My Way

December 24, 2020

Metaducks
Jairo A. del Rio

Acknowledgements

Although everybody knows an acknowledgement list will always be incomplete, I want to mention some of the people who made this possible:

Sam Carter for the original idea and the nice package she created and maintains. Paulo Cereda, whose suggestion avoided this module to be called a plain '`ducks`' or a more boring possibility. Hans Hagen, Wolfgang Schuster, Aditya Mahajan, Henri Menke and the ConTEXt user base for answering my questions and giving tips and tricks to properly use ConTEXt with its unbeatable tools. And all TEX users, no matter if Plain, LATEX, ConTEXt or another, for such a great and diverse community. All duck lovers around the world, regardless of language, religion or nationality. Ducks rock (and swim)! For all of you is this module.

Rationale

I love ducks as much as I love my girlfriend, who is a duck breeder, and almost as much I'm abhorred by TikZ's feature creep, its slowness and its huge manual. Don't misunderstand me, I hugely appreciate the effort users and maintainers put on TikZ and even desire some features/libraries to be implemented in ConTEXt, such as `graphdrawing`; however, I still don't like TikZ. Besides that, TikZ is rather geared to LATEX and keeping things right with ConTEXt sometimes is a toothache. I think some TEX users who migrate from LATEX to ConTEXt or just like ducks and ConTEXt, but are uncomfortable with TikZ, would be pleased with this little module.

Differences and caveats

Because this is a port from a package written in TikZ, it's worthwhile to explain differences and limitations with respect to LATEX's TikZducks. There is, for instance, a difference between PGF/TikZ and ConTEXt way of handling with key-value pairs. `pgfkeys` is very happy with the following:

```
\somecommand[option1,key1=value1,key2=value2,option2]
```

ConTEXt, on the other hand, doesn't usually mix those alternatives, and in order not to overcomplicate this module (after all, MetaPost gives faster results when cautiously exploited), this module is based on a rather straightforward key-value interface. For instance, `body=\MPcolor{blue}` won't work and you should use `bodycolor=\MPcolor{blue}` instead. The same applies for eyes, bill and head. Another instance: if you need a crozier for your duck, the way to specify is

```
\ducks
    [
    crozier=true, %Don't forget this!
```

```
    croziercolor={.5[\MPcolor{orange},black]}
    ]
```

If you only set `crozier=true`, a default color will be selected.

Additionally, colors should be passed in a MetaPost-understandable way, e.g. `\MPcolor{brown}`, `1` (RGB white) or `(0.5,0.4,1)`. `xcolor`'s color mixing isn't supported, but remember you're still able to define colors in such a way with `\enabledirectives[colors.pgf]` in your preamble. Like this:

```
\enabledirectives[colors.pgf]
\usecolors[svg]
\usemodule[metaducks]
%Only for two colors!
\definecolor[mycolor][gold!50!violet]
\starttext
\ducks[bodycolor=\MPcolor{mycolor}]
\stoptext
```

Surely a cleverer alternative is possible and I'll have to find out it (possibly guided by Hans and Wolfgang). In the meantime those are your options.

Another point on color: this module includes a companion called `xcolor` (actually `colo-imp-xcolor.mkiv`) so ducks are roughly the same both in ConTEXt and LaTEX. Because of this, we're actually using colors named `xcoloryellow`, `xcolorcyan` and so on to avoid clashes with other color schemes in ConTEXt. So, you should be able of using something like

```
\usecolors[xwi]
\ducks[bodycolor={\MPcolor{gold}}]
```

and keep your ducks of the same color.

As for size, and unlike MetaPost, TikZ defaults to centimeters. To adjust that, I've included an `unitsize` key in order to scale your picture accordingly. Just ensure to pass an unit Metafun/MetaPost can interpret correctly.

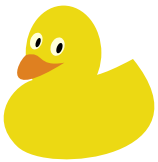
Most of TikZducks options are available, but don't forget to initialize them via `<option>=true` and only then `<option>color=...`, when possible. Random ducks are also available. Obviously you should be familiar with TikZducks manual to see all available options (or you could look at `t-metaducks.mkvi`, too).

So far, stripes aren't implemented (but I promise I'll do soon).

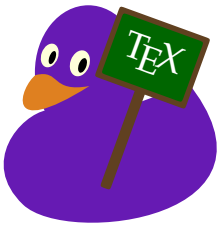
Some examples

```
\ducks
```

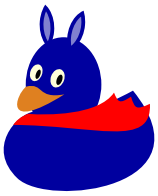
Ducks for ConTEXt!



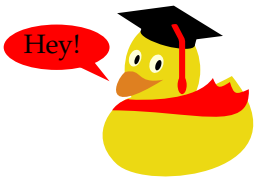
```
\ducks
[
unitsize=1.4142cm,
signpost=true,
bodycolor={(.4,0.1,0.7)},
signtext=\TEX
]
```




```
\randomducks[bodycolor=darkblue]
```



```
\randomducks[speech=true,bubblecolor=red,speechtext=Hey!]
```




```
\ducks[snowduck=true,snowduckcolor=.9]
```

Ducks for ConTEXt!

And a gift for Christmas or whatever you celebrate in December:

```
\ducks[unitsize=2cm,santa=true,mug=true]
```

To do

- A better manual.
- Support for stripes.
- Further customization.
- Easier color specification.
- IMPORTANT: Acess from MP environments.
- Complete this list (!).

source code of this document

```
\usemodule[mag-01]
\usemodule[metaducks]
\startdocument
  [title={Metaducks},
   subtitle={Ducks for \CONTEXT !},
   author={Jairo A. del Rio},
   date=\currentdate,
  ]

\startsubject[title=Acknowledgements]

Although everybody knows an acknowledgement list will always be incomplete, I want to mention some
  of the people who made this possible:

Sam Carter for the original idea and the nice package she created and maintains. Paulo Cereda, whose
  suggestion avoided this module to be called a plain \quote{\type{ducks}} or a more boring possi-
  bility. Hans Hagen, Wolfgang Schuster, Aditya Mahajan, Henri Menke and the \CONTEXT\ user base for
  answering my questions and giving tips and tricks to properly use \CONTEXT\ with its unbeatable
  tools. And all \TEX\ users, no matter if Plain, \LATEX, \CONTEXT\ or another, for such a great and
  diverse community. All duck lovers around the world, regardless of language, religion or national-
  ity. Ducks rock (and swim)! For all of you is this module.

\stopsubject

\startsubject[title=Rationale]

I love ducks as much as I love my girlfriend, who is a duck breeder, and almost as much I'm abhorred
  by TikZ's feature creep, its slowness and its huge manual. Don't misunderstand me, I hugely ap-
  preciate the effort users and maintainers put on TikZ and even desire some features/libraries to
  be implemented in \CONTEXT, such as \type{graphdrawing}; however, I still don't like TikZ. Be-
  sides that, TikZ is rather geared to \LATEX\ and keeping things right with \CONTEXT\ sometimes is
  a toothache. I think some \TEX\ users who migrate from \LATEX\ to \CONTEXT\ or just like ducks and
  \CONTEXT, but are uncomfortable with TikZ, would be pleased with this little module.

\stopsubject

\startsubject[title=Differences and caveats]

Because this is a port from a package written in TikZ, it's worthwhile to explain differences and
  limitations with respect to \LATEX's TikZducks. There is, for instance, a difference between  PGF
  /TikZ and \CONTEXT\ way of handling with key-value pairs. \type{pgfkeys} is very happy with the
  following:

\starttyping
\somecommand[option1,key1=value1,key2=value2,option2]
\stoptyping

\CONTEXT, on the other hand, doesn't usually mix those alternatives, and in order not to overcom-
  plicate this module (after all, \MetaPost\ gives faster results when cautiously exploited), this
  module is based on a rather straightforward key-value interface. For instance, \type{body=\MPcolor
  {blue}} won't work and you should use \type{bodycolor=\MPcolor{blue}} instead. The same applies
  for eyes, bill and head. Another instance: if you need a crozier for your duck, the way to specify
  is

\starttyping
\ducks
    [
    crozier=true, %Don't forget this!
```

source code of this document

```
    croziercolor={.5[\MPcolor{orange},black]}
    ]
\stoptyping

If you only set \type{crozier=true}, a default color will be selected.

Additionally, colors should be passed in a \METAPOST-understandable way, e.g. \type{\MPcolor{brown
  }}, \type{1} (RGB white) or \type{(0.5,0.4,1)}. \type{xcolor}'s color mixing isn't supported, but
  remember you're still able to define colors in such a way with \type{\enabledirectives[colors.pgf]
  } in your preamble. Like this:

\starttyping
\enabledirectives[colors.pgf]
\usecolors[svg]
\usemodule[metaducks]
%Only for two colors!
\definecolor[mycolor][gold!50!violet]
\starttext
\ducks[bodycolor=\MPcolor{mycolor}]
\stoptext
\stoptyping

Surely a cleverer alternative is possible and I'll have to find out it (possibly guided by Hans and
  Wolfgang). In the meantime those are your options.

Another point on color: this module includes a companion called \type{xcolor} (actually \type{colo-
  imp-xcolor.mkiv}) so ducks are roughly the same both in \CONTEXT\ and \LATEX. Because of this, we
  're actually using colors named \type{xcoloryellow}, \type{xcolorcyan} and so on to avoid clashes
  with other color schemes in \CONTEXT. So, you should be able of using something like

\starttyping
\usecolors[xwi]
\ducks[bodycolor={\MPcolor{gold}}]
\stoptyping

and keep your ducks of the same color.

As for size, and unlike \MetaPost, TikZ defaults to centimeters. To adjust that, I've included an
  \type{unitsize} key in order to scale your picture accordingly. Just ensure to pass an unit Meta-
  fun/\MetaPost\ can interpret correctly.

Most of TikZducks options are available, but don't forget to initialize them via \type{<option>=true
  } and only then \type{<option>color=...}, when possible. Random ducks are also available. Obvi-
  ously you should be familiar with TikZducks manual to see all available options (or you could look
  at \type{t-metaducks.mkvi}, too).

So far, stripes aren't implemented (but I promise I'll do soon).

\stopsubject

\startsubject[title=Some examples]

\startbuffer
\ducks
\stopbuffer

\typebuffer \getbuffer

\startbuffer
```

source code of this document

```
\ducks
[
unitsize=1.4142cm,
signpost=true,
bodycolor={(.4,0.1,0.7)},
signtext=\TEX
]
\stopbuffer

\typebuffer \getbuffer

\startbuffer
\randomducks[bodycolor=darkblue]
\stopbuffer

\typebuffer \getbuffer

\startbuffer
\randomducks[speech=true,bubblecolor=red,speechtext=Hey!]
\stopbuffer

\typebuffer \getbuffer

\startbuffer
\ducks[snowduck=true,snowduckcolor=.9]
\stopbuffer

\typebuffer \getbuffer

And a gift for Christmas or whatever you celebrate in December:

\startbuffer
\ducks[unitsize=2cm,santa=true,mug=true]
\stopbuffer

\typebuffer \getbuffer

\stopsubject

\startsubject[title=To do]

\startitemize[packed]
\startitem A better manual. \stopitem
\startitem Support for stripes. \stopitem
\startitem Further customization. \stopitem
\startitem Easier color specification. \stopitem
\startitem IMPORTANT: Acess from MP environments. \stopitem
\startitem Complete this list (!).\stopitem
\stopitemize

\stopsubject

\stopdocument
```

source code of this document