Balanced Annihilation

Downloads

Intern

- Spring 104.0 Engine Installer for Windows
- Spring 104.0 Engine Portable for Windows
- Spring 104.0 Engine for Linux, precompiled x64 static
- SpringLobby 0.264 Installer for Windows
- SpringLobby 0.264 Portable for Windows
- SpringLobby 0.264 Sources for Linux
- Balanced Annihilation 10.10 - Mod for Spring
- Balanced Annihilation - Mappack

Extern

- Spring engine
- SpringLobby
- Balanced Annihilation 10.10 - Mod for Spring

Game principle

Multiplayer Information

- Internet play: yes
- LAN play: yes
- Lobby search: yes
- Direct IP: yes
- Play via gameranger: no
- coop: yes
- Singleplayer campaign: no
- Hotseat: no

Balanced Annihilation (BA) is based on the game mechanics of Total Annihilation (TA), but implemented via the Spring engine. Its a mod for the Spring engine, which is used by a variety of game mods. The tech tree is somewhat different and the game play was improved. Another game

based on the TA game mechanics is [Tech Annihilation](#).

Installation Client

Windows

1. First install Spring. Newer versions also include SpringLobby. But you will lack any game mods, which utilize the engine, as well as any maps. Theoretically you might already jump to step [Multiplayer matches](#), as game mods and maps can be downloaded from other players via SpringLobby automatically. But if you want to test the game in singleplayer against some bots first, you will need at least one game mod and a map.
2. Open folder `c:\users\<username>\Documents\My Games\spring\games` (or create it, if not present) and place the BA mod there. This still will leave you without any maps.
3. A good interface for online play is provided by [SpringLobby](#). If it is not already present in the Spring package, download it, and place it in a folder of your liking. It is, however recommended, to place it in a dedicated folder, if installed manually.
4. You might need to configure the SpringLobby settings first:
    1. Use the menu **Edit → Preferences → Add new** and
    2. select the file `unitsync.dll` inside your Spring folder.
5. You might as well extract the map pack into folder `c:\users\<username>\Documents\My Games\spring\`. (Eventually there should be a folder `[…]\spring\maps` present). The Linux paths are similar: `<homefolder>/.spring/[…]`.
6. As a last stept, use **Tools → Reload maps/games** to update the in-lobby list of games and maps.

Linux

Usually, Spring and SpringLobby are available from the Linux packaging repository or your Linux distribution. But it is not uncommon, that the available versions are (sometimes quite) outdated. So you might find some other packing sources with newer version to add, to keep up to date. Of course it is a question of personal preferences, what you think about adding external sources to your packaging sources. In any case, after that, it is a lot more comfortable to get the latest versions.

You might as well place Spring with any version of your liking in your user space. You get the required portable packages [here](#). SpringLobby is also affected by a tendency of being outdated in official distributions repositories. If you still want to have the latest SpringLobby version available, have a look at section [Compiling SpringLobby](#). It is however, not as important to have the latest SpringLobby version, as is with Spring and the game mods themselves. This includes BA. As the mods are always placed in the user space, you need not worry about this.

Installation Lobby server

As the use of Spring and in turn BA is a lot easier with a Lobby, but not everybody wants to host his/her games on the official lobby servers, you might be interested in running a private version of the lobby server. It uses the so-called »Uberserver«. The following steps are tested/described for Ubuntu

Linux, but in principle should work very similar at any Debian derived distribution and be similar on other distributions.

- Get Uberserver [here](). Download the `*.zip` file (*not* just `server.py`!) or use `git clone`, to get the files directly from the git repository.
- Some system packages are required to be installed (Python 3 is required as well, but usually is installed already), the rest are python packages:
  1. `sudo apt-get install python3-pip geoip-database python-geoip libgeoip-dev`
  2. `sudo pip3 install Twisted sqlalchemy pyopenssl service_identity GeoIP`
- You can now start the Uberserver from the base folder of the files you downloaded earlier `./server.py --latestspringversion 104.0`. You might want to place this in an Init script or a Systemd Service. There are some other starting options. This is what the command line help can tell you:

[Command line help]()

```
Usage: server.py [OPTIONS]...
Starts uberserver.
()
Options:
  -h, --help
      { Displays this screen then exits }
  -p, --port number
      { Server will host on this port (default is 8200) }
  -n, --natport number
      { Server will use this port for NAT transversal (default is 8201) }
  -g, --loadargs filename
      { Reads additional command-line arguments from file }
  -o, --output /path/to/file.log
      { Writes console output to file (for logging) }
  -u, --sighup
      { Reload the server on SIGHUP (if SIGHUP is supported by OS) }
  -v, --latestspringversion version
      { Sets latest Spring version to this string. Defaults to "*" }
  -m, --maxthreads number
      { Uses the specified number of threads for handling clients }
  -s, --sqlurl SQLURL
      { Uses SQL database at the specified sqlurl for user, channel, and ban
storage. }
  -c, --no-censor
      { Disables censoring of #main, #newbies, and usernames (default is to
censor) }
  --proxies /path/to/proxies.txt
      { Path to proxies.txt, for trusting proxies to pass real IP through
local IP }
  -a --agreement /path/to/agreement.txt
      { sets the pat to the agreement file which is sent to a client
registering at the server }
  -r --redirect "hostname/ip port"
```

```
      { redirects connecting clients to the given ip and port
SQLURL Examples:
  "sqlite:////absolute/path/to/database.txt"
     { uses a database in the file specified }
  "sqlite:///relative/path/to/database.txt"
     { note sqlite is slower than a real SQL server }
  "mysql://user:password@server:port/database?charset=utf8&use_unicode=0"
     { requires the MySQLdb module }
  "oracle://user:password@server:port/database"
     { requires the cx_Oracle module }
  "postgres://user:password@server:port/database"
     { requires the psycopg2 module }
  "mssql://user:password@server:port/database"
     { requires pyodbc (recommended) or adodbapi or pymssql }
  "firebird://user:password@server:port/database"
     { requires the kinterbasdb module }
()
Usage example (this is what the test server uses at the moment):
 server.py -p 8200 -n 8201
()
```

The version number for `--latestspringversion` should always be set to the very latest version of Spring.

Multiplayer matches

Lobby server accounts

Multiplayer matches can be created rather easily via the lobby servers. The manual way is quite a hassle in comparison. The mobilen Infanterie also runs a lobby server, see status page. To join a lobby server, use **Server → Connect…** and you will get to the login screen. Enter the server address of the mobile Infanterie (`www.mobile-infanterie.de`)and the standard port 8200. If you are logging in for the first time, you will have to create a user account, first. It does not require any personal data. Only a username and a password. The registration is accessible via Tab **Register**. After creating a user account you can join that lobby server via tab **Login**).

Opening and joining matches

You can open up new matches via the **Battle-Room** tab or join an existing match via tab **Battlelist**. If you are playing actually really online, it might be useful to tick the option **Hole Punching** in the section **NAT traversal** if you cannot or do not want to open ports on your router. This may not be a good idea, if you are actually playing on LAN. If you tick the option, you will not be able to connect to the host.

Setting up a match

In general, there is one confusing detail about creating new matches: There is a big difference between "Team" and "Ally". Ally is, what other games usually refer as "team", players allied with each other, controlling their own faction, meaning units, buildings ans ressources. In BA, that is a bit different, as there can be more than just one player inside one team, and teams can be allied to each other (or not). But all players inside one team share the same resources. In essence, there are two players controlling the very same faction. So if you are only interested in vanilla play, one player for each team, never place two players in the same team, but place friendly players in the same Ally. If you try to place some players in the same team, BA/Spring has a tendency of not starting properly.

Bots

There is some selection possible, what bots you want to have in your match. They differ a bit in their play style. Some of them are not useful at for BA. E. g., some simply don't do anything at all, others crash the game altogether. Here is a short overview of how the different bots behave:

- **AAI 0.9**: Rather simple AI. At least in version 0.9 it is recommended, to not have more than three of them in the match, as the game has a tendency of taking to long, until the point, when Spring crashes.
- **E323AI 3.25.0**: Builds up rather slowly, but wait until it finds your presence on the map…
- **KAIK 0.13**: Hard opponent.
- **RAI 0.601**: Has a tendency of harass-like play style. Has also a tendency of crashing the game, when it takes to long.
- **Shard dev**: Hard opponnent. Has a harass-like play style, but never seems to finish you off.
- **CppTestAI 0.1**: Does absolutely nothing.
- **NullAI 0.1**: Does absolutely nothing.
- **HughAI**: Does absolutely nothing. Crashes the game if you kill its commander.
- **NullJavaAI 0.1**: Does absolutely nothing. Crashes the game if you kill its commander.
- **NullOOJavaAI 0.1**: Does absolutely nothing. Crashes the game if you kill its commander.

(These AI tests were done with Spring 100.00 and BA 9.07.)

Compiling SpringLobby

There are precompiled binaries for SpringLobby for Windows, so actually no need to compile SpringLobby for Windows yourself. For Linux that's a different matter, as SpringLobby is usually outdated on the distributions repository. So it might be in order, to compile SpringLobby on your system for yourself. Here is a description on how to do that on Ubuntu (and probably any other Debian based distribution):

1. Download the Sources from http://springlobby.info/landing/index.php and extract them somewhere.
2. Do the steps described in the file INSTALL:
    1. `cmake .` configures your source package, so that the next step is executed in a specifically fitting way for your system. (Hint: CMake 3.1 or newer ist required (SpringLobby Version 0.264).)

2. `make` runs the actual compiling.
3. `sudo make install` installs SpringLobby. (This step might not require super user permissions, if the installation path was set to a destination in your user space via `ccmake .`)

It should work like this or very similar on any other current Linux distribution. For Ubuntu 17.04 the following Packages are required (if not present already): `build-essential`, `cmake`, `libwxgtk3.0-dev`, `libcurl4-nss-dev`, `zlib1g-dev`, `libpng-dev`, `libalure-dev` and `libboost-all-dev`. Or in one command:

`sudo apt-get install build-essential cmake libwxgtk3.0-dev libcurl4-nss-dev zlib1g-dev libpng-dev libalure-dev libboost-all-dev`

Weblinks

- BA home page
- BA forums
- BA Discord

See also

- Total Annihilation Mechanics

Back to the games database