gurps — A LaTeX package for GURPS typesetting*

Nathanael Farley†

Released 1970/01/01

Abstract

GURPS is an excellent RPG system. LaTeX is an excellent typesetting system. Together, they make excellent game aids. At least, they can with this package!

Contents

1 Motivation

I often like to make ***GURPS*** game aids for my own games. Over time, I started to collect various macros that made it easier. Eventually, I had enough game aids that I decided to write the macros up in a package, then I realised I could release on CTAN and share the ***GURPS*** goodness!

2 Quickstart

A brief overview of the commands in this package can be found in Table 1.

3 In-depth usage

The macros in this package can be divided into roughly three sections:

1. character creation
2. common typographical features of ***GURPS***
3. macros for abiding by SJGames online policy

[*] This file describes version v0.3, last revised 1970/01/01.

[†] E-mail: nasfarley88@gmail.com

Table 1: Basic commands in gurps package.

Macro	**Example**
\dice{⟨*no. of dice*⟩}[⟨*modifier*⟩]	3d+2
\gurpsbook{⟨*name of book*⟩}[⟨*page(s)*⟩]	***GURPS Zombies*** p. 1
\gurps	***GURPS***
\SJGamesOnlinePolicyDisclaimer	The material presented here is my original creation, intended for use with the ***GURPS*** system from Steve Jackson Games. This material is not official and is not endorsed by Steve Jackson Games.
\SJGamesOnlinePolicyNotice	***GURPS*** is a registered trademark of Steve Jackson Games, and is copyrighted by Steve Jackson Games. All rights are reserved by Steve Jackson Games. This material is used here in accordance with the SJ Games online policy.
\SJGamesOnlinePolicyGameAid{⟨*author name*⟩}	***GURPS*** is a trademark of Steve Jackson Games, and its rules and art are copyrighted by Steve Jackson Games. All rights are reserved by Steve Jackson Games. This game aid is the original creation of AUTHOR NAME and is released for free distribution, and not for resale, under the permissions granted in the Steve Jackson Games Online Policy.
character environment, \ST, \advantage, etc.	See subsection 3.1

3.1 Character creation

character

Character (i.e. NPCs) creation can be tricky and involve a *lot* of boilerplate code. This package provides most of the boilerplate code so you only specify what is *different* from assumed values (e.g. If your character has a ST 13 but DX 10, IQ 10, HT 10, you only need to specify ST)

It is not currently possible to (easily) change the look of the stats block for the character environment. This is planned for a future release.

Warning!

This feature is still in alpha. The external API (\skill, \advantage, etc.) should not change, but the internal TeX and layout is *very* likely to change.

On to an example[1]:

```
% \begin{character}
%   \ST{9}
%   \HP{12}
%   \advantage{Natural afro}[1]
%   \disadvantage{Big teeth}[-1]
%   \skill{Nunchuck}[DX/Easy]{16}
%   \meleeattack{
%     name=Punch,
%     level=18,
%     damage={5d cr},
%     reach={C,1},
%     notes={Believe it!}
%   }
% \end{character}
%
```

creates ...

Basic Attributes ST 9[-10]; DX 10[0]; IQ 10[0]; HT 10[0]

Secondary Characteristics HP 12[6]; Per 10[0]; Will 10[0]; FP 10[0]; Basic Speed 5[0]; Basic Move 5[0]

Other DR 0; Dodge 8; SM 1; sw 1d-1; thr 1d-2

[1] Any resemblance to Napolean Dynamite is completely ... intentional!

Advantages Natural afro[1]; TeX-nical afro[1]

Disadvantages Big teeth[-1]

Skills Nunchuck 16[20]

Attacks

- ***Punch*** $\boxed{18}$ ***5d cr*** Believe it! *Reach: C,1*

3.2 Typographical features

\dice

Using the \dice{⟨*no. of dice*⟩}[⟨*modifier*⟩] provides dice notation consistent with ***GURPS*** materials. E.g.

\dice{3} → 3d
\dice{4}[-1] → 4d–1
\dice{2}[1] → 2d+1

\gurpsbook

The command \gurpsbook{⟨*book name*⟩}[⟨*page number(s)*⟩] formats ***GURPS*** books like Steve Jackson Games. E.g.

\gurpsbook{High Tech} → ***GURPS High Tech***
\gurpsbook{Zombies}[3] → ***GURPS Zombies*** p. 3
\gurpsbook{Warehouse~23}[1, 3--5] → ***GURPS Warehouse 23*** pp. 1, 3–5

3.3 Online policy macros

\SJGamesOnlinePolicyDisclaimer
\SJGamesOnlinePolicyNotice
\SJGamesOnlinePolicyGameAid

The macros \SJGamesOnlinePolicyDisclaimer, \SJGamesOnlinePolicyNotice and \SJGamesOnlinePolicyGameAid{⟨*author name*⟩} provide common boilerplate statements taken directly from SJGames online policy with small modifications to make them more suitable to game aids created with LuaLaTeX.

\gurps

The command \gurps is also included to provide bold and italicised ***GURPS*** as requested in the online policy

3.3.1 \SJGamesOnlinePolicyDisclaimer text

Using the code \SJGamesOnlinePolicyDisclaimer produces the text:

The material presented here is my original creation, intended for use with the ***GURPS*** system from Steve Jackson Games. This material is not official and is not endorsed by Steve Jackson Games.

3.3.2 \SJGamesOnlinePolicyNotice text

Using the code \SJGamesOnlinePolicyOnlineNotice produces the text:

GURPS is a registered trademark of Steve Jackson Games, and is copyrighted by Steve Jackson Games. All rights are reserved by Steve Jackson Games. This material is used here in accordance with the SJ Games online policy.

3.3.3 \SJGamesOnlinePolicyGameAid text

Using the code \SJGamesOnlinePolicyGameAid{NAME} produces the text:

GURPS is a trademark of Steve Jackson Games, and its rules and art are copyrighted by Steve Jackson Games. All rights are reserved by Steve Jackson Games. This game aid is the original creation of NAME and is released for free distribution, and not for resale, under the permissions granted in the Steve Jackson Games Online Policy.

4 Implementation

Loading required libraries and the lua logic for this class.

```
⟨*package⟩
\RequirePackage{etoolbox}
\RequirePackage{xparse}
\RequirePackage{xspace}
\RequirePackage{luacode}
\RequirePackage{tikz}
\RequirePackage{calc}
\PassOptionsToPackage{inline}{enumitem}
\RequirePackage[inline]{enumitem}

\RequirePackage{hyperref}
\RequirePackage{xstring}
\RequirePackage{xkeyval}

\luadirect{require("gurps.lua")}
\luadirect{require("gurps_character.lua")}
```

`\gurps` Steve Jackson Games asks that the name ***GURPS*** is always in bold and italicised. This macro provides that.

```
\NewDocumentCommand{\gurps}{s}{%
    \IfBooleanTF#1
    % With star
    {\href{http://www.sjgames.com/gurps/}{\textbf{\textit{GURPS}}\xspace}}
    % Without star
    {\textbf{\textit{GURPS}}\xspace}%
}
```

`\dice` Notates dice in the ***GURPS*** way (e.g. drops the 6 from d6) and puts it in an `\mbox`. First argument number of dice. Second argument is a modifier and optional. E.g. `\dice{3}[1]` creates 3d+1.

```
\NewDocumentCommand{\dice}{mO{0}}{%
  \luadirect{print_dice([[#1]], [[#2]])}%
}
```

`\gurps@porpp` Helper macro to print pp. or p. when listing page numbers

```
\NewDocumentCommand{\gurps@porpp}{m}{\IfInteger{#1}{p.}{pp.}}
```

`\gurpsbook` Works like this: ***GURPS Horror*** p. 45 or ***GURPS Banestorm*** pp. 3, 45–48

```
\NewDocumentCommand{\gurpsbook}{mo}{%
  \textbf{\textit{\gurps #1}}\IfValueT{#2}{ \gurps@porpp{#2}~#2}%
}
```

`\SJGames` Steve Jackson Games printed with link (starred) or without (unstarred).

```
\NewDocumentCommand{\SJGames}{s}{%
    \IfBooleanTF#1%
    % With star
    {\href{http://www.sjgames.com/}{Steve~Jackson~Games}\xspace}%
    % Without star
    {Steve~Jackson~Games\xspace}%
}
```

`\SJGamesOnlinePolicyDisclaimer` Online policy disclaimer, basically straight from the Steve Jackson Games online policy.

```
\NewDocumentCommand{\SJGamesOnlinePolicyDisclaimer}{}{%
  The material presented here is my original creation, intended for use with the
  \gurps* system from \SJGames*. This material is not official and is
  not endorsed by \SJGames.
}
```

`\SJGamesOnlinePolicyNotice` The disclaimer *almost* as it appears on the online policy. The words ‘the art’ have been removed since by default no art (including logos) are included in LaTeX documents.

```
\NewDocumentCommand{\SJGamesOnlinePolicyNotice}{}{%
  \gurps* is a registered trademark of \SJGames*, and is copyrighted by
  \SJGames. All rights are reserved by \SJGames. This material is used
  here in accordance with the SJ Games
  \href{http://www.sjgames.com/general/online_policy.html}{online policy}.
}
```

\SJGamesOnlinePolicyGameAid This text is required for all game aid's produced for ***GURPS*** but without an official license. It takes one argument: author name.

```
\NewDocumentCommand{\SJGamesOnlinePolicyGameAid}{m}{%
  \gurps* is a trademark of \SJGames*, and its rules and art are
  copyrighted by \SJGames*. All rights are reserved by
  \SJGames. This game aid is the original creation of #1 and
  is released for free distribution, and not for resale, under the
  permissions granted in the
  \href{http://www.sjgames.com/general/online_policy.html}%
        {\SJGames Online Policy}.
}
```

4.1 Character environment

This section lists the `character` environment and all the macros that go into it.

Future plans!

At the moment, there's a real mix of commands with and without `@` that should probably be switched to be internal/external so that the user can configure them (e.g. `\gurps@char@print@attr`). In the future, this will happen. For now ... feel free to edit them and see what happens! :)

Also in the future, there will eventually be 'theme' support such that the user can configure different ways of printing the character (e.g. disable points for a monster manual, use a `tcolorbox` for the statblock, etc.). Can't say when yet, but pull requests are welcome!

character, character* This environment defines a 'character' i.e. anything with full stats. A `character*` will define a character but not print it.

It takes `[`⟨*character key*⟩`]` which provides a means to access the character later in the file. It is set with `\SetCharacterKey` and reset with `\ResetCharacterKey`. In order to access the character later, just use `\SetCharacterKey{`⟨*character key*⟩`}`, use the commands (e.g. `\gurps@char@advantages`) and then reset[2].

```
\NewDocumentEnvironment{character}{sO{_}}{%
  \SetCharacterKey{#2}%
  \luadirect{new_character(_GCHARACTERKEY)}%
}{%
  \gurps@char@checkandfixattrsandpoints%
  \IfBooleanTF{#1}{}{\printcharacter}%
  \ResetCharacterKey%
}
```

\SetCharacterKey Sets the key for the current character. This sets the lua string _GCHARACTER which acts as a key for the global Lua table _GCHARACTERS.

```
\NewDocumentCommand{\SetCharacterKey}{m}{%
  \luadirect{_GCHARACTERKEY = \luastring{\unexpanded{#1}}}%
}
```

\ResetCharacterKey Resets the character key to _.

```
\NewDocumentCommand{\ResetCharacterKey}{}{%
  \SetCharacterKey{_}%
}
```

[2] Although resetting is not strictly necessary, global variables are a Bad Thing™ so we reset in order to emulate some kind of scope. Resetting just changes the character key to _.

\gurps@char@insertattr

Macro for inserting an attribute to the character. Anything inserted with :nolatex: in the name is not added to the character. (This is to aid importing directly from a *.gcs file.)

First, we define the xkeyval keys for the function. These are set up to convert the values directly into the right Lua types.

```
\define@key{gurps@char@insertattr}{name}{%
  \def\gurps@char@insertattr@name{\unexpanded{[[#1]]}}} %TODO check this << has
                                   %fixed it
\define@key{gurps@char@insertattr}{type}{%
  \def\gurps@char@insertattr@type{\luastring{\unexpanded{#1}}}}
\define@key{gurps@char@insertattr}{level}{%
  \def\gurps@char@insertattr@level{#1}}
\define@key{gurps@char@insertattr}{diceexpr}{%
  \def\gurps@char@insertattr@diceexpr{\luastring{#1}}}
\define@key{gurps@char@insertattr}{points}{%
  \def\gurps@char@insertattr@points{#1}}
\define@key{gurps@char@insertattr}{basedon}{%
  \def\gurps@char@insertattr@basedon{\luastring{\unexpanded{#1}}}}
\define@key{gurps@char@insertattr}{difficulty}{%
  \def\gurps@char@insertattr@difficulty{\luastring{\unexpanded{#1}}}}
```

Then we define \gurps@char@insertattr{⟨*key-value pairs*⟩}. The macro first checks we are in a character environment (throwing an error if we are not), defines the keys, then dœs the Lua stuff.

Side note: anything with :nolatex: in the name is discarded and not put into the character.

```
\NewDocumentCommand{\gurps@char@insertattr}{m}{
  \def\@myenvname{character}
  \ifx\@currenvir\@myenvname
  % pass
  \else
  \PackageError{gurps}{Not in character environment! Currently in \@currenvir{} environment.}{}
  \fi

  \setkeys{gurps@char@insertattr}{
    name=NotSet,
    type=NotSet,
    points=nil,
    level=nil,
    basedon=DX,
    difficulty=NotSet,
    diceexpr=NotSet,
  }

  \setkeys{gurps@char@insertattr}{#1}

  % If the name has the tag :nolatex:, don't add it! Otherwise, add it.
  \luadirect{
    if not string.find(\luastring{\gurps@char@insertattr@name}, \luastring{:nolatex:}) then
      % If arg is blank (e.g. identity()), then this returns nil.
      function identity(x)
        return x
      end

      x = {
        name=\gurps@char@insertattr@name,
        type=\gurps@char@insertattr@type,
        points=\gurps@char@insertattr@points,
        level=identity(\gurps@char@insertattr@level),
        basedon=\gurps@char@insertattr@basedon,
        difficulty=\gurps@char@insertattr@difficulty,
        diceexpr=\gurps@char@insertattr@diceexpr
      }

      if_else_packageerror(
      function() return is_valid_type(x.type) end,
      "Type '" .. x.type .. "' is not a valid attribute type!"
      .. " Did you remember to set the type?"
      )

      if_else_packageerror(
      function() return is_valid_points(x.points) end,
      "Points value '" .. tostring(x.points) .. "' is not valid!"
      )

      if x.type == "skill" or x.type == "spell" then
        if_else_packageerror(
          function() return is_valid_difficulty(x.difficulty) end,
          "Difficulty value '" .. x.difficulty .. "' is not valid!"
        )
        end
      insert_attr(_GCHARACTERKEY, x)
    end
  }%
}
```

4.2 A few toggles

These should be set by lua stuff.

```
\newtoggle{hasadvantages}
\newtoggle{hasdisadvantages}
\newtoggle{hastraits}
\newtoggle{hasskills}
\newtoggle{hasspells}
\newtoggle{hasattacks}
```

\gurps@char@checkandfixtoggles

This command fixes all the toggles used in printing the character.

```
\NewDocumentCommand{\gurps@char@checkandfixtoggles}{}{%
\luadirect{check_and_fix_toggles(_GCHARACTERKEY)}%
}
```

4.3 Printing the character

\printcharacter

(This function is mostly for debug at the moment)

```
\NewDocumentCommand{\printcharacter}{o}{%
  \IfValueTF{#1}{\SetCharacterKey{#1}}{}%
  \gurps@char@checkandfixtoggles%
  \charactersection*{Basic Attributes}%
  \gurps@char@basicattributes%
  \charactersection*{Secondary Characteristics}%
  \gurps@char@secondarycharacteristics%
  \charactersection*{Other}%
  \gurps@char@properties%
  \iftoggle{hasadvantages}{%
    \charactersection*{Advantages}%
    \gurps@char@advantages%
  }{}%
  \iftoggle{hasdisadvantages}{%
    \charactersection*{Disadvantages}%
    \gurps@char@disadvantages%
  }{}%
  \iftoggle{hasskills}{%
    \charactersection*{Skills}%
    \gurps@char@skills%
  }{}%
  \iftoggle{hasspells}{%
    \charactersection*{Spells}%
    \gurps@char@spells%
  }{}%
  \iftoggle{hasattacks}{%
    \charactersection*{\strut Attacks}\ \linebreak%
    \gurps@char@attacks%
  }{}%
  \IfValueTF{#1}{\ResetCharacterKey}{}%
}
```

\gurps@char@getattrlevel

```
\NewDocumentCommand{\gurps@char@getattrlevel}{m}{%
  \luadirect{
    x = filter({name=\luastring{#1}}, get_character(_GCHARACTERKEY))
    if x then
      tex.sprint(x.level)
    else
      tex.sprint([[\PackageError{gurps}{Stat '#1' not found!}]])
    end
  }%
}
```

4.3.1 Basic attributes

\ST Sets the strength of the current character

```
\NewDocumentCommand{\ST}{mO{nil}}{
  \gurps@char@insertattr{
    name=ST,
    type=basic_attribute,
    level=#1,
    points=#2
  }%
}
```

\DX Sets the dexterity of the current character

```
\NewDocumentCommand{\DX}{mO{nil}}{%
  \gurps@char@insertattr{
    name=DX,
    type=basic_attribute,
    level=#1,
    points=#2
  }%
}
```

\IQ Sets the IQ of the current character

```
\NewDocumentCommand{\IQ}{mO{nil}}{%
  \gurps@char@insertattr{
    name=IQ,
    type=basic_attribute,
    level=#1,
    points=#2
  }%
}
```

\HT Sets the health of the current character

```
\NewDocumentCommand{\HT}{mO{nil}}{%
  \gurps@char@insertattr{
    name=HT,
    type=basic_attribute,
    level=#1,
    points=#2
  }%
}
```

4.3.2 Secondary Characteristics

This section defines secondary characteristics (as ***GURPS Basic Set***). Points are set in a later function `\gurps@char@checkandfixattrsandpoints` if they are not set here.

`\HP` Sets the HP of the current character

```
\NewDocumentCommand{\HP}{mO{nil}}{%
  \gurps@char@insertattr{
    name=HP,
    type=secondary_characteristic,
    level=#1,
    points=#2
  }%
}
```

`\Per` Sets the perception of the current character

```
\NewDocumentCommand{\Per}{mO{nil}}{%
  \gurps@char@insertattr{
    name=Per,
    type=secondary_characteristic,
    level=#1,
    points=#2
  }%
}
```

`\Will` Sets the will of the current character

```
\NewDocumentCommand{\Will}{mO{nil}}{%
  \gurps@char@insertattr{
    name=Will,
    type=secondary_characteristic,
    level=#1,
    points=#2
  }%
}
```

`\FP` Sets the FP of the current character

```
\NewDocumentCommand{\FP}{mO{nil}}{%
  \gurps@char@insertattr{
    name=FP,
    type=secondary_characteristic,
    level=#1,
    points=#2
  }%
}
```

`\SM` Sets the SM of the current character

```
\NewDocumentCommand{\SM}{m}{%
  \gurps@char@insertattr{
    name=SM,
    type=property,
    level=#1
  }%
}
```

`\DR` Sets the DR of the current character. Takes `\DR[`⟨*specialisation*⟩`]{`⟨*level*⟩`}`.

```
\NewDocumentCommand{\DR}{om}{
  \IfValueTF{#1}{%
    \gurps@char@insertattr{%
      name={DR (#1)},%
      type=property,%
      level=#2%
    }%%
  }{%
```

```
    \gurps@char@insertattr{%
      name=DR,%
      type=property,%
      level=#2%
    }%
  }%
}
```

\thr Sets the thr of the current character

```
\NewDocumentCommand{\thr}{m}{
  \gurps@char@insertattr{
    name=thr,
    type=property,
    diceexpr=#1
  }%
}
```

\sw Sets the sw of the current character

```
\NewDocumentCommand{\sw}{m}{
  \gurps@char@insertattr{
    name=sw,
    type=property,
    diceexpr=#1
  }%
}
```

\basicspeed Sets the Basic Speed of the current character

```
\NewDocumentCommand{\basicspeed}{mO{nil}}{
  \gurps@char@insertattr{
    name={Basic Speed},
    type=secondary_characteristic,
    level=#1,
    points=#2
  }%
}
```

\basicmove Sets the Basic Move of the current character

```
\NewDocumentCommand{\basicmove}{mO{nil}}{
  \gurps@char@insertattr{
    name={Basic Move},
    type=secondary_characteristic,
    level=#1,
    points=#2
  }%
}
```

\gurps@char@insertattack

```
\define@key{gurps@char@insertattack}{name}{
  \def\gurps@char@insertattack@name{\luastring{\unexpanded{#1}}}}
\define@key{gurps@char@insertattack}{type}{
  \def\gurps@char@insertattack@type{\luastring{\unexpanded{#1}}}}
\define@key{gurps@char@insertattack}{damage}{
  \def\gurps@char@insertattack@damage{\luastring{\unexpanded{#1}}}}
\define@key{gurps@char@insertattack}{level}{\def\gurps@char@insertattack@level{#1}}
\define@key{gurps@char@insertattack}{reach}{
  \def\gurps@char@insertattack@reach{\luastring{\unexpanded{#1}}}}
\define@key{gurps@char@insertattack}{range}{
  \def\gurps@char@insertattack@range{\luastring{\unexpanded{#1}}}}
\define@key{gurps@char@insertattack}{notes}{
  \def\gurps@char@insertattack@notes{\luastring{\unexpanded{#1}}}}

  % Set default values
\setkeys{gurps@char@insertattack}{
  name=NotSet,
  type=NotSet,
  damage=0d,
  level=nil,
  reach=NotSet,
  range=NotSet,
  notes={},
}
\NewDocumentCommand{\gurps@char@insertattack}{sm}{
  \def\@myenvname{character}
  \ifx\@currenvir\@myenvname
  % pass
  \else
  \PackageError{gurps}{Not in character environment! Currently in \@currenvir{} environment.}{}
  \fi

  \IfBooleanTF{#1}{}{
    \setkeys{gurps@char@insertattack}{
      name=NotSet,
      type=NotSet,
      damage=0d,
      level=nil,
      reach=NotSet,
      range=NotSet,
      notes={},
    }

    \setkeys{gurps@char@insertattack}{#2}
  }
  % \show\gurps@char@insertattack@name
  % \show\gurps@char@insertattack@type
  % \show\gurps@char@insertattack@level
  % \show\gurps@char@insertattack@damage
  % \show\gurps@char@insertattack@reach
  % \show\gurps@char@insertattack@range
  % \show\gurps@char@insertattack@notes
  \luadirect{x = {
      name=\gurps@char@insertattack@name,
      type=\gurps@char@insertattack@type,
      level=\gurps@char@insertattack@level,
      damage=\gurps@char@insertattack@damage,
      reach=\gurps@char@insertattack@reach,
      range=\gurps@char@insertattack@range,
      notes=\gurps@char@insertattack@notes,
    }
    if_else_packageerror(
      function() return is_valid_type(x.type) end,
      "Type '" .. x.type .. "' is not a valid attribute type!"
        .. " Did you remember to set the type?"
    )
    insert_attr(_GCHARACTERKEY, x)
  }
}

```

\rangedattack

```
\NewDocumentCommand{\rangedattack}{m}{%
  \setkeys{gurps@char@insertattack}{#1}%
  \setkeys{gurps@char@insertattack}{type=ranged_attack}

  \gurps@char@insertattack*{}
}
```

\gurps@char@print@attack

```
\NewDocumentCommand{\gurps@char@print@attack}{mmmO{RangeOrReach}mm}{%
  \textbf{\textit{#1 \fbox{#2} #3}} #6 \textit{#4: #5}%
}
```

\meleeattack

```
\NewDocumentCommand{\meleeattack}{m}{%
  \setkeys{gurps@char@insertattack}{#1}%
  \setkeys{gurps@char@insertattack}{type=melee_attack}

  \gurps@char@insertattack*{}
}
```

\advantage Adds an advantage to a character. NOTE: this only works in a `character`.

```
\NewDocumentCommand{\advantage}{mO{nil}}{%
  \gurps@char@insertattr{%
    name={#1},%
    type=advantage,%
    points=#2%
  }%
}
```

\disadvantage Adds a disadvantage to a character. NOTE: this only works in a `character` or `lens` environment.

```
\NewDocumentCommand{\disadvantage}{mO{nil}}{%
  \gurps@char@insertattr{%
    name={#1},%
    type=disadvantage,%
    points=#2%
  }%
}
```

\levelledadvantage Adds an advantage to a character. NOTE: this only works in a `character` or `lens` environment.

```
% TODO finish this!
\NewDocumentCommand{\levelledadvantage}{mmO{nil}}{%
  \gurps@char@insertattr{%
    name={#1},%
    type=advantage,%
    level=#2,
    points=#3%
  }%
}
```

\levelleddisadvantage Adds a disadvantage to a character. NOTE: this only works in a `character` or `lens` environment.

```
\NewDocumentCommand{\levelleddisadvantage}{mmO{nil}}{%
  \gurps@char@insertattr{%
    name={#1},%
    type=disadvantage,%
    level=#2,
    points=#3%
  }%
}
```

\gurps@char@SpellOrSkill Macro for skills or spells (since functionally they are the same in ***GURPS***.). The `[`⟨*points*⟩`]` can be a specifier e.g. `DX/Easy` or points e.g. `48`.

```
\NewDocumentCommand{\gurps@char@SpellOrSkill}{mm}{
  \NewDocumentCommand{#1}{mom}{%
    \IfValueTF{##2}{%
      \makeatletter%
      \luadirect{
        tmpone = \luastring{\unexpanded{##1}}
        tmptwo = \luastring{\unexpanded{##2}}
        tmpthree = \luastring{\unexpanded{##3}}
      }%
      \IfInteger{##2}{%
```

```
        \luadirect{
          tex.sprint(
          [[\gurps@char@insertattr{]]
            .. [[name={]] .. tmpone .. [[},]]
            .. [[type=#2,]]
            .. [[points=##2,]]
            .. [[level=]] .. tmpthree
            .. "}"
          )
       }
      }{%
        \luadirect{
          basedon = tmptwo:gsub('/.*', '')
          difficulty = tmptwo:gsub('.*/', '')
          tex.sprint(
          [[\gurps@char@insertattr{]]
            .. [[name={]] .. tmpone .. [[},]]
            .. [[type=#2,]]
            .. [[basedon=]] .. basedon .. ','
            .. [[difficulty=]] .. difficulty .. ','
            .. [[level=]] .. tmpthree
            .. "}"
          )
        }%
      }%
      \makeatother%
    }{%
      \gurps@char@insertattr{%
        name={##1},%
        type=#2,%
        level=##3
      }%
    }%
  }
}
```

\skill

```
\gurps@char@SpellOrSkill{\skill}{skill}
```

\spell

```
\gurps@char@SpellOrSkill{\spell}{spell}
```

4.3.3 Stat block formatting

\charactertitle As a title for the character stats block, we define \charactertitle

```
\newcounter{charactertitle} % to keep LaTeX happy
\newcommand{\charactertitlemark}[1]{} % ditto
\newcommand\charactertitle{%
  \@startsection{charactertitle}%
  {100}%        level for secnumdepth and tocdepth
  {\z@}%       indentation
  {\topsep}%   space before
  {\topsep}%   space below
  {\raggedright\Large\bfseries}% format of the text
}
```

\charactersection For sections in the the character stats block, we define \charactersection

```
\newcounter{charactersection} % to keep LaTeX happy
\newcommand{\charactersectionmark}[1]{} % ditto
\newcommand\charactersection{%
  \@startsection{charactersection}%
    {101}%        level for secnumdepth and tocdepth
    {\z@}%       indentation
  {\topsep}%   space before
  {-1em}%   space below
  {\raggedright\bfseries}% format of the text
}
% \NewDocumentCommand{\charactersection}{m}{%
%   \noindent\par\noindent\textbf{\large #1}\\\noindent%
% }
```

who's code is taken almost without change from `https://tex.stackexchange.com/questions/97565/flexible-vspac`

charactertraitlist

We define the `charactertraitlist` to list the traits. By default, this is an `itemize*` from the `enumitem` package with some customisations

```
\NewDocumentEnvironment{charactertraitlist}{}
{\noindent\begin{itemize*}[itemjoin={{; }},label=,afterlabel={}]}
  {\end{itemize*}}
```

attacklist

We define the `attacktraitlist` to list the traits. By default, this is an `itemize*` from the `enumitem` package with some customisations

```
\NewDocumentEnvironment{attacklist}{}
{%
  \noindent%
  \begin{itemize}[%
    % itemjoin={\\},% TODO figure out why this causes a problem!
    label={--},%
    wide,
    nosep
    % afterlabel={},% Default of ~ is good
    ]%
  }{\end{itemize}}
```

\gurps@char@print@attr

It's like this: `\gurps@char@print@attr[`⟨*level/diceexpr*⟩`]{`⟨*name*⟩`}[`⟨*points*⟩`]`. This macro is usually called by the lua function `attr_to_tex` (which returns a string)

```
\NewDocumentCommand{\gurps@char@print@attr}{omo}{%
  \IfValueTF{#3}{%
    \def\tmp@points{[#3]}%
  }{%
    \def\tmp@points{}%
  }%
  \IfValueTF{#1}{%
    #2~#1\tmp@points%
  }{%
    #2\tmp@points%
  }%
}
```

\gurps@char@attacks

```
\NewDocumentCommand{\gurps@char@attacks}{}{%
  \luadirect{attacklist(_GCHARACTERKEY)}
}
```

\gurps@char@TraitListMaker

```
\NewDocumentCommand{\gurps@char@TraitListMaker}{mmO{nil}}{
  \NewDocumentCommand{#1}{}{%
    \makeatletter%
    \luadirect{traitlistmaker(#2, _GCHARACTERKEY, #3)}%
    \makeatother%
  }%
}
```

\gurps@char@totalpoints

```
\NewDocumentCommand{\gurps@char@totalpoints}{O{_}}{%
  \luadirect{tex.sprint(sum_points(\luastring{\unexpanded{#1}}))}~pt%
}
```

\gurps@char@traits

```
537 \gurps@char@TraitListMaker{\gurps@char@traits}{is_trait}
```

\gurps@char@advantages

```
538 \gurps@char@TraitListMaker{\gurps@char@advantages}{is_advantage}
```

\gurps@char@disadvantages

```
539 \gurps@char@TraitListMaker{\gurps@char@disadvantages}{is_disadvantage}
```

\gurps@char@skills

```
540 \gurps@char@TraitListMaker{\gurps@char@skills}{is_skill}
```

\gurps@char@spells

```
541 \gurps@char@TraitListMaker{\gurps@char@spells}{is_spell}
```

\gurps@char@basicattributes

```
542 \gurps@char@TraitListMaker{\gurps@char@basicattributes}{is_basic_attribute}[basic_attributes_sorter]
```

ps@char@secondarycharacteristics

```
543 \gurps@char@TraitListMaker{\gurps@char@secondarycharacteristics}{is_secondary_characteris-
    tic}[secondary_characteristics_sorter]
```

\gurps@char@properties

```
544 \gurps@char@TraitListMaker{\gurps@char@properties}{is_property}
```

s@char@checkandfixattrsandpoints

```
545 \NewDocumentCommand{\gurps@char@checkandfixattrsandpoints}{}{%
546   \luadirect{check_and_fix_attrs_and_points(_GCHARACTERKEY)}%
547 }

548 \endinput
549 ⟨/package⟩
```

Change History

v0.1

General: First public release 1

v0.1.1

General: Minor changes to CTAN upload 1

v0.2

General: Better character environment; \gurpsbook command added. 1

v0.3

General: Completely changed underlying character lua structure. Removed lens environment (to be put in at a later date). TODO 1

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.