

SNORE

61

Generated by Doxygen 1.5.5

Wed Sep 3 13:40:00 2008

Contents

1	SNORE	1
1.1	What is BloodBowl?	1
1.2	What is TowBowlTactics?	2
1.3	What is SNORE?	2
2	Workflow	3
2.1	Program Flow	4
2.2	The four different actions	4
3	Frequently Anticipated Questions	7
3.1	How do I install SNORE?	8
3.2	Do I need the Smarty Template Engine?	8
3.3	How do I add support for another language?	8
4	License	9
5	Class Index	17
5.1	Class List	17
6	File Index	19
6.1	File List	19
7	Class Documentation	21
7.1	mySmarty Class Reference	21
7.2	Races Class Reference	24
7.3	Skills Class Reference	26
7.4	TeamLoader Class Reference	27
7.5	TeamSaver Class Reference	30
7.6	Translation Class Reference	33
7.7	UserInterface Class Reference	36

8	File Documentation	37
8.1	config.php File Reference	37
8.2	faq.doxy File Reference	39
8.3	helper.php File Reference	40
8.4	index.php File Reference	41
8.5	license.doxy File Reference	42
8.6	Races.php File Reference	43
8.7	Skills.php File Reference	44
8.8	snore.doxy File Reference	45
8.9	TeamLoader.php File Reference	46
8.10	TeamSaver.php File Reference	47
8.11	Translation.php File Reference	48
8.12	UserInterface.php File Reference	49
8.13	workflow.doxy File Reference	50

Chapter 1

SNORE

Version:

SVN 71

Author:

Marc-Antonio Bisotti

Date:

06.2008

1.1 What is BloodBowl?

Note:

From the Wikipedia Article <http://en.wikipedia.org/wiki/Bloodbowl>

Blood Bowl is a two-player, turn-based board game that typically uses 28 mm miniatures to represent a contest between two teams on a playing field. A board containing a grid overlay represents the field. Using dice, cards, and counters, the players attempt to outscore each other by entering the opponent's end zone with a player who possesses the ball.

The "Blood" in Blood Bowl is represented by the violent actions available to players. Game play is based on a hybrid of American Football, Rugby, and ultra-violent fictional sports events such as Rollerball. Players may attempt to injure or maim the opposition in order to make scoring easier by reducing the number of enemy players on the field.

The player races are drawn from the ranks of fantasy races and have characteristics that reflect the abilities of those races. Elves tend to be agile and good at scoring, while Dwarves and Orcs are more suited to a grinding, physical style of play. Players are also divided up into positions suited to their role on the field, including Throwers, Catchers, and Linemen.

In league play, players gain additional skills and abilities based on their accumulation of experience points. Players face potential injury or even death on the field throughout their careers. Teams improve by the purchase of off-field staff such as cheerleaders, assistant coaches, and apothecaries.

1.2 What is TowBowlTactics?

Towbowl Tactics est le portage informatique du célèbre jeu de plateau Blood Bowl. Projet né en 2001 de la volonté d'une seule personne : Pascal Bourut (Toweld), fan de BloodBowl et désirant retrouver sur machine les moments partagés autour d'une table de jeu.

Depuis, de multiples versions se sont succédés suivant un rythme très aléatoire car on manque toujours de main d'œuvre et de compétences.

SDL et IA bienvenus. Développeurs, le document de référence sur les avancées du code, les choix technologiques et les règles établies est à l'heure actuelle [propal.pdf](#). Toute la documentation nécessaire (compilation, track...) se trouve sur le site, [ici](#). Alors, si vous vous sentez une âme d'aventurier, si vous voulez participer à la programmation d'un jeu de football fantastique, n'hésitez pas à laisser un message sur le forum ou à contacter tuxrouge (admin-arobase-towbowltactics-point-com) qui se fera un plaisir de répondre à toutes vos questions et de vous harceler par la suite. Et, bien sûr, vous pouvez nous rejoindre à la taverne avec irc (serveur: [irc.freenode.net](irc://irc.freenode.net) canal: [tbt](#)).

1.3 What is SNORE?

1.3.1 About SNORE

SNORE (Super New Online Roster Editor) is an online replacement for the paper rosters you would use for the board game. It helps you to keep track of your players during league play, or simply to line up a fresh team for your next friendly match. Saving and loading of rosters is supported, additionally, you can print out the rosters in a printer-friendly format.

SNORE follows the rules of the LRB 5.0 (journeymen, grab/frenzy-exclusiveness, ...), but allows you to disobey the rules (you can't do so by inattention) should you wish to do so.

The chosen license is the GPL V2.

1.3.2 About this Documentation

This documentation tries to cover the server side of things, especially what one could call the Model and Controller aspects. If you want to learn about the templates, consult Smarty's (the used template engine) documentation here: <http://www.smarty.net/docs.php>

To date, there is no documentation of the client side (i.e. Javascript). For help, check the project's forums: http://www.towbowltactics.com/e107_plugins/forum/forum.php

Chapter 2

Workflow

2.1 Program Flow

The file [index.php](#) in the base directory controls the application's flow, which is composed of 4 different actions:

- the user can save a roster he filled out
- show an empty roster for a race the user selected
- load a roster the user uploaded and show it
- show the welcome page which serves as a gate for the requests the user can make

2.2 The four different actions

To determine what to do, the application checks PHP's POST and GET variables in a specific order.

2.2.1 Saving

```
if ( isset($_POST['TEAM']) ) {
    header('Content-type: application/xml');
    header('Content-Disposition: attachment; filename="'. $_POST['TEAM'] .'.xml"');
    echo TeamSaver::save($_POST);
}
```

If `$_POST['TEAM']` is set, it means the user has filled out a roster and clicked the submit button. The data is then delegated to the [TeamSaver](#) class, which processes the input and returns XML, which is presented to the user.

2.2.2 New Roster

If a valid race was selected, show an empty roster for that race.

```
elseif ( isset($_GET['race']) ) {
    $race_id = htmlentities($_GET['race']);
    if ( is_numeric($race_id) && $race_id >= 0 && $race_id <= 20 ) {
        show_roster($race_id);
    }
}
```

But if it is not valid, the user is redirected to the welcome page.

```
else {
    $getError_race = TRUE; // for the template
    show_index(); // race is not valid, show welcome-page instead of roster
}
```

2.2.3 Loading

If `$_POST['upload']` is set, the user uploaded a roster.

```
elseif ( isset($_POST['upload']) ) {
```

The roster is loaded, and the roster is shown:


```
$team = TeamLoader::load($_FILES['userfile']['tmp_name']);  
show_roster($team['id'], $team);
```

Again, if there was a problem, the user is redirected to the welcome page. The var `getError_upload` is set to toggle the display of a warning to the user.

```
$getError_upload = TRUE; // for the template  
show_index(); // there was a problem with the upload, show welcome-page
```

2.2.4 Default

Finally, if nothing happened, show the welcome-page.

```
else {  
    show_index();  
}
```


Chapter 3

Frequently Anticipated Questions

3.1 How do I install SNORE?

Copy all the files to your web server. The latter will need write access to the folders under *views/tmp*.

SNORE was tested with Apache and Lighttpd, and will only work with version 5 of PHP (to the technically inclined: because of the use of the XmlWriter and XmlReader class).

3.2 Do I need the Smarty Template Engine?

SNORE comes bundled with the smarty template engine. You will find the installation in your snore directory under *libs/smarty*.

However, if you want to use your own installation of smarty, feel free to adapt the line

```
require_once PROJECT_DIR.'/libs/smarty/Smarty.class.php';
```

in the [config.php](#) to point to the right file.

3.3 How do I add support for another language?

- copy one of the existing folders in data (like 'en' for instance)
- translate all the files inside it
 - don't touch the tags in the xml files
 - don't change the filenames
 - don't change the order of the skills in skills.yaml
- in [config.php](#), add the short name of your language (which has to be the same name as the directory) to the `$allowed_languages` array.

```
// Example:  
$allowed_languages = array('en','de','fr'); // old  
$allowed_languages = array('en','de','fr','es'); // new
```

- now you need to display the new language to the user so he can chose it. In *views/templates/index.tpl* add another html block like the one between line 20 and 23. Example, for spanish:

```
<form action="index.php" method="post" style="float: left;">  
  <input type="hidden" name="lang" value="es"></input>  
  <button type="submit"></button>  
</form>
```

Chapter 4

License

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it,

either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it,

under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot

distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into

proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.

Chapter 5

Class Index

5.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

mySmarty (Set up the template engine Smarty)	21
Races (Parse correct races.xml file)	24
Skills (Parse the skills.yaml file)	26
TeamLoader (Parse a roster)	27
TeamSaver (Return a roster as xml)	30
Translation (Builds a dictionary and translates a roster with it)	33
UserInterface (Returns all the text that composes the UI)	36

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

config.php (Configuration file)	37
faq.doxy	39
helper.php (Miscellaneous helper funtions)	40
index.php (Main file, or "controller")	41
license.doxy	42
Races.php (Implement Races class)	43
Skills.php (Implement Skills class)	44
snore.doxy	45
TeamLoader.php (Implement TeamLoader class)	46
TeamSaver.php (Implement TeamSaver class)	47
Translation.php (Implement Translation class)	48
UserInterface.php (Implement UserInterface class)	49
workflow.doxy	50

Chapter 7

Class Documentation

7.1 mySmarty Class Reference

Set up the template engine Smarty.

Public Member Functions

- [__construct \(\)](#)
Use the constructor to set all desired variables.

Public Attributes

- [\\$template_dir](#)
This is the name of the default template directory.
- [\\$compile_dir](#)
Directory in which Smarty stores the compiled templates.
- [\\$config_dir](#)
Directory used to store template-configuration files.
- [\\$cache_dir](#)
Directory in which Smarty stores the cache.
- [\\$debugging](#)
Toggles an useful debugging console.
- [\\$caching](#)
Toggles caching of rendered templates.
- [\\$compile_check](#)
Check to see if a template has to be recompiled.

- [\\$cache_lifetime](#)

Time until cache expires in seconds.

7.1.1 Detailed Description

A flexible way to set up Smarty is to extend the base class and initialize the Smarty environment. So instead of repeatedly setting directory paths, assigning the same vars, etc., it is done in one place.

Described in <http://www.smarty.net/manual/en/installing.smarty.extended.php>

SNORE uses own smarty-plugins which are stored in the directory which gets appended to the `plugins_dir` array of the base class. Described in <http://www.smarty.net/manual/en/variable.plugins.dir.php>

7.1.2 Constructor & Destructor Documentation

7.1.2.1 `mySmarty.__construct()`

7.1.3 Member Data Documentation

7.1.3.1 `mySmarty.$template_dir`

Described in <http://www.smarty.net/manual/en/api.variables.php#variable.template.dir>

7.1.3.2 `mySmarty.$compile_dir`

Directory must be writable by the web server! Described in <http://www.smarty.net/manual/en/variable.compile.dir.php>

7.1.3.3 `mySmarty.$config_dir`

SNORE does not use any of those for now, but they may be useful. Described in <http://www.smarty.net/manual/en/variable.config.dir.php>

7.1.3.4 `mySmarty.$cache_dir`

Must be writable by the web server! Described in <http://www.smarty.net/manual/en/variable.cache.dir.php>

7.1.3.5 `mySmarty.$debugging`

I really recommend to set this to TRUE when working on the templates, especially during refactoring. Described in <http://www.smarty.net/manual/en/variable.debugging.php>

7.1.3.6 mySmarty.\$caching

It is wise to disable caching during the development phase, but remember to enable it (set to 1) for production, because it REALLY increases the speed of the application. Described in <http://www.smarty.net/manual/en/variable.caching.php>

7.1.3.7 mySmarty.\$compile_check

Slight overhead, can be set to FALSE if in production mode, but don't be surprised if you don't see the changes you have made in the templates after their first compilation. You have to delete the cache AND the compiled templates then. Described in <http://www.smarty.net/manual/en/variable.compile.check.php>

7.1.3.8 mySmarty.\$cache_lifetime

The cache never expires when cache_lifetime is set to -1. This is the right thing to do in SNORE, because of the static nature of it's content. Again, remember to delete the cache and the compiled templates if you tinkered with the templates in production. Described in <http://www.smarty.net/manual/en/variable.cache.lifetime.php>

7.2 Races Class Reference

Parse correct races.xml file.

Static Public Member Functions

- static `get` (\$race_id, \$lang=`LANG`)
return everything there is to know about a specific race
- static `list_all` (\$lang=`LANG`)
list all races that exist
- static `list_positions` (\$race_id, \$lang=`LANG`)
list the positionals of a specific race

7.2.1 Detailed Description

The member functions will all return results in the language given as a parameter.

7.2.2 Member Function Documentation

7.2.2.1 static Races.get (\$ race_id, \$ lang = LANG) [static]

Parameters:

- ← *\$race_id* the position in which the race appears in the file
- ← *\$lang* language so the application knows which race file to parse
- *\$data* an array and simplexml-object crossbreed

This really is the fastest way to get all information we want, because simplexml_load_file maps the whole file to a sort of array.

Referenced by show_roster().

7.2.2.2 static Races.list_all (\$ lang = LANG) [static]

Parameters:

- ← *\$lang* the current language
- *\$list* a numbered array

The parsing is done with the XmlReader class, which is perhaps not as easy to use as the simplexml one, but it is more memory-efficient to parse the file by hand instead of mapping everything to a huge array, only to traverse it and grab the race-names.

Referenced by Translation.__construct(), and show_index().

7.2.2.3 static Races.list_positions (\$ *race_id*, \$ *lang* = LANG) [static]

Parameters:

- ← *\$race_id* the position in which the race appears in the file
- ← *\$lang* the current language
- *\$list* numbered array listing the names of all the positions

Referenced by Translation.__construct(), and Translation.list_positions().

7.3 Skills Class Reference

Parse the skills.yaml file.

Static Public Member Functions

- static `all_nested` (\$lang=LANG)
returns all skills grouped by categories
- static `all_flat` (\$lang=LANG)
returns all skills

7.3.1 Detailed Description

This class is used every time we need a list of all skills that exist in the game.

7.3.2 Member Function Documentation

7.3.2.1 static Skills.all_nested (\$ lang = LANG) [static]

Parameters:

- ← *\$lang* desired language of the results
- *\$list* two-dimensional array of skills grouped by categories

This is used when the roster is generated. Ultimately, the template `_javascript.tpl` produces javascript-arrays, which are very important to the skill-handling in the javascript.

Referenced by `show_roster()`.

7.3.2.2 static Skills.all_flat (\$ lang = LANG) [static]

Parameters:

- ← *\$lang* desired language of the results
- *\$list* simple numbered array of all skills

Used twice (once for every language) everytime a translation is built. The arrays are later combined.

Referenced by `Translation.__construct()`.

7.4 TeamLoader Class Reference

Parse a roster.

Public Member Functions

- [__destruct](#) ()
Automatic closing of the XmlReader.

Static Public Member Functions

- static [load](#) (\$file, \$lang=[LANG](#))
loads the roster from an xml file.

Private Member Functions

- [__construct](#) (\$file)
prepare for parsing
- [to_array](#) ()
parse the xml and transform it into an array
- [handle_text](#) ()
handle text when parsing xml
- [attributes](#) ()
handle attributes when parsing xml

Private Attributes

- [\\$XmlReader](#)
- [\\$result](#)
- [\\$temp_player](#)
- [\\$last_opened_element_name](#)
- [\\$parsing_inside_player_element](#)

7.4.1 Detailed Description

Given a roster stored in a xml file, loading it with [TeamLoader:load](#) will parse it into an array, so it can be used by the application.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 TeamLoader.__construct (\$file) [private]

Parameters:

← *\$file* the path of the roster

Gets called by the static method load(\$file).

7.4.2.2 TeamLoader.__destruct ()

7.4.3 Member Function Documentation

7.4.3.1 static TeamLoader.load (\$file, \$lang = LANG) [static]

Parameters:

← *\$file* the path of the roster

← *\$lang* (optional) language the roster is translated into

→ *\$roster* an array in the language in \$lang

usage, for instance for handling an uploaded roster: \$roster = Teamloader.load(\$_FILES['userfile']['tmp_name']); The result (which was translated into the language given in LANG) can directly be passed to the templates.

References Translation.for_loading().

7.4.3.2 TeamLoader.to_array () [private]

Straight-forward xml parsing with XmlReader. Reacts to start and end tags. It knows certain tags (team, player, inj) where it has to look for attributes, which are then handled by the [attributes\(\)](#) method. Text is delegated to [handle_text\(\)](#).

When parsing a player, content is buffered into the \$temp_player array. Then, after we have parsed the player, we know his number and can attach him to the team array with his number as index.

References attributes(), and handle_text().

7.4.3.3 TeamLoader.handle_text () [private]

Referenced by to_array().

7.4.3.4 TeamLoader.attributes () [private]

Does not get called for every tag with attributes, but only for tags that are known to have attributes in [to_array\(\)](#).

Referenced by to_array().

7.4.4 Member Data Documentation

7.4.4.1 TeamLoader.\$XmlReader [private]

7.4.4.2 TeamLoader.\$result [private]

7.4.4.3 TeamLoader.\$temp_player [private]

7.4.4.4 TeamLoader.\$last_opened_element_name [private]

7.4.4.5 TeamLoader.\$parsing_inside_player_element [private]

7.5 TeamSaver Class Reference

Return a roster as xml.

Static Public Member Functions

- static [save](#) (\$post_data, \$lang=[LANG](#))
given data, returns xml

Private Member Functions

- [__construct](#) ()
setup
- [prepare](#) (&\$data)
prepares the data for translation
- [write_team](#) ()
builds the xml of the roster
- [write_team_attributes](#) ()
Write the attributes of the team element.
- [write_players](#) ()
build the players element
- [write_player](#) (\$index)
write a player element
- [write_player_attributes](#) (\$index)
write the attributes of the player element
- [write_player_skills](#) (\$index)
write the skills of a player
- [write_player_injuries](#) (\$index)
write the injuries of a player
- [correct_fullEndElement_bug](#) ()
append a newline after a the /inj tag

Private Attributes

- [\\$XmlWriter](#)
- [\\$data](#)
- [\\$result](#)

7.5.1 Constructor & Destructor Documentation

7.5.1.1 TeamSaver.__construct () [private]

Create XmlWriter instance, and set some options, like indentation. Prepare needed variables.

7.5.2 Member Function Documentation

7.5.2.1 static TeamSaver.save (\$post_data, \$lang = LANG) [static]

Parameters:

- ← *\$post_data* data which comes from the html-form
- ← *\$lang* \$language of the data
- *\$result* xml string

References \$result, and Translation.for_saving().

7.5.2.2 TeamSaver.prepare (&\$data) [private]

Parameters:

- ← *\$data* the data

In the user interface, list content (skills, injuries) is stored in a string (example: block,dodge or M,N,-AG). This has to be transformed into an array.

Additionally, some temporary variables can be removed.

References \$data.

7.5.2.3 TeamSaver.write_team () [private]

Starts and ends the document. Inbetween, some parts have been refactored out into the other write_* methods.

References write_players(), and write_team_attributes().

7.5.2.4 TeamSaver.write_team_attributes () [private]

Referenced by write_team().

7.5.2.5 TeamSaver.write_players () [private]

Delegates the grunt-work to [write_player\(\)](#).

References write_player().

Referenced by write_team().

7.5.2.6 TeamSaver.write_player (\$ *index*) [private]

References write_player_attributes(), write_player_injuries(), and write_player_skills().

Referenced by write_players().

7.5.2.7 TeamSaver.write_player_attributes (\$ *index*) [private]

Referenced by write_player().

7.5.2.8 TeamSaver.write_player_skills (\$ *index*) [private]

Referenced by write_player().

7.5.2.9 TeamSaver.write_player_injuries (\$ *index*) [private]

Referenced by write_player().

7.5.2.10 TeamSaver.correct_fullEndElement_bug () [private]

XmlWriter->fullEndElement doesn't append a newline after the tag, so we look for the tag using regular expressions.

7.5.3 Member Data Documentation**7.5.3.1 TeamSaver.\$XmlWriter [private]****7.5.3.2 TeamSaver.\$data [private]**

Referenced by prepare().

7.5.3.3 TeamSaver.\$result [private]

Referenced by save().

7.6 Translation Class Reference

Builds a dictionary and translates a roster with it.

Static Public Member Functions

- static [test](#) (\$lang_from, \$lang_to, \$race_id)
Prints a dictionary for testing purposes.
- static [for_saving](#) (\$data, \$lang_from, \$lang_to='en')
Translates data that is going to be saved in a roster.
- static [for_loading](#) (\$data, \$lang_to, \$lang_from='en')
Translates data that is going to be saved in a roster.

Private Member Functions

- [__construct](#) (\$lang_from, \$lang_to, \$race_id)
Builds the dictionary used for the translations.
- [list_positions](#) (\$race_id, \$lang)
lists the positions of a certain race.

Private Attributes

- [\\$translation](#)
the dictionary

7.6.1 Detailed Description

There are two different static methods for translation, [Translation.for_saving\(\)](#) and [Translation.for_loading\(\)](#), because the data has a different form in each case.

The dictionary is the same for both, though.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 [Translation.__construct](#) (\$ lang_from, \$ lang_to, \$ race_id) [private]

Parameters:

- ← *\$lang_from* the 'keys' in the dictionary
- ← *\$lang_to* the 'values' in the dictionary
- ← *\$race_id* the id of a race

Why do we need a `$race_id`? Because we need to translate the players' positions (like Lineman, Thrower) too, and it would be too time consuming to look up them up for each race.

Once the translation is built, it is stored in the class variable `$translation`.

References `Skills.all_flat()`, `Races.list_all()`, `Races.list_positions()`, and `ManagerInterface.stats()`.

7.6.3 Member Function Documentation

7.6.3.1 static `Translation.test ($ lang_from, $ lang_to, $ race_id)` [static]

Parameters:

- ← *\$lang_from* the 'keys' in the dictionary
- ← *\$lang_to* the 'values' in the dictionary
- ← *\$race_id* the id of a race

It has the same parameters as `__construct()`.

References `$translation`.

7.6.3.2 static `Translation.for_saving ($ data, $ lang_from, $ lang_to = 'en')` [static]

Parameters:

- ← *\$data* the data
- ← *\$lang_from* the language of the data
- ← *\$lang_to* the language the data is going to be translated into
- *\$data* the data after translation

In the typical scenario for SNORE, the data comes in the language of the user interface (for example french for a french user). Because all rosters are stored in english, the data has to be translated to english, which is thus the default for `$lang_to`.

Referenced by `TeamSaver.save()`.

7.6.3.3 static `Translation.for_loading ($ data, $ lang_to, $ lang_from = 'en')` [static]

Parameters:

- ← *\$data* the data
- ← *\$lang_to* the language the data is going to be translated into
- ← *\$lang_to* the language of the data
- *\$data* the data after translation

In SNORE, the rosters are stored in english. Hence the default for `$lang_from` is english. When loaded, the roster is translated into `$lang_to`.

References `list_positions()`.

Referenced by `TeamLoader.load()`.

7.6.3.4 Translation.list_positions (\$ race_id, \$ lang) [private]

Parameters:

- ← *\$race_id* a race
- ← *\$lang* the language
- *\$positions* positions of race \$race_id in language \$lang

References Races.list_positions().

Referenced by for_loading().

7.6.4 Member Data Documentation

7.6.4.1 Translation.\$translation [private]

Referenced by test().

7.7 UserInterface Class Reference

Returns all the text that composes the UI.

Static Public Member Functions

- static `get` (\$page, \$lang=`LANG`)
Parse correct yaml file.
- static `stats` (\$lang=`LANG`)
List stats and injury abbreviations.

7.7.1 Detailed Description

The `UserInterface` class is mostly a wrapper around the Spyc YAML library.

7.7.2 Member Function Documentation

7.7.2.1 static `UserInterface.get` (\$ *page*, \$ *lang* = `LANG`) [static]

Parameters:

- ← *\$page* requested page, i.e. roster or index
- ← *\$lang* desired language of the results
- *\$data* associative array

Referenced by `show_index()`, and `show_roster()`.

7.7.2.2 static `UserInterface.stats` (\$ *lang* = `LANG`) [static]

Parameters:

- ← *\$lang* desired language of the results
- *\$stats* numbered array where values are the abbreviations

Referenced by `Translation.__construct()`.

Chapter 8

File Documentation

8.1 config.php File Reference

configuration file.

Classes

- class `mySmarty`

Set up the template engine Smarty.

Enumerations

- enum `PROJECT_DIR`

Base directory of the application.

- enum `LANG`

The language the user has set.

Variables

- `$allowed_languages = array('en','de','fr')`

Languages that are safe to use for SNORE.

8.1.1 Detailed Description

Define global constants used in the whole application, and set up the template engine too.

8.1.2 Enumeration Type Documentation

8.1.2.1 `enum LANG`

Important global variable, that controls which localisation-files are parsed, and thus the language of the user-interface.

8.1.2.2 `enum PROJECT_DIR`

Used for includes, like in [mySmarty](#).

8.1.3 Variable Documentation

8.1.3.1 `$allowed_languages = array('en','de','fr')`

This is to control and restrict the possible directories in which the application may want to open files. Remember: Never trust user input!

8.2 `faq.doxy` File Reference

8.3 helper.php File Reference

Miscellaneous helper funtions.

Functions

- `check_lang` (\$allowed)
Returns the language the user set.
- `__autoload` (\$class_name)
autoload magic as seen in <http://de2.php.net/autoload>
- `show_index` ()
shortcut for calls to the template engine
- `show_roster` (\$race_id, \$loaded_team=NULL)
shortcut for calls to the template engine
- `insert_getError_race` (\$params)
template function to display an error message
- `insert_getError_upload` (\$params)
template function to display an error message

8.3.1 Detailed Description

Functions that are used on several occasions, but don't fit into any particular class or group are declared here.

8.3.2 Function Documentation

8.3.2.1 `__autoload` (\$ *class_name*)

8.3.2.2 `check_lang` (\$ *allowed*)

Performs a check to see if the language is allowed in the config.

8.3.2.3 `insert_getError_race` (\$ *params*)

8.3.2.4 `insert_getError_upload` (\$ *params*)

8.3.2.5 `show_index` ()

References `ManagerInterface.get()`, and `Races.list_all()`.

8.3.2.6 `show_roster` (\$ *race_id*, \$ *loaded_team* = NULL)

References `Skills.all_nested()`, `Races.get()`, and `ManagerInterface.get()`.

8.4 index.php File Reference

Main file, or "controller".

8.4.1 Detailed Description

See page [Controller](#) for more details.

8.5 `license.doxy` File Reference

8.6 Races.php File Reference

implement [Races](#) class

Classes

- class [Races](#)

Parse correct races.xml file.

8.6.1 Detailed Description

8.7 Skills.php File Reference

implement [Skills](#) class

Classes

- class [Skills](#)

Parse the skills.yaml file.

8.7.1 Detailed Description

8.8 snore.doxy File Reference

8.9 TeamLoader.php File Reference

implement [TeamLoader](#) class

Classes

- class [TeamLoader](#)
Parse a roster.

8.9.1 Detailed Description

8.10 TeamSaver.php File Reference

implement [TeamSaver](#) class

Classes

- class [TeamSaver](#)
Return a roster as xml.

8.10.1 Detailed Description

8.11 Translation.php File Reference

implement [Translation](#) class

Classes

- class [Translation](#)
Builds a dictionary and translates a roster with it.

8.11.1 Detailed Description

8.12 `UserInterface.php` File Reference

implement `UserInterface` class

Classes

- class `UserInterface`

Returns all the text that composes the UI.

8.12.1 Detailed Description

8.13 workflow.doxy File Reference

Index

- `$XmlReader`
 - `TeamLoader`, [29](#)
- `$XmlWriter`
 - `TeamSaver`, [32](#)
- `$allowed_languages`
 - `config.php`, [38](#)
- `$cache_dir`
 - `mySmarty`, [22](#)
- `$cache_lifetime`
 - `mySmarty`, [23](#)
- `$caching`
 - `mySmarty`, [22](#)
- `$compile_check`
 - `mySmarty`, [23](#)
- `$compile_dir`
 - `mySmarty`, [22](#)
- `$config_dir`
 - `mySmarty`, [22](#)
- `$data`
 - `TeamSaver`, [32](#)
- `$debugging`
 - `mySmarty`, [22](#)
- `$last_opened_element_name`
 - `TeamLoader`, [29](#)
- `$parsing_inside_player_element`
 - `TeamLoader`, [29](#)
- `$result`
 - `TeamLoader`, [29](#)
 - `TeamSaver`, [32](#)
- `$temp_player`
 - `TeamLoader`, [29](#)
- `$template_dir`
 - `mySmarty`, [22](#)
- `$translation`
 - `Translation`, [35](#)
- `__autoload`
 - `helper.php`, [40](#)
- `__construct`
 - `mySmarty`, [22](#)
 - `TeamLoader`, [28](#)
 - `TeamSaver`, [31](#)
 - `Translation`, [33](#)
- `__destruct`
 - `TeamLoader`, [28](#)
- `all_flat`
 - `Skills`, [26](#)
- `all_nested`
 - `Skills`, [26](#)
- `attributes`
 - `TeamLoader`, [28](#)
- `check_lang`
 - `helper.php`, [40](#)
- `config.php`, [37](#)
 - `$allowed_languages`, [38](#)
 - `LANG`, [38](#)
 - `PROJECT_DIR`, [38](#)
- `correct_fullEndElement_bug`
 - `TeamSaver`, [32](#)
- `faq.doxy`, [39](#)
- `for_loading`
 - `Translation`, [34](#)
- `for_saving`
 - `Translation`, [34](#)
- `get`
 - `Races`, [24](#)
 - `ManagerInterface`, [36](#)
- `handle_text`
 - `TeamLoader`, [28](#)
- `helper.php`, [40](#)
 - `__autoload`, [40](#)
 - `check_lang`, [40](#)
 - `insert_getError_race`, [40](#)
 - `insert_getError_upload`, [40](#)
 - `show_index`, [40](#)
 - `show_roster`, [40](#)
- `index.php`, [41](#)
- `insert_getError_race`
 - `helper.php`, [40](#)
- `insert_getError_upload`
 - `helper.php`, [40](#)
- `LANG`
 - `config.php`, [38](#)
- `license.doxy`, [42](#)
- `list_all`

- Races, 24
- list_positions
 - Races, 24
 - Translation, 34
- load
 - TeamLoader, 28
- mySmarty, 21
 - \$cache_dir, 22
 - \$cache_lifetime, 23
 - \$caching, 22
 - \$compile_check, 23
 - \$compile_dir, 22
 - \$config_dir, 22
 - \$debugging, 22
 - \$template_dir, 22
 - __construct, 22
- prepare
 - TeamSaver, 31
- PROJECT_DIR
 - config.php, 38
- Races, 24
 - get, 24
 - list_all, 24
 - list_positions, 24
- Races.php, 43
- save
 - TeamSaver, 31
- show_index
 - helper.php, 40
- show_roster
 - helper.php, 40
- Skills, 26
 - all_flat, 26
 - all_nested, 26
- Skills.php, 44
- snore.doxy, 45
- stats
 - UserInterface, 36
- TeamLoader, 27
 - \$XmlReader, 29
 - \$last_opened_element_name, 29
 - \$parsing_inside_player_element, 29
 - \$result, 29
 - \$temp_player, 29
 - __construct, 28
 - __destruct, 28
 - attributes, 28
 - handle_text, 28
 - load, 28
 - to_array, 28
- TeamLoader.php, 46
- TeamSaver, 30
 - \$XmlWriter, 32
 - \$data, 32
 - \$result, 32
 - __construct, 31
 - correct_fullEndElement_bug, 32
 - prepare, 31
 - save, 31
 - write_player, 31
 - write_player_attributes, 32
 - write_player_injuries, 32
 - write_player_skills, 32
 - write_players, 31
 - write_team, 31
 - write_team_attributes, 31
- TeamSaver.php, 47
- test
 - Translation, 34
- to_array
 - TeamLoader, 28
- Translation, 33
 - \$translation, 35
 - __construct, 33
 - for_loading, 34
 - for_saving, 34
 - list_positions, 34
 - test, 34
- Translation.php, 48
- UserInterface, 36
 - get, 36
 - stats, 36
- UserInterface.php, 49
- workflow.doxy, 50
- write_player
 - TeamSaver, 31
- write_player_attributes
 - TeamSaver, 32
- write_player_injuries
 - TeamSaver, 32
- write_player_skills
 - TeamSaver, 32
- write_players
 - TeamSaver, 31
- write_team
 - TeamSaver, 31
- write_team_attributes
 - TeamSaver, 31