

# Solar Imperium 2.6.x Mini developer guide

written by Yanick Bourbeau in approximative english :)

## First things first, a note from myself:

Solar Imperium was a one developer project for a long periods now, I was helped by some awesome persons during the last 2 years in different fields like writing documentation, giving feedback and finding bugs, but nobody really developed pieces of code for the project. When a project start to contains several thousands lines of code, I have to sadly admit thats only one person is not enough :) In this view, I want to make Solar Imperium a nice LGPL'ed game project managed by several developers instead of being alone on this one. Please bear in mind that is my first attempt at creating a multi-developers project and if you know better ways of working together, feel free to give me some feedback.

*Thank you and I hope you will have fun developing Solar Imperium*

## What is Solar Imperium?

Solar Imperium is a web game based on a classic game called “Solar Realms Elite”, I played this game when I was young (around 13) and liked it alot. Back on theses days, the game was played on several Bulletin Board Systems, a normal game had ~20 players. Solar Imperium implement all the basic aspects of SRE plus very new concepts. Now the game system can handle as many games as you want and each game can host ~40 players before being overcrowded. Two kind of games can be created, the normal invasion game and the research race. In the normal invasion game, after a specified period of days, the biggest empire won. In research race, the first empire to complete last level of research won the game.

## **Technologies behind Solar Imperium**

Solar Imperium is a project developed for 3 years now (part time), the initial release was black on white cheap PHP scripts without images or fancy scripts. The game have evolved alot since then. Solar Imperium 2.6 is developed using PHP5 (the code work with PHP4 too). Dynamic data such as players information or empire events are stored in a MySQL database. I use SMARTY as template engine, AFLAX for the resizable starmap and JSGRAPHICS for all dynamic images.

### **PHP5 (<http://www.php.net>)**

I use PHP5 as primary language for developing this game, the game is also comaptible with PHP4 code because I dont really use newest features found in PHP5. On some PHP4 systems the game may generate extra warnings but no errors. In fact, the main game server: [www.solarimperium.com](http://www.solarimperium.com) run on PHP4.

### **MySQL (<http://www.mysql.org>)**

I use SQL to do all the database stuff, the only supported SQL server is MySQL for now, but I don't see any problem to use MSSQL or Postgres for example. I use the ADODB library to communicate with MySQL. Please note I use MySQL INNODB mode because the game need some transactions support. The default MySAM mode does not give any transactions support. I tested the game on MySQL 4.

### **SMARTY (<http://smarty.php.net>)**

Smarty is the templating system, its a very advanced library thats allow me to split calculation code and display code into 2 different files. You have to keep all the game calculation code in the .php file and all the layout/formatting code in the template files (html). If you dont know about Smarty, PLEASE GO READ THE DOCUMENTATION, its a requirement if you want to deal with templates.

### **AFLAX (<http://www.aflax.org>)**

This library allow you to build a FLASH file using javascript, you can build everything in javascript and then display it into a FLASH component. This is only used for the starmap code.

### **Canvas / JSGRAPHICS ([http://www.walterzorn.com/jsgraphics/jsgraphics\\_e.htm](http://www.walterzorn.com/jsgraphics/jsgraphics_e.htm))**

I use this javascript library to create fake images using javascript, all the rendering is done on the client side and the server does not have to generate full bitmaps on every refresh. A good example of the javascript library in action is PERFORMANCE tab.

### **Other CSS / Javascript / AJAX**

I have used several third party libraries and made some original content to give me specific functions like dynamic content hovering tooltips.

## **Setting up your own development environment**

First you need to install APACHE2 + MySQL + PHP5, this documentation will not cover these basic steps because it's very easy to find good information about how to install these softwares on Internet. Once you have a basic system working, you need to extract the content of the game zip file in a directory on your web server, like : solar26. To download the latest package for the game, visit <http://sourceforge.net/projects/solar/>.

Then just call install.php from a web browser (not from CLI/console) and follow the installer step by step. If everything goes nicely, you now have a basic system you can tweak and modify. The first registered player becomes the administrator. Note that PHPMysqlAdmin may be a nice tool to help you with SQL database management and querying.

## The project filesystem layout

Directories:

.	Contains several .php files, these scripts are used for basic player management and all non-ingame display of information (scoreboard, starmap)
cronjob	Contains few .php files, these scripts are used to update the state of all the games on the server. For example, these scripts give back new turns, calculate invasion casualties, etc...
css	Contains few .css files, used by templates
game	Contains several .php files, these scripts are used for the game itself, displaying various menus and items
images	Contains images used by the players system and by the game, also contains generated logos
include	Contains all the repetitive code used by the game and by the players system. Also include some third parties like my own error handler, ADODB, Smarty, etc...
locale	Contains gettext-compatible locales
scripts	Contains javascript part of the game / players system
templates	Contains alot of files, all these files are templates, some are used for events, some for pages and menus
templates_c	Compiled templates, used internally by Smarty

## The project database layout

System tables:

system_tb_chat_logs	Contains all the messages exchanged in the chat room, these entries are periodically deleted (oldest entries)
system_tb_chat_sessions	Contains currently opened chat sessions (online)
system_tb_forum	Contains all the forum entries, I use the same table for both main topics and replies
system_tb_games	List available games on the server
system_tb_hall_of_fame	List past winners on current server
system_tb_history	When a game finish, the last status of each empires is stored in this table, the player can consult after game statistics (turns, networth, etc)
system_tb_messages	Contains messages sent by administrator to the players
system_tb_players	List of all players, including email and password (MD5)
system_tb_sessions	Connected players are stored here, not the same as chat sessions
system_tb_stats	Used to monitor how much players signed up and/or connected each day.
system_tb_warrant	Contains warrants issued (Kick and ban from admin)

Game tables, created for each game added, in the same database:

gameX_tb_army	Army information of a specific empire
gameX_tb_armyconvoy	Moving army convoys, both attacking and defending convoys
gameX_tb_bond	Bonds taken by empires
gameX_tb_coalition	Contains coalitions information
gameX_tb_coordinator	This is a old table used for misc. Stuff and game variables
gameX_tb_empire	Contains information about a empire
gameX_tb_event	Contains events, alot of them generally, oldest events are periodically removeds
gameX_tb_hall_of_fame	Contains previous games hall of fame (top 3 players for each game)
gameX_tb_invasion	When a invasion happen, the resulting information is stored in this table
gameX_tb_loan	Loans taken by empires
gameX_tb_market	Dynamic galactic market information stored in this table
gameX_tb_member	List all coalition members, used with gameX_tb_coalition
gameX_tb_pirate	Contains a list of pirates and they army information, automatically resetted on game reset.

gameX_tb_planets	Empire planets
gameX_tb_production	Empire planets production
gameX_tb_research_done	List of researchs done by empire
gameX_tb_research_tech	List of all reseach techs (english only, may need to find a way to allow multiple definitions using gettext?)
gameX_tb_session	List of all players playing as a empire in this game
gameX_tb_shoutbox	Coalition shoutbox, cheap and very basic system
gameX_tb_stats	Each turns statistics (resources, units, various information) stored in this table, used to draw performance graphics
gameX_tb_supply	Supply/units production settings for a empire
gameX_tb_trace	Spy on each operations made by the empire, allow the admin to catch cheaters, very useful :)
gameX_tb_tradeconvoy	Contains trade convoys
gameX_tb_treaty	List of all currently active treaties and pending offers

## Development guidelines to respect

The code at this stage is far from being perfect but I tried to keep the code clean and to re-use common objects when needed and to follow these few simple rules:

- All formatting code must be on SMARTY/Template side, for example, instead of using:

```
$TPL->assign("money",number_format($money));
```

you might use:

```
$TPL->assign("money",$money);
```

and use this tag in the template:

```
{ $money|number_format }.
```

If what I am writing sounds like Klingon, go read the smarty documentation, it's a must.

- Do use Empire, Army, Planets and other game objects when you want to change/edit empire information, do not use SQL directly. This guideline may be ignored in the "cronjob" part of the script.
- Use ADODB, not mysql native functions, in the game, a \$DB variable is usually created for this purpose.
- Don't call \$DB->CompleteTrans() except if you ARE KNOWING what you do.
- Please comment your code, I know the code is not well commented but we have to start somewhere :)
- If you want to give a specific information to an empire, create a new event type and use EventCreator object to help you out.
- If you create game constants, please store them in rules files



## What are the much needed features to implement?

You can modify/tweak/add whatever features you want, but if you are searching what to improve in the game, I have a list of well known missing features:

- 1) No chemical/psyonic attack yet, the old SRE had some, usually theses attacks target the civilians but not the planets, I guess theses attacks can be bought from black market like the nuclear devices.

From original SRE documentation:

### *Chemical Warfare*

*Poisonous chemicals have been banned in our galaxy, but exist in many of the barbaric galaxies in the universe. Occasionally these chemicals are brought to our galaxy for causing human misery. Due to the highly uncivilized nature of these weapons, the Galactic Coordinator is tightly enforcing the ban. If you found even possessing chemicals that can be used for chemical warfare, you will be instantly annihilated by The Galaxy's army. Don't even think of using these.*

### *Psionic Bombs*

*While normal attacks are targeted towards normal physical destruction, these bombs target the mind. The population of your opponent will be thrown into anarchy as mass confusion and havoc reigns. In addition, your opponent's soldiers will be totally demoralized.*

- 2) A simplified documentation at [www.galaxypedia.com](http://www.galaxypedia.com), the current documentation is based on the old 1.6.5 code, very outdated.
- 3) A tutorial mode for new comers in the form of floating windows with dismiss button, giving hints and even what values to use. The tutorial mode can be disabled from the players profile menu or directly from any floating windows.
- 4) Complete gettext translation to french and other foreign languages
- 5) Find a way to make research tech descriptions multi-language enabled
- 6) Make the HTML code W3C compliant
- 7) Simplify the HTML interface, maybe rounded borders (again)
- 8) Add more information in the star map (Draw carriers, cruisers icons directly in the starmap)
- 9) Create some AI/dummy empires to create "quests" or to practice-play.

Thats the end, I hope you now know the project a little bit better, I know that this mini-documentation cannot answer all your questions, its why you can contact me at [ybourbeau@mrgtech.ca](mailto:ybourbeau@mrgtech.ca)

Thank you for reading  
Yanick Bourbeau  
ybourbeau@mrgtech.ca