METAFUN

simple fonts

context 2020 meeting

Metafonts

- Because MetaPost is based on METAFONT it make sense to use of for making fonts.
- Making a font is an art in itself, something that is actually proven by many bad looking fonts, but we have plenty of choice nowadays.
- We tend to use free fonts and often being made by volunteers we can hardly have any demands.
- So, instead of complaining (which is not nice anyway) we can try to (at least temporary) come up with a solution ourselves.
- We're actually talking about missing glyphs here and MetaPost can be of help.
- Also keep in mind that we always had this option or variants of it in ConTEXt, it's just that we can make nicer interfaces now.
- So, don't expect something spectacular.

What is is not

Years ago mechanisms were added to MkIV to come up with more fancy shapes in for instance math. Actually Alan needed it and I wanted a root symbol to look like school times.

```
\useMPlibrary[mat]

\setupmathradical[color=darkgray,alternative=mp]

% \definemathradical [sqrt] [mp=minifun::math:radical:default]
```

So:

```
\scale[height=2cm]{$ \sqrt {a+b+c+d} $}
```

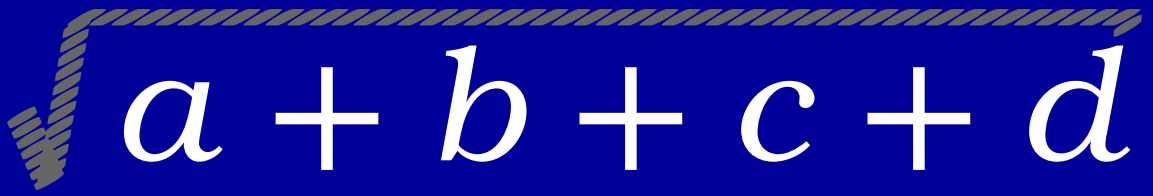
Gives:

$$\sqrt{a+b+c+d}$$

And with:

```
\startuniqueMPgraphic{minifun::math:radical:default}
draw
    math_radical_simple(OverlayWidth,OverlayHeight,OverlayDepth,OverlayOffset)
    withpen pencircle
        xscaled (2OverlayLineWidth)
        yscaled (1OverlayLineWidth/4)
        rotated 30
    dashed evenly
    withcolor OverlayLineColor ;
\stopuniqueMPgraphic
```

We get



Also think of stackers:

```
\setupmathstackers [both]   [color=darkgray,alternative=mp]
\setupmathstackers [top]    [color=darkgray,alternative=mp]
\setupmathstackers [bottom] [color=darkgray,alternative=mp]
```

$\overline{a+b+c+d}$ $\underline{a+b+c+d}$ $\overline{\underline{a+b+c+d}}$

$\overline{a+b+c+d}$ $\underline{a+b+c+d}$ $\overline{\underline{a+b+c+d}}$

$\overbrace{a+b+c+d}$ $\underbrace{a+b+c+d}$ $\overbrace{\underbrace{a+b+c+d}}$

$\overline{a+b+c+d}$ $\underline{a+b+c+d}$ $\overline{\underline{a+b+c+d}}$

$\overleftarrow{a+b+c+d}$ $\underrightarrow{a+b+c+d}$

$\underleftarrow{a+b+c+d}$ $\underrightarrow{a+b+c+d}$

But, these are just overlays and nothing special: we simply don't use the normal font route not fancy Lua tricks either (in principle MkII could do this). I might upgrade it some day (no real demand so far, just fun stuff).

Real fonts

- For text we need an efficient way to define extra shapes.
- We don't really want inline graphics every time we use a glyph.
- We also want to cut and paste properly.
- Basically the fact that we drop in shapes should be hidden.

- We use the same (generic) subsystem that is also used for color fonts, bitmap emoji, svg fonts, etc.
- Shapes end up as Type3 fonts. These have some specific properties and limitations, but we can actually make Unicode fonts.
- The system is not burdened by much overhead and most happens at embedding time.

```
\definefont[DemoFontA][Serif*default @ 10pt]
\definefont[DemoFontB][Serif*default @ 12pt]
\definefont[DemoFontC][Serif*default @ 14pt]
\definefont[DemoFontD][SerifBold*default @ 14pt]

\startlines
\DemoFontA first\endash second\emdash third\char"2015\relax fourth
\DemoFontB first\endash second\emdash third\char"2015\relax fourth
\DemoFontC first\endash second\emdash third\char"2015\relax fourth
\DemoFontD first\endash second\emdash third\char"2015\relax fourth
\stoplines
```

first–second—thirdfourth

first–second—thirdfourth

first–second—thirdfourth

first–second—thirdfourth

```
\definefontfeature[exampleone][metapost=symbolsone]

\definefont[DemoFontA][Serif*default,exampleone @ 10pt]
\definefont[DemoFontB][Serif*default,exampleone @ 12pt]
\definefont[DemoFontC][Serif*default,exampleone @ 14pt]
\definefont[DemoFontD][SerifBold*default,exampleone @ 14pt]
```

first–second—third—fourth

first–second—third—fourth

first–second—third—fourth

first–second—third—fourth

```
\startMPcalculation{simplefun}

    vardef QuotationDashOne =
        draw image (
            interim linecap := squared ;
            save l ; l := 0.2 ;
            draw (l/2,3) -- (10-l/2,3) withpen pencircle scaled l ;
        )
    enddef ;

    lmt_registerglyphs [
        name     = "symbolsone",
        units    = 10,
        usecolor = true,
        width    = 10,
        height   = 3.1,
        depth    = 0,
    ] ;

    lmt_registerglyph [
        category = "symbolsone",
        unicode  = "0x2015",
        code     = "QuotationDashOne ;"
    ] ;

\stopMPcalculation
```

```
\definefontfeature[exampletwo][metapost=symbolstwo]

\definefont[DemoFontA][Serif*default,exampletwo @ 10pt]
\definefont[DemoFontB][Serif*default,exampletwo @ 12pt]
\definefont[DemoFontC][Serif*default,exampletwo @ 14pt]
\definefont[DemoFontD][SerifBold*default,exampletwo @ 14pt]
```

first–second—third—fourth

first–second—third—fourth

first–second—third—fourth

first–second—third—fourth

```
\startMPcalculation{simplefun}

    vardef QuotationDashTwo =
        draw image (
            interim linecap := squared ;
            save l ; l := 0.4 ;
            string weight ; weight := getparameter "mpsfont" "parentdata" "shared" "rawdata" "metadata" "weight" ;
            if     weight = "semibold" : l := l * 2;
            elseif weight = "bold"     : l := l * 3; fi
            draw (l/2,3) -- (10-l/2,3) withpen pencircle scaled l
            withcolor yellow ;
        )
    enddef ;

    lmt_registerglyphs [
        name     = "symbolstwo",
        units    = 10,
        usecolor = false,
        width    = 10,
        height   = 3.1,
        depth    = 0,
    ] ;

    lmt_registerglyph [
        category = "symbolstwo",
        unicode  = "0x2015",
        code     = "QuotationDashTwo ;"
    ] ;

\stopMPcalculation
```

More examples

We give some examples (these are also in the modules). Overloading math symbols:

```
meta-imp-kindergarten.mkxl
```

Extending fonts with Don Knuths dices and tiles (symbols, ligatures, proper Unicode):

```
meta-imp-gamesymbols.mkxl
```

An implementation of Don Knuths ThirtySix font in various variants (color, random, shapes):

```
meta-imp-threesix.mkxl
```