

The **dndicons** package

A set of high quality icons made with *TikZ* for use in
notes for tabletop role-playing games

Jasper Habicht*

Version 1.2.0, released on 20 November 2023

1 Introduction

The **dndicons** package provides a set of high quality icons made with *TikZ* for use in notes for tabletop role-playing games. The icons are meant to be used in the body text, but they can also be used in other contexts such as graphics or diagrams.

Since the commands to typeset the icons typeset **tikzpicture** environments, these commands should not be used inside another **tikzpicture**. However, because the package defines the icons as *TikZ* shapes, it is possible to use the icons in **tikzpicture** environments directly. Apart from that, as of version 1.1.0, the package provides a way to define custom commands to typeset the icons as boxed material which is safe to use in an **tikzpicture** context.

2 Loading the package

The **dndicons** package is loaded by calling `\usepackage{dndicons}` in the preamble of the document. The package loads the **tikz** package. The package does not provide any options.

3 Usage

Once loaded, the package provides a set of commands that can be used to print icons inline. The package also defines a set of node shapes that can be used inside a **tikzpicture** environment.

3.1 Global style and default color

All icons share the *TikZ* style **dnd icon** that has no options per default but can be used to style all icons at once. For example, if the setting `\tikzset{dnd icon/.append style={draw=red}}` is placed at the beginning of

*E-mail: mail@jasperhabicht.de

the document, all icons will be drawn in red. Per default, the icons are drawn in the color of the surrounding text.


Because the package defines the icons as TikZ shapes, every command can actually be used together with every shape. However, the combinations of shapes and commands as described in the following are preferable.


In the following subsections, all commands and shapes are explained in detail. Inside a `tikzpicture` environment, it is possible to place a node using as option the name of one of the shapes defined by the package. For example, `\node[eightside] {};` would output a node with the relevant shape. See the below subsection for more detailed information.


3.2 Icon `\die`











`\die` The command `\die[<style>]{<shape>}[<options>]{<integer>}` prints an icon to depict a die with a certain number of sides. Two special icons exist for a two-sided die (which would be equivalent to a coin) and for a hundred-sided die (which typically comes in the shape of a sphere). There is also a special icon for a fudge die.

For the six-sided die, nine additional shapes exists representing the values one to nine as pips. Also, additional shapes exist representing the plus or minus side of a fudge die.

The command takes two mandatory commands, the first of which describes the shape (see previous subsection) and the second can take an integer that is placed in front of the shape. Thus, `\die[eightside]{2}` results in 2  (meaning 2 eight-sided dice are rolled).

The command also takes two optional arguments, the second of which can take arbitrary TikZ options to style the icon. The options affect the shape, not the integer when it is printed before the icon. As an example, `\die[eightside][blue, thick]{2}` results in 2 .

The first optional argument can take the value `normal` or `large`, `normal` being the default value. With the value `large`, the icon is drawn larger and the additional integer is printed inside of the shape instead of before it. As an example, `\die[large]{eightside}{2}` results in . Note that the integer will always be printed on top of the shape, even if the shape does not have an open center as in the case of the `fudge` shape.

Command	Icon	Shape
<code>\die[<i><style></i>]{<i><shape></i>}[<i><options></i>]{<i><integer></i>}</code>		
		<code>twoside</code>
		<code>fourside</code>
		<code>sixside</code>
		<code>eightside</code>
		<code>tenside</code>
		<code>twelveside</code>
		<code>twentyside</code>
		<code>hundredsideside</code>
		<code>fudge</code>
		<code>sixside one</code>

Command	Icon	Shape
		sixside two
		sixside three
		sixside four
		sixside five
		sixside six
		sixside seven
		sixside eight
		sixside nine
		fudge plus
		fudge minus

3.3 Icons `\ability` and `\saving`









`\ability` The command `\ability[<style>]{<shape>}[<options>]` prints icons depicting an ability of a character. The abilities are represented by animal-like shapes. The relevant shape should be given as mandatory argument to the command. The second optional argument can take arbitrary TikZ options to style the icon.

The first optional argument can take the value `positive` or `negative`, `positive` being the default value. With the value `negative`, the icon is drawn negative inside a circle. As an example, `\ability[negative]{charisma}` results in

`\saving` The command `\saving[<style>]{<shape>}[<options>]` prints the shapes available to the `\ability` icon inside a small shield. It can take the same values for the mandatory argument as the `\ability` command. The optional argument can take arbitrary TikZ options to style the icon.











The first optional argument can take the value `normal` or `empty`, `normal` being the default value. With the value `empty`, the icon inside the shield is not printed. In this case, the mandatory argument can be left empty. As an example, `\saving[empty]{}` results in

Command	Icon	Shape
<code>\ability[<i><style></i>]{<i><shape></i>}[<i><options></i>]</code>		strength
		dexterity
		dexterity alt
		constitution
		intelligence
		wisdom
		charisma
		luck
		armor
		proficiency
<code>\saving{<i><shape></i>}[<i><options></i>]</code>		strength
		dexterity

Command	Icon	Shape
		dexterity alt
		constitution
		intelligence
		wisdom
		charisma
		luck
		armor
		proficiency


3.4 Icon `\spell`






`\spell` The command `\spell{<shape>}[<options>]` prints icons depicting the effect of a spell or how it is to be effected. The optional argument can take arbitrary TikZ options to style the icon.




Command	Icon	Shape
<code>\spell{<shape>}[<options>]</code>		linear
		conic
		quadratic
		cubic
		spheric
		cylindric
		verbal
		somatic
		material
		focus

3.5 Icon `\spellschool`

`\spellschool` The command `\spellschool[<style>]{<shape>}[<options>]` prints icons representing the school a spell belongs to. The second optional argument can take arbitrary TikZ options to style the icon.


























The first optional argument can take the value **negative** or **positive**, **negative** being the default value. With the value **positive**, the icon is drawn negative inside a circle. As an example, `\spellschool[positive]{evocation}` results in .













Command	Icon	Shape
<code>\spellschool[<style>]{<shape>}[<options>]</code>		abjuration
		conjuration
		divination
		enchantment
		evocation

Command	Icon	Shape
		illusion
		necromancy
		transmutation

3.6 Icons `\damage`, `\attack`, and `\condition`

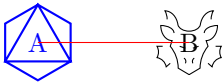
- `\damage` The command `\damage{<shape>}[<options>]` prints icons depicting the damage of an attack. The icon is printed inside a circle. The optional argument can take arbitrary *TikZ* options to style the icon.
- `\attack` The command `\attack{<shape>}[<options>]` prints icons depicting the kind of an attack. The optional argument can take arbitrary *TikZ* options to style the icon.
- `\condition` The command `\condition{<shape>}[<options>]` prints icons depicting a condition of a character. The optional argument can take arbitrary *TikZ* options to style the icon.

Command	Icon	Shape
<code>\damage{<shape>}[<options>]</code>		acid
		bludgeoning
		cold
		fire
		force
		lightning
		necrotic
		piercing
		poison
		psychic
		radiant
		slashing
		thunder
		healing
<code>\attack{<shape>}[<options>]</code>		melee
		ranged
		magic
		singlehanded
		doublehanded
<code>\condition{<shape>}[<options>]</code>		buff
		blinded
		charmed
		deafened
		exhausted
		frightened

Command	Icon	Shape
		grappled
		incapacitated
		invisible
		paralyzed
		petrified
		poisoned
		prone
		restrained
		stunned
		unconscious
		hearing
		seeing

3.7 Direct use of shapes

Because the icons are defined as TikZ shapes, they can directly be applied to TikZ nodes. However, the shapes don't have a shape border and no anchors except for the `center` anchor that sits exactly in the center of the shape. Therefore, if nodes with these shapes are connected using edges, the `center` anchor will be used to connect the nodes. If nodes with these shapes are being positioned, only the `center` anchor is available. Text content of these nodes is simply printed on top of the center of the node. Compare the following example.



```


\begin{tikzpicture}
  \node[eightside, blue, thick]
    at (0,0) (A) {A};
  \node[charisma] at (2,0) (B) {B};
  \draw[red] (A) -- (B);
\end{tikzpicture}

```

3.8 Boxing of icons

Because the icons cannot simply be used inside `tikzpicture` environments, the package provides a workaround to place icons inside of boxes for later use. Icons that are boxed this way can safely be used inside `tikzpicture` environments.

`\provideprotectedndicon` The command `\provideprotectedndicon{<command>}[<style>]{<shape>}[<options>]{<box name>}` can be used to create a box that contains the icon that would be created using one of the regular commands this package provides. For example, `\provideprotectedndicon{die}[large]{eightside}[blue, thick]{mybox}` would store the icon of the eight-sided die with the relevant style and TikZ options into a newly created box named `mybox`. Note that no integer can be added to the `\die` command.

`\useprotectedndicon` Using the command `\useprotectedndicon{<box name>}`, the previously defined box can be used to place the relevant icon. With the above definition having been made, `\useprotectedndicon{mybox}` would result in .