

Project: Little Goblin, a browser game framework and reference implementation

Version 2011-01-02 // 0.4

This document is intended for students of web design and programming courses and their administrative staff / teachers who are looking for an interesting project which offers the chance to learn more about browser game design and implementation.

About browser games in general:

Browser games have a long tradition, but only in the last few years have they reached a higher level of publicity and increased commercial success. The first games relied on banner ads for revenue and often struggled to support the web hosting costs. Today, Zynga is a giant among game producers, with a reported revenue of \$600 million and 1300 employees.¹ But the market is still open for small independent productions by small teams², although the bar for successful game titles continues to rise.

Browser games are suitable for small teams and can be implemented with a reasonable amount of work. The internet offers a good feedback mechanism to get user input for improvement and also makes it easy to roll-out updates and perform A/B-testing of new features.

Little Goblin

Little Goblin is an open source browser game framework which is currently under development. Its goal is to provide developers and designers with the means to create web games quickly and without having to “invent the wheel” again and again. This means the framework implements user management, security aspects and often encountered game structures (for example, a mechanism to schedule recurring game events).

The reference implementation is aimed at providing a demonstration and test platform for the individual features. It is currently being designed as a fantasy game. The framework itself, once completed, can be used for a multitude of content and game types.

Little Goblin as a project for fledgling web designers and programmers

The code is under development, yet what is there is already usable. There is still much work to be done, especially in the area of web design and internationalization. The scope of this project is large, so it is to be expected that some modules may take a long time to finish. Thus, project participants will need to know that their work will be a valuable and measurable contribution, but they are not expected to complete large parts of the project all by themselves (in the time frame of a study project).³

1 <https://secure.wikimedia.org/wikipedia/en/wiki/Zynga>

2 The most successful independent game of 2010 is Minecraft, a Java applet / application with 700'000 sold copies. Of course, many games are not cost effective.

3 Meaning: it would be great to have more people working on this project, and each contribution can be as large or small as the team member's time budget and skills allow. Creating open source software is about doing something fun and worthwhile, scratching an itch, learning new stuff and giving back to society (and sometimes making money, too). It is not about pushing rocks up a slope.

Game Design

As the framework grows and offers more features, the reference implementation is subject to change so those new options can be explored by players and developers. The game design aspect is currently covered by the project administrator, but is of course open for suggestions by other team members.

Web design layout / CSS

At the moment, the layout is generally working on Firefox, but will need to be overhauled completely. Most elements of the dynamically generated pages carry css class selectors, so changes to the layout can mostly be made by editing the CSS. The project does not yet use a general CSS framework (like 960.gs).

Web interface programming

Some pages employ AJAX, but this has not been tested on other browsers or optimized for barrier free web access. Depending on the skills of the interface programmer, it could be possible to add more features like a dynamic map of the game world or interactive item handling (for example, drag and drop of a player character's items from his inventory into a shop).

Statistics and data mining

Strategy online games need statistics to display a multitude of rankings. When competing with 10000 other players, one of the key questions is: how good am I (in relation to other players)? Graphical display of game data regarding current status or changes in status over time is also highly desirable. Public user pages are also a possible feature. Those pages need to offer good usability and an efficient layout for information presentation.

Internationalization

The project uses properties files with English messageId / text pairs and can be translated into other languages. Later on, the system will transition to a database based backend so users and developers can edit the translations via web front end. Import/export features for professional translation tools are probably a good idea, but not yet planned.

The first step in creating a multi-language version of Little Goblin would be to expand the language data by French, German and perhaps Spanish. Depending on volunteers and / or commercial success, other languages may follow until the framework covers the EU (or beyond).

Documentation

There is currently not much in the way of documentation – but as the project grows we need both end user and developer documentation, with screen shots and detailed tutorials.

Backend code

The server side code is mostly written in Groovy by the project administrator. Sub-systems may be developed in other languages that use HTTP for communication with the main server process.

Interface with online services

- Facebook connect
 - login via Facebook account
 - send game status events to Facebook
- Connection to online payment providers (PayPal, other micropayment providers) – the framework allows the users to buy credits / coins for use in in-game transactions or to unlock special game features.
- Twitter

Pro / chances:

- A public open source project which can be used for reference (unlike proprietary custom projects where all assets belong to a company and often are not available for public reference).
- A professional project environment (Sourceforge offers mailing list, bug and feature tracker, code repository, mirroring and download, forums etc).
- Broad project scope allows for made-to-fit team positions.

Cons / possible problems:

- Lack of large corporate sponsors – we cannot provide major investment or funds for graphics, sound and video production.
- Date of readiness cannot be given – we will of course try to create visible milestones during the project's time frame, but this cannot be 100% guaranteed. But still, everything that is produced by the team members will be published on SourceForge and on the reference server for examination and testing.
- The project uses Groovy Server Pages (GSP) as its template mechanism for dynamic web page rendering. It can be learned quickly and is well documented, but of course some time is needed for familiarization.
- Broad project scope requires good project management and pre-defined goals for individual team members.

Positions (may be shared)

Currently, the following areas of expertise are needed:

- Translation Management
- Interface to other web sites (Facebook, payment providers, Twitter)
- GUI-design / CSS-layout
- JavaScript programming for user interface
- Documentation
- Server side programming
- Testing

For best results and learning opportunities we will try to define specific packages of work so team

members can create modules or features which can be attributed to them directly.⁴

Technical background list

- HTML 4, 5 where useful
- JavaScript (jQuery)
- Java / Groovy / Grails
- database independent (currently: MySQL and Postgres)
- platform independent (runs on Windows and Linux)
- open source: project is hosted at SourceForge: <https://sourceforge.net/projects/littlegoblin/> and available under the Apache 2.0 License
- I18n-ready: code uses properties files (later on: a database) for multiple languages.
- Project language: English, German, or as needed / supplied by team members.

Project data:

Sourceforge: <https://sourceforge.net/projects/littlegoblin/>

Test server: <http://schedim.de>

Developer's blog: <http://dewarim.com/browsergame/status>

Project Manager & Administrator: Ingo Wiarda

Mail: ingo_wiarda@dewarim.de

Phone: +49 162 23 89 333

⁴ Specialization and a sense of ownership is often more useful than being a Jack-of-all-trades and master-of-none.