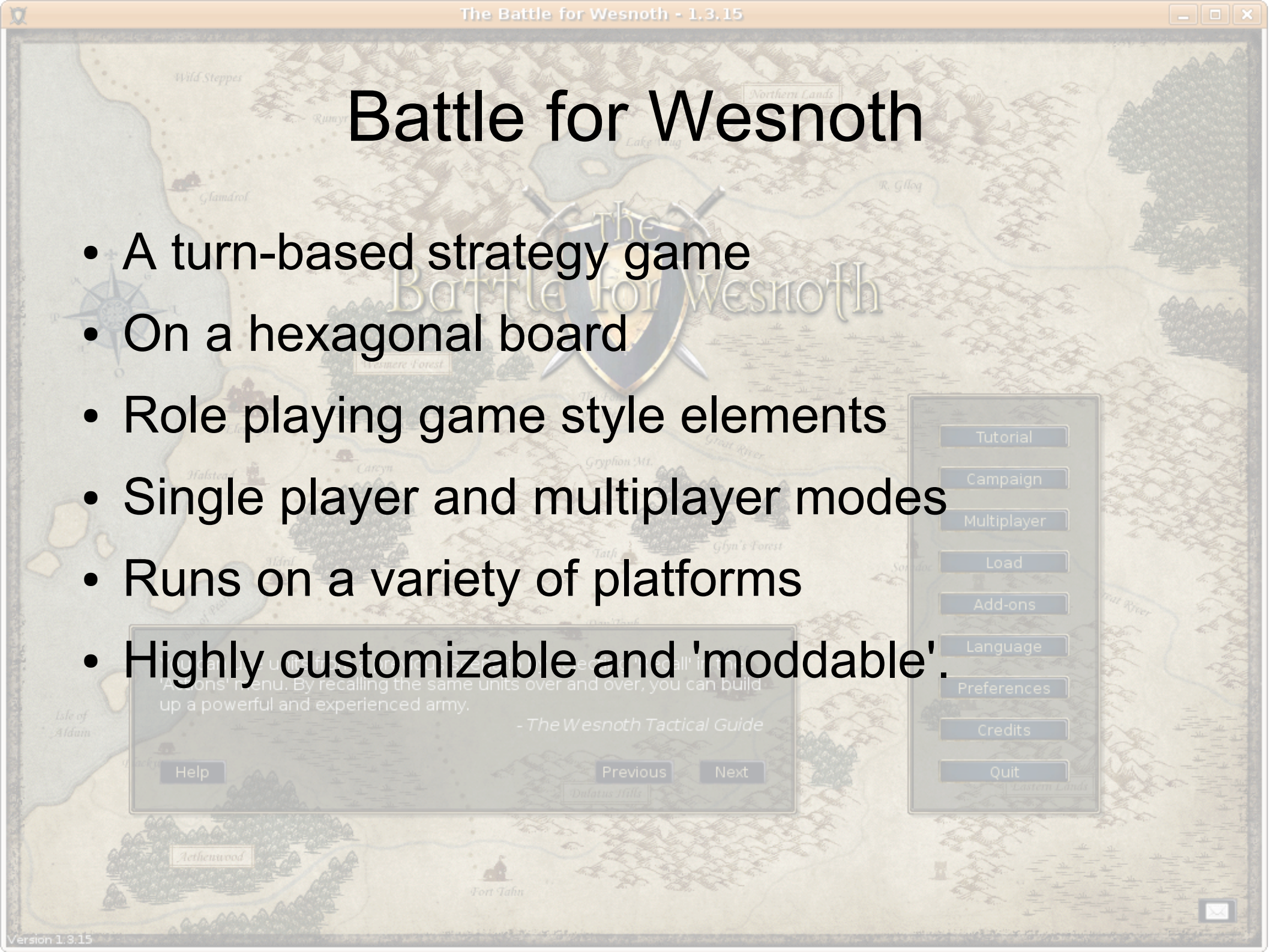

The Battle for Wesnoth - 1.3.15
Battle for Wesnoth
A turn-based strategy game
On a hexagonal board
Role playing game style elements
Single player and multiplayer modes
Runs on a variety of platforms
Highly customizable and 'moddable'.
up a powerful and experienced army.
- The Wesnoth Tactical Guide
Help
Previous
Next
Tutorial
Campaign
Multiplayer
Load
Add-ons
Language
Preferences
Credits
Quit
Version 1.3.15

So what's special about it?

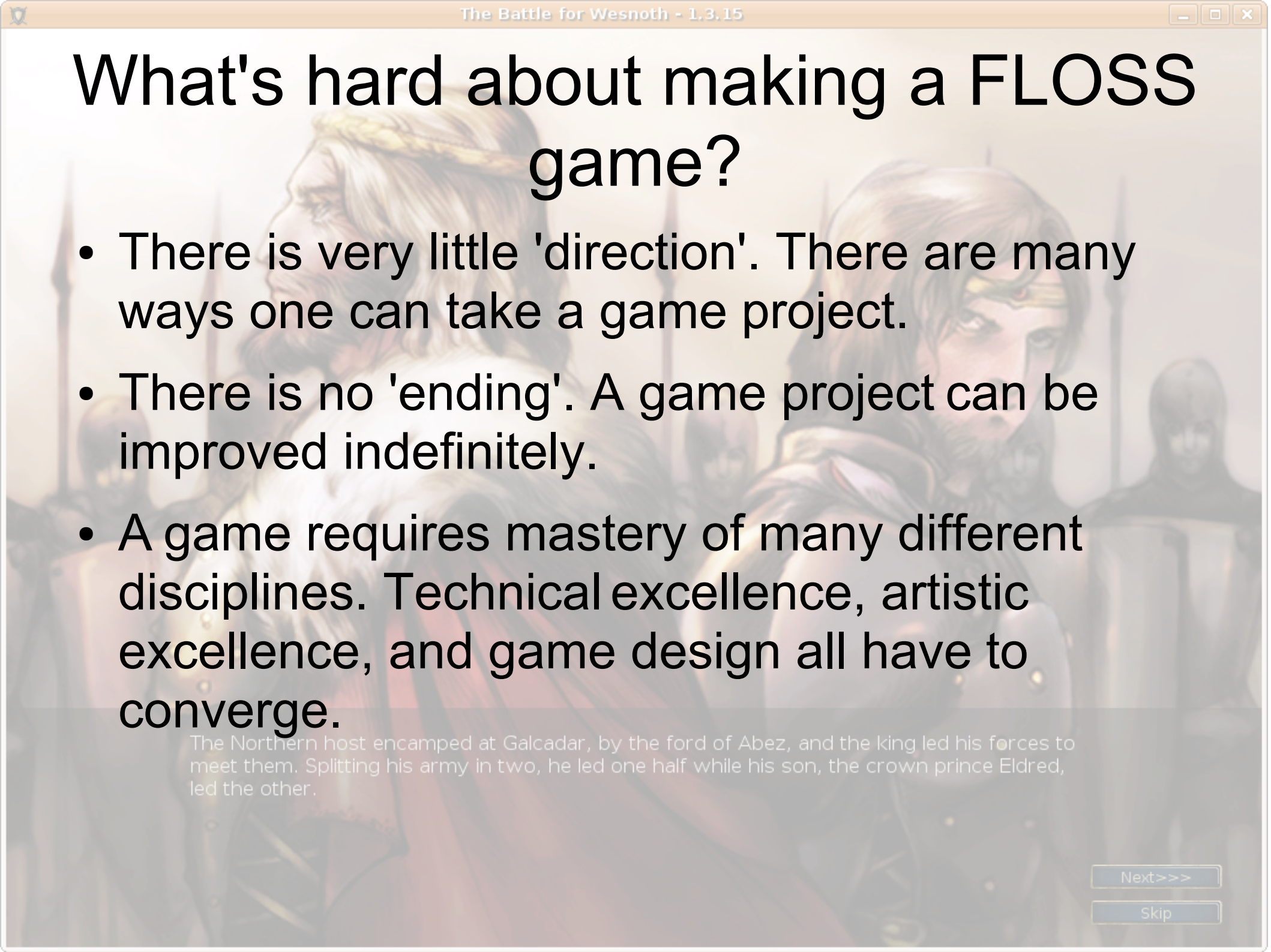
- A large developer and player community
- A mature project, but with active development and many improvements
- High quality artwork: both graphics and music
- Very well-balanced by a tireless team of playtesters
- Fun, unique gameplay

Technologies used

- Advanced C++, with some use of Boost
- The Simple Directmedia Layer (SDL) libraries: SDL, SDL_net, SDL_ttf, SDL_image
- gettext for internationalization
- Python to allow scriptable AIs
- Otherwise, most of Wesnoth's technology is “home grown”.

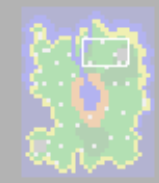
The Wesnoth community

- http://forum.wesnoth.org -- a web forum (phpbb)
- #wesnoth, #wesnoth-dev, #wesnoth-mp and others on irc.freenode.net
- wesnoth-dev@gna.org -- developer mailing list
- http://bugs.wesnoth.org -- bug tracking system


The Battle for Wesnoth - 1.3.15
What's hard about making a FLOSS game?
There is very little 'direction'. There are many ways one can take a game project.
There is no 'ending'. A game project can be improved indefinitely.
A game requires mastery of many different disciplines. Technical excellence, artistic excellence, and game design all have to converge.
The Northern host encamped at Galcadar, by the ford of Abez, and the king led his forces to meet them. Splitting his army in two, he led one half while his son, the crown prince Eldred, led the other.
Next>>>
Skip

How it all began

- In June, 2003, I developed a very simple hex war game and released it as “Battle for Wesnoth 0.1”.
- All major gameplay features were already present in this version.
- Francisco Munoz sent me some improved artwork for the game.
- Further releases were made; a forum set up; a community began forming.



The Wesnoth Engine

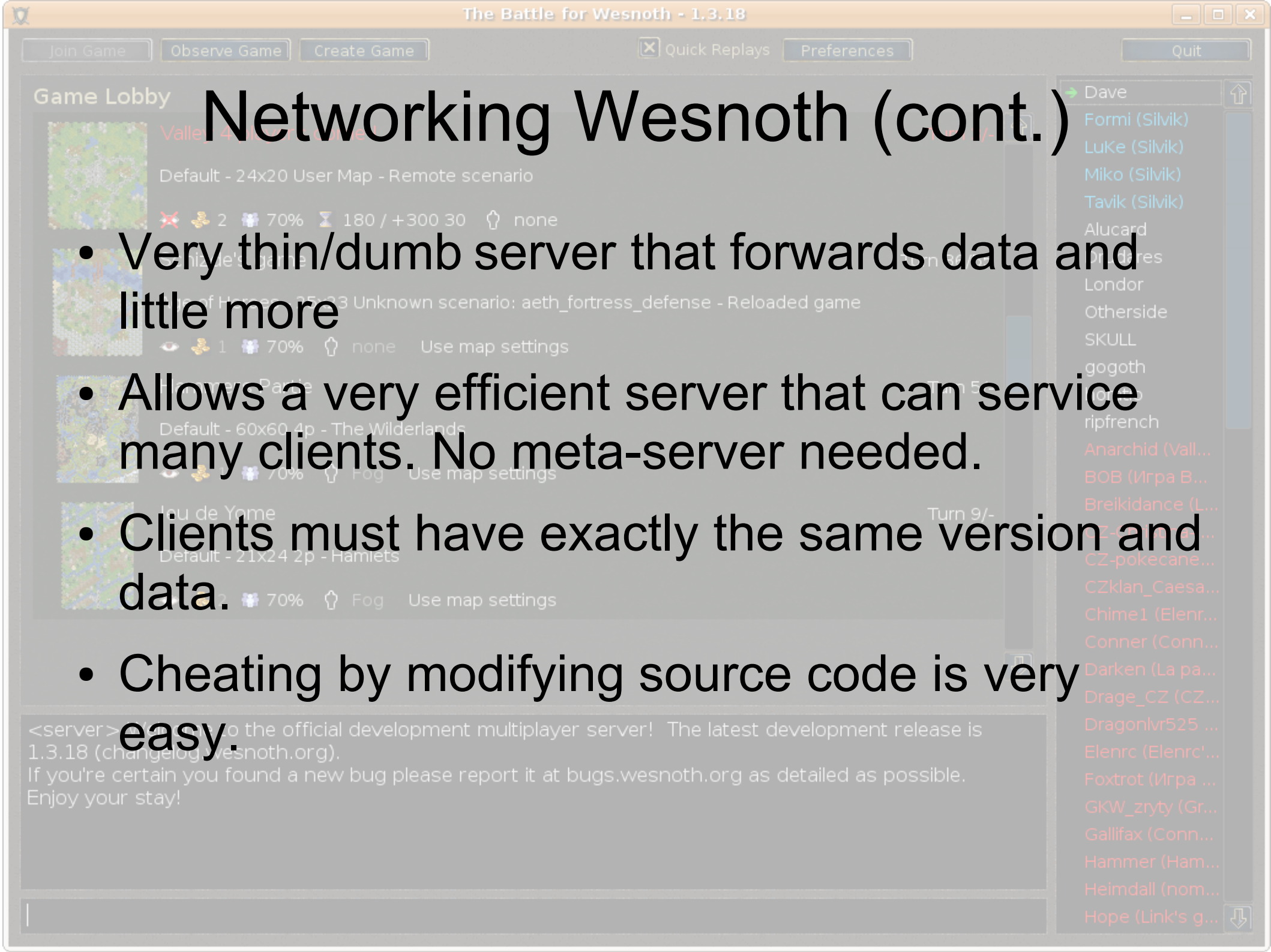
- ~90k non-blank lines of C++ code.
- Modern style of C++, using the STL, templates, exceptions, and some parts of Boost. RAII is used heavily; very few memory leaks.
- Minimal dependencies; we program many things ourselves
- Includes an AI, WML parser, random map generator, theme/widget engine, and support for all game logic.s

Problems with Wesnoth's Design

- SDL: little new development, slow to do many things. However, OpenGL has been determined not to be a reasonable alternative.
- Other SDL libraries (SDL_ttf, SDL_net) have proved to have various stability and other problems.
- Sometimes slow
- Memory hungry

Networking Wesnoth

- Not originally designed to be networked.
- Originally, saves could only take place at the end of a scenario.
- Later, saves implemented as start-scenario + deltas (“replay”)
- This allowed for replays of a game to be stored
- Also allowed sending deltas over the network to implement network multiplayer



Networking Wesnoth (cont.)

- Very thin/dumb server that forwards data and little more
- Allows a very efficient server that can service many clients. No meta-server needed.
- Clients must have exactly the same version and data.
- Cheating by modifying source code is very easy.

Wesnoth Multiplayer

- A sub-community focused on multiplayer soon formed.
- Multiplayer developers began concentrating on tuning and playtesting the game more finely than before.
- Six different factions, many maps, finely tuned and balanced.

Wesnoth Markup Language (WML)

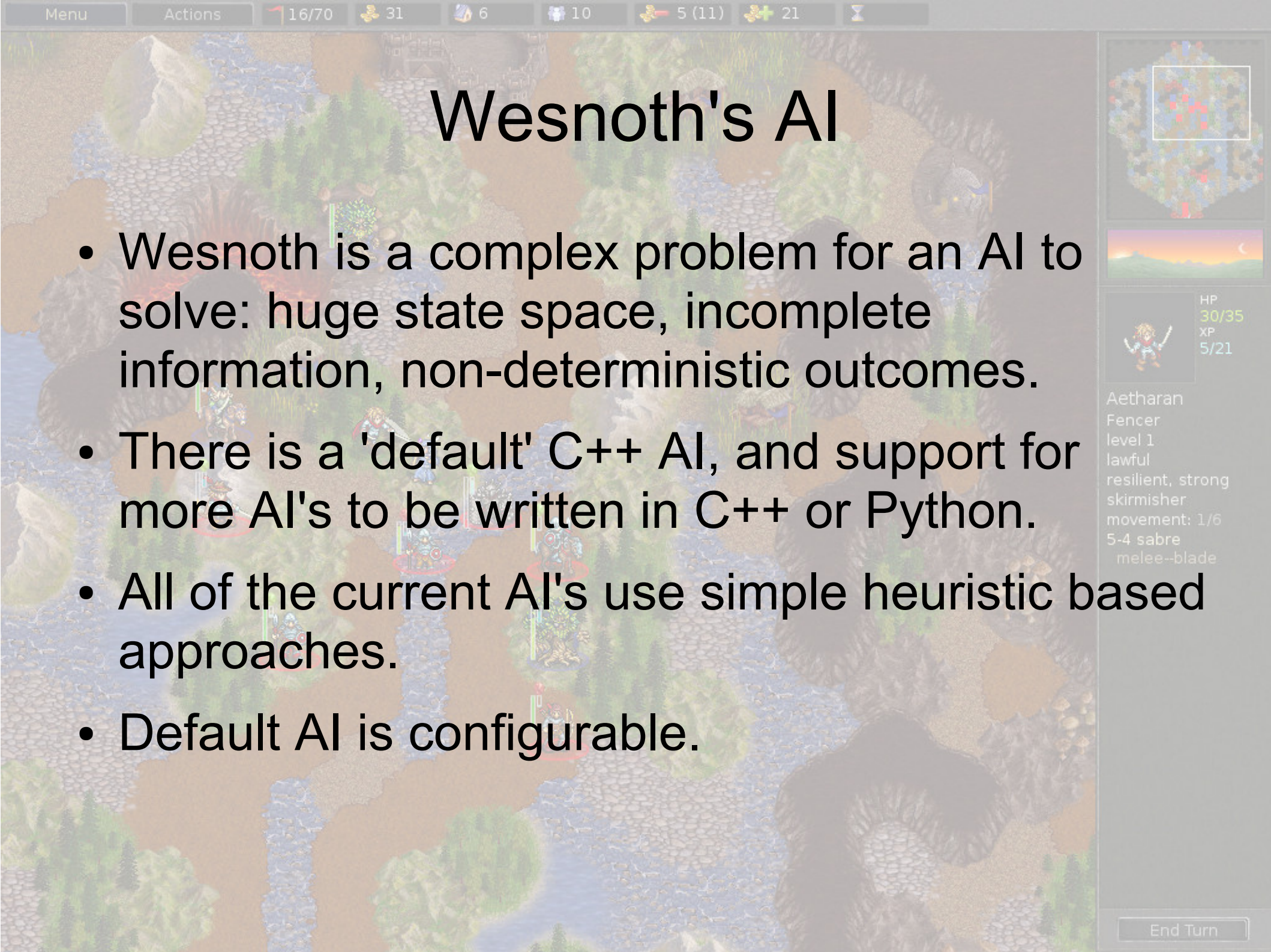
- An XML-like language which is used throughout Wesnoth.
- Is used to create scenarios, campaigns, define units, define display and theme settings, and as the save game and network protocol format.
- Also supports a pre-processor to make things easier.
- Has evolved greatly over time.

Wesnoth Map Editor

- Much of Wesnoth's code is reused to make a map editor.
- Allows easily and advanced creation of maps.
- Doesn't support any WML. One must add units, events, etc to a map oneself.

Wesnoth's AI

- Wesnoth is a complex problem for an AI to solve: huge state space, incomplete information, non-deterministic outcomes.
- There is a 'default' C++ AI, and support for more AI's to be written in C++ or Python.
- All of the current AI's use simple heuristic based approaches.
- Default AI is configurable.

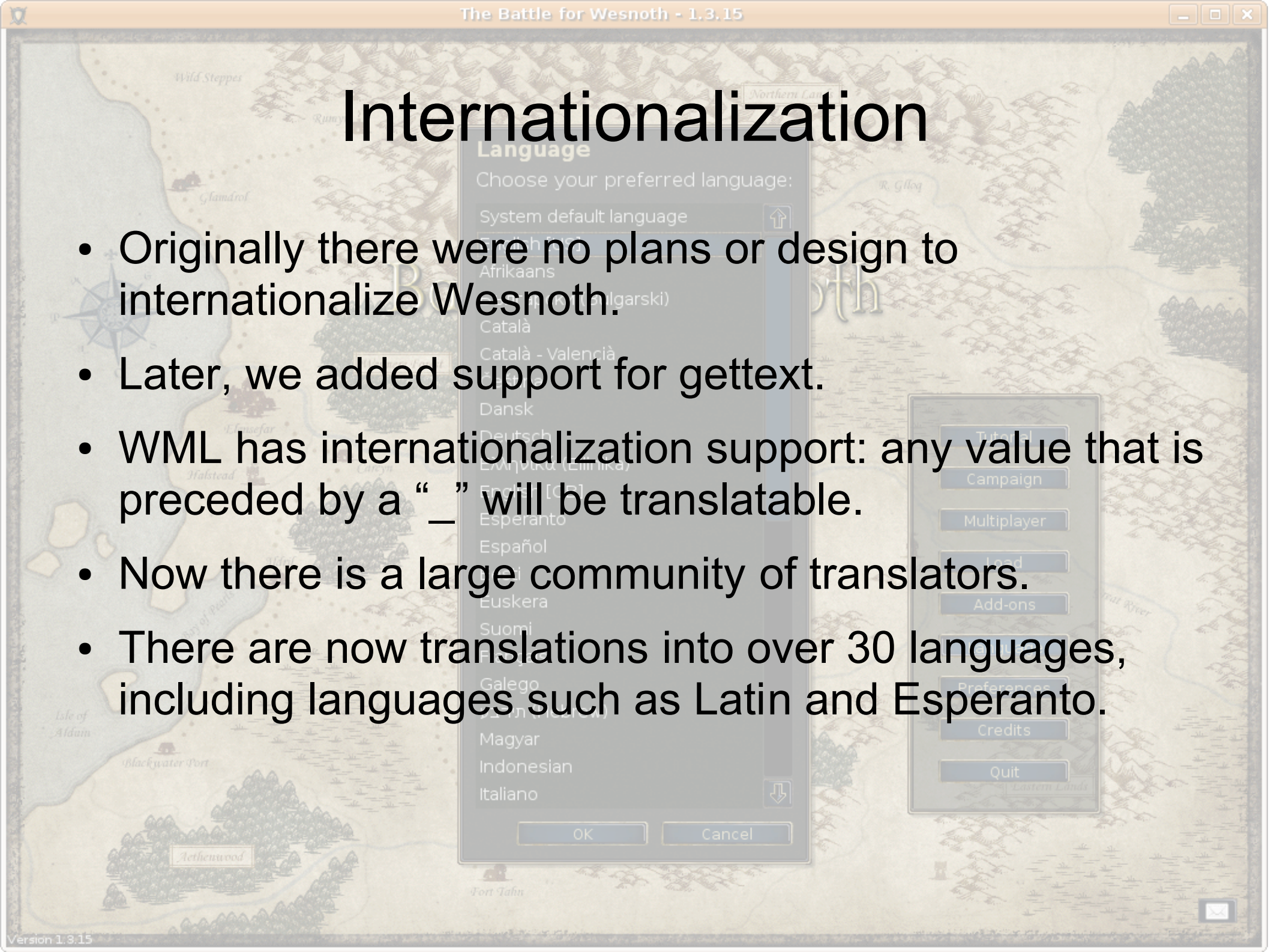


Wesnoth's Artwork

- Wesnoth began with no artists at all.
- Made adding art as easy as possible to attract artists.
- Maintained a policy of “if someone does the art for this feature, I **will** do the code”
- Many of Wesnoth's current artists taught themselves art during development.
- Strong artists work with weaker artists.
- Artists misunderstanding or disliking the GPL and FLOSS has been an ongoing problem.

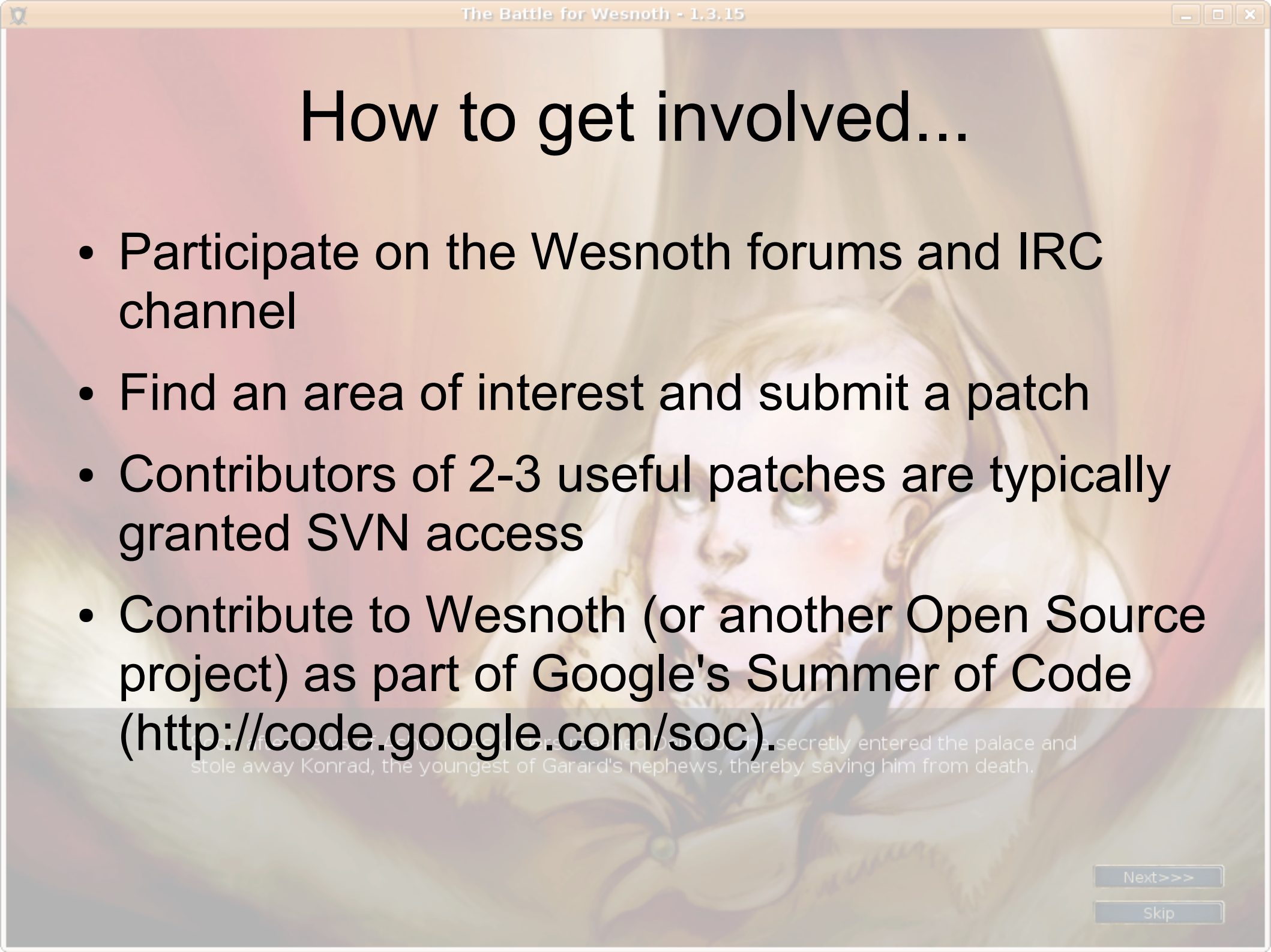
Internationalization

- Originally there were no plans or design to internationalize Wesnoth.
- Later, we added support for gettext.
- WML has internationalization support: any value that is preceded by a “_” will be translatable.
- Now there is a large community of translators.
- There are now translations into over 30 languages, including languages such as Latin and Esperanto.



Other Cool Features

- Wesnoth's community constantly produces cool tools and features.
- Add-on Server
- stats.wesnoth.org (Rusty Russell)
- units.wesnoth.org
- phpbb forum extensions
- WML lint (Eric Raymond)


The Battle for Wesnoth - 1.3.15
How to get involved...
• Participate on the Wesnoth forums and IRC channel
• Find an area of interest and submit a patch
• Contributors of 2-3 useful patches are typically granted SVN access
• Contribute to Wesnoth (or another Open Source project) as part of Google's Summer of Code (http://code.google.com/soc).
stole away Konrad, the youngest of Garard's nephews, thereby saving him from death.
Next>>>
Skip