document

```
The criterium (here \BeeStep) can be made smaller when you encounter problems. If
we set it to 1mm, we get one case where the amount of lines jumps~2 and the loop
is exit unexpected. Of course one can catch such cases but it does not make much
sense in such a one||shot macro.

The previous setup is applied as follows:

\startbuffer
\startsetups beeloner
  \framed
    [offset=overlay,
     frame=off,
     background=beecell,
     foregroundstyle=\BeeFont]
    {\setups[beeloop]}
\stopsetups
\stopbuffer

\getbuffer \typebuffer

We will now put several variants alongside. For this we use a layer:

\startbuffer
\startsetups beesample

\definelayer
  [beekeeper]
  [width=13cm,
   height=9cm]

\setlayer
  [beekeeper]
  [preset=lefttop]
  {\scale[width=5cm]{\def\BeeLines{16}\setups[beeloner]}}

\setlayer
  [beekeeper]
  [preset=leftbottom]
  {\scale[width=5cm]{\def\BeeLines{17}\setups[beeloner]}}

\setlayer
  [beekeeper]
  [preset=righttop]
  {\scale[width=5cm]{\def\BeeLines{18}\setups[beeloner]}}

\setlayer
  [beekeeper]
  [preset=rightbottom]
  {\scale[width=5cm]{\def\BeeLines{19}\setups[beeloner]}}

\setlayer
  [beekeeper]
  [preset=middle]
  {\scale[width=5cm]{\def\BeeLines{20}\setups[beeloner]}}

\tightlayer[beekeeper]

\stopsetups
\stopbuffer

\getbuffer \typebuffer
```

source code of this document

```
\startbuffer[a]
\startsetups [beetext]
  \getbuffer[parasol]
\stopsetups

\definecolor[BeeColor][BeeColorA] \setups[beesample]
\stopbuffer

\startbuffer[b]
\startsetups [beetext]
  \getbuffer[beekeeper]
\stopsetups

\definecolor[BeeColor][BeeColorB] \setups[beesample]
\stopbuffer

\startpostponing

\placefigure
  [here]
  [fig:parasol]
  {Parasol}
  {\getbuffer[a]}

\placefigure
  [here]
  [fig:beekeeper]
  {The Beekeeper}
  {\getbuffer[b]}

\page

\stoppostponing

The first samples, shown in \in {figure} [fig:parasol], will be typeset using:

\typebuffer[a]

The second example, shown in \in {figure} [fig:beekeeper], is done in a similar
way. We redefine the \type {beetext} setup.

\typebuffer[b]

You can zoom in on cells using your viewer. An enlarged example is shown in \in
{figure} [fig:big].

\startbuffer
\definecolor[BeeColor][BeeColorC]%
\startcombination
  {\scale
     [width=.475\textwidth]
     {\startsetups[beetext]\getbuffer[parasol]\stopsetups
      \def\BeeLines{17}\setups[beeloner]}}
  {Parasol}
  {\scale
     [width=.475\textwidth]
     {\startsetups[beetext]\getbuffer[beekeeper]\stopsetups
      \def\BeeLines{20}\setups[beeloner]}}
  {The Beekeeper}
\stopcombination
\stopbuffer
```

source code of this document

```
\typebuffer

Choosing the best alternative is a matter of taste. If you ever get a change to
see the \CD\ (a good buy anyway) you will note the difference. It is possible to
improve the spacing at the top and bottom but we leave this as an exercise.

\placefigure
  [here]
  [fig:big]
  {An few enlarged examples.}
  {\getbuffer}

The downside of this exercise was that in the process my laptop suddenly made
some funny noises and made me end up with a cracked \CD. So in the end the
message may be not to bother too much about badly typeset paragraphs in \CD\
booklets.

\vbox to \vsize \bgroup

  \vfil

  \hbox to \hsize \bgroup \hss
    \scale
       [height=.45\textheight]
       {\startsetups[beetext]\getbuffer[parasol]\stopsetups
       \defineoverlay[beecell][]\def\BeeLines{17}\setups[beeloner]}%
  \hss \egroup

  \vfil \vfil

  \hbox to \hsize \bgroup \hss
    \scale
       [height=.45\textheight]
       {\startsetups[beetext]\getbuffer[beekeeper]\stopsetups
        \defineoverlay[beecell][]\def\BeeLines{20}\setups[beeloner]}%
  \hss \egroup

  \vfil

\egroup

\stopdocument
```

source code of this document