IN DEBIAN

The Bub’s Brothers: a 21 year old ITP comes back to play

PUBLISHED 23 November 2026 EN

«Great to see that someone is trying to make that old project survive!» — Armin Rigo, author of The Bub’s Brothers, 2026

«Something I wasn’t expecting to ever hear about again, TBH. Good to hear.» — David Gowers, who redrew its graphics in 2007

«it works! :)» — Alex Myczko, Debian Developer, 2026, after the first game over the Internet in years

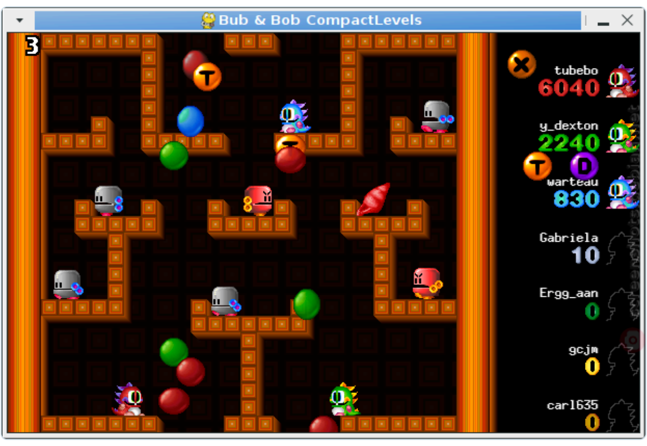
Chapter 1: A Mac game, cloned for the network (2003)

In November 1997 Sebastian Wegner, known as McSebi, released **Bub & Bob** for the Mac as shareware, his own take on Taito’s arcade classic Bubble Bobble. Two dragons, bubbles, monsters trapped inside them, fruit everywhere.

On 31 January 2003 Armin Rigo, one of the founders of PyPy, made the first commit of a network clone of McSebi’s game, written in Python. The old site said it plainly: *«This version is a direct clone of the MacOS game Bub & Bob of McSebi»*, and thanked him: *«Thanks Sebi for all the cool graphics and sounds!»*

It didn’t stay a one-man project for long. Gio, Odie, Michel-Stéphane and Armin wrote levels in Python, paxed improved the random levels, and David Gowers redrew the graphics in 2007. And in 2005 McSebi did something that turned out to matter a lot: he put his graphics and sounds under the Artistic License.

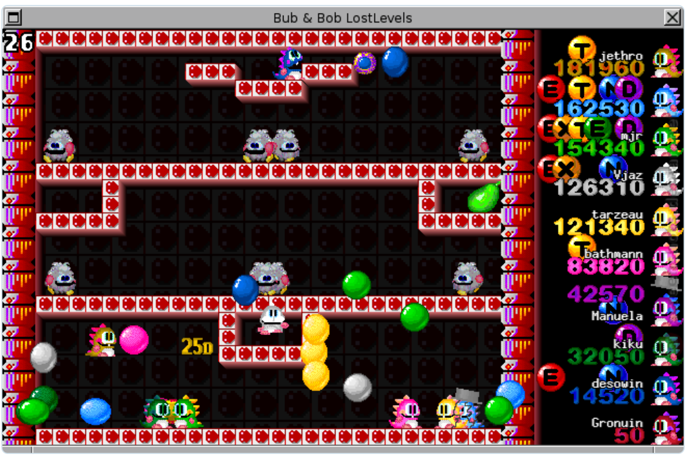
The last commit is from 12 October 2013. Then silence.



Seven players on version 1.6.2, the last release. Screenshot: screenshots.debian.net.

Chapter 2: Ten years in Debian (2005 to 2019)

In 2005 a Debian Developer filed an ITP, an “Intent To Package”, for the game (#308148). It reached the archive in February 2008, and for more than ten years `apt install bubbros` got you a LAN party in one command.



Around 2008, in Debian: a ten-player game over the network. Look for tarzeau on the scoreboard: that’s Alex. Screenshot: screenshots.debian.net.

Then Python 2 reached its end. In October 2018 someone asked for a move to python3-pygame (#912480). Nobody did it. On 30 September 2019 the game left Debian, and the removal bug says why in four words: *«python2 removal; dead upstream»* (#936248).

Chapter 3: A Debian Developer with a long memory (2026)

That Debian Developer was Alex Myczko, and he never forgot the game. In 2021 he imported it from SourceForge into his own GitHub repository, and in 2024 Christopher Bock sent a first automatic conversion to Python 3. Then, in 2026, I wrote my first post on Planet Debian, about rescuing old software. Alex read it, got in touch, and asked if I fancied porting the game to Python 3.

Of course I did. Bringing old games and software back is what I enjoy most. I spent 20 years tracking down the authors of Conquer, a 1987 Unix game, until it became free software, and I adopted xwpe, a 1993 programming editor. A game people loved, thrown out only because the language under it moved on? That’s my kind of project.

At some point I asked him on GitHub whether that 2005 ITP was his. His answer: *«yeah that was me»*.

> - ***2005:*** *Alex files the ITP.*
> - ***2019:*** *«python2 removal; dead upstream».*
> - ***2026:*** *«yeah that was me».*

Chapter 4: Bringing it back (2026)

Alex reviewed and merged everything in his repository and tagged the releases, 2.0 up to 2.3. Along the way anzz1 joined in with a lot of fixes, from sharper scaling to releasing the keys after a rewind. Most of the port was the usual Python 2 to 3 work. A few problems were properly fun, though.

The Time Machine

One bonus is a clock that rewinds the game. On Python 2 it relied on `statesaver.c`, a C module whose own comment brags *«This one can copy running generators and their frames!»*. It did exactly that, poking straight into CPython's frame structures:

```
f2->f_lasti = f->f_lasti;
f2->f_iblock = f->f_iblock;
memcpy(f2->f_blockstack, f->f_blockstack, sizeof(PyTryBlock)*f->f_iblock);
```

There's no way to port that. Those structures are private and change with every Python release, and PyPy can't copy generators either. Without the module the game just printed *«'statesaver' module not compiled, no clock bonus»*.

So now the game simply plays the past again. When a board starts it saves a checkpoint of plain data: players, dragons, monsters, bonus state, and the state of the three random generators it uses. Every input from the network (keys, players joining or leaving) is logged with the pass number it arrived in. To rewind, the game rebuilds the board from the checkpoint and runs it forward again, with the sound off, up to the moment you land on.

```
cp['random'] = random.getstate()
cp['bonusrandom'] = (bonuses.random, bonuses.random.getstate())
cp['localrandom'] = bonuses.DustStar.localrandom.getstate()
```

That only works if the game is fully deterministic, so there's a test for it. It runs a server with two scripted players and hashes the screen and the game state after every pass. Two runs with different `PYTHONHASHSEED` values must give identical hashes. A rewind recomputes 1,644 passes, and the 602 passes that follow must match a run that never went back. (commit)

Bytes on the wire

The network protocol glued message codes, which were `str`, onto `struct.pack` output. Python 3 says no to that, so the server died building the very first icon. Now everything on the wire is `bytes`, and `message()` raises `TypeError` if you hand it a `str` instead of guessing an encoding:

```
-MSG_DEF_PLAYFIELD = "p"
+MSG_DEF_PLAYFIELD = b"p"
-        limit = ord(data[0]) + 1
+        limit = data[0] + 1      # indexing bytes gives an int
```

Then there's division. I walked the syntax tree for every `/` in the code, 82 of them, and 13 divided two ints and needed `//`. One of them, `'>'*(width/2-2)`, blew up only when you loaded a level without wind. The nice part: the protocol didn't change, so a Python 2 client still plays against a Python 3 server, and the other way round. (commit)

The clicks that missed

To join a game you click a dragon. Alex tried it on Wayland and wrote: *«i saw graphics, heard music, but no input worked»*. The GTK client read the mouse on the top-level window. With client-side decorations, the window's coordinates include the title bar and the shadow, so on GNOME, KDE and Weston every click landed 26 pixels to the right and 60 lower. The dragons sit at the bottom of the screen, so your click landed below the playfield and nobody could join. The fix was to read the mouse on the drawing area instead:

```
-    w.connect("button-press-event", self.button_press_event)
+    self.area.connect("button-press-event", self.button_press_event)
```

(PR #8)

Wayland and full screen

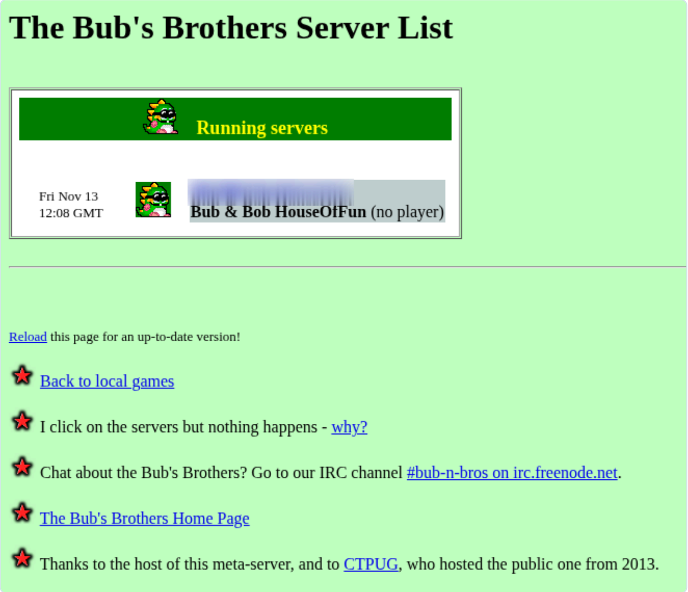
The client picked the first display driver that loaded, and X11 was first in the list, even with no X server around. Now X11 goes last. Full screen asked for a 656x400 video mode, and Wayland compositors never let an app change the mode: Sway left it in a window, others drew a tiny playfield in a corner. pygame 2's `SCALED` flag fixed full screen. For a resized or maximized window the game now scales the picture itself, keeping the proportions.

To stop this from breaking again there are display tests with real input: X11, Weston, Sway, KDE and GNOME, each in its own headless session. Most of them get input from `xdotool`. One gotcha: you must not pass `--window`, because then the events are synthetic, and Weston's X11 backend throws those away. GNOME has no X input at all, so there the test drives the mouse through Mutter's RemoteDesktop interface over D-Bus. Each test clicks a dragon, presses keys, and checks that the game saw them and that the picture fills the window. With the old code, the GTK client fails on Weston, KDE and GNOME. (commit)

Then Alex wrote from Hyprland: *«fullscreen works now. and also input works (i only had completely forgotten how it works, clicking dragon, then keys for left/right/jump/bubble :) we should setup an internet server, and have some fun»*

The dead server

The public list of games used to live on codespeak.net, the server from the early PyPy days, which is long gone. The game could hang or spit errors trying to reach it. Now the server is a setting, and there's a systemd unit to run your own. Alex set one up at bubble.aiei.ch, then wrote *«it works! :)»*, and: *«@vejeta what about you join?»* So I joined, and we played over the Internet, 13 years after the last commit.



The Bub's Brothers Server List

Running servers

Fri Nov 13
12:08 GMT

[redacted] **Bub & Bob HouseOfFun** (no player)

Reload this page for an up-to-date version!

- Back to local games
- I click on the servers but nothing happens - why?
- Chat about the Bub's Brothers? Go to our IRC channel #bub-n-bros on irc.freenode.net.
- The Bub's Brothers Home Page
- Thanks to the host of this meta-server, and to CTPUG, who hosted the public one from 2013.

The Internet server list is back, on Alex's meta-server. I blurred the server's address.

Chapter 5: Where did the music come from?

Debian wants to know the licence of every file, and the game has seven subgames, each with its own tune. Nobody knew where those tunes came from. The old CVS log named just one: *«Red Baron music.»* The others only said things like *«ext6.»*

I compared them against a big pile of game soundtracks and module music, and all I learned was that each tune sounds like the game its subgame copies: Arkanoid, Pac-Man, Galaga, Tetris, Lemmings. That much I could have guessed. I asked Armin. *«I have absolutely no idea any more where the music came from.»* Fair enough, it had been more than twenty years.

Then he read a draft about the game, and something clicked: *«I'm pretty sure they are all extracts of music from OCRemix.org.»*

That was the clue. OverClocked ReMix is a huge fan site of video game remixes. I took every remix posted there before the tunes were committed, about 1,200, and slid each loop along them with a spectrum correlation, at a few playback speeds, then checked the hits on the raw waveform. Four of them matched: "Black Block" by DJComet for Arkanoid (played a semitone lower), "64-bit" by Peterrian Productions for Pac-Man, "MkVaffQuasi Ultimix" by MkVaff for Tetris, and "DABOMB" by alk for Lemmings. Every second of each loop scores 0.94 or more against its remix, and under 0.3 against the rest. The script is in the repo, so anyone can check.

The Tetris one surprised me. The old README said it was *recorded from the game*, the Game Boy Tetris. It isn't. MkVaff's remix just opens with a Game Boy imitation good enough to fool everybody for twenty years.

Galaga was missing. It wasn't among the remixes OCRemix still publishes, and that turned out to be the hint: the site has withdrawn quite a few remixes over the years. MusicBrainz keeps a list of them, and an old mirror of the removed files survives in the Wayback Machine. In there was "Galagzee" by zee-roh. Waveform correlation 0.98, against 0.014 for a random control. Gone from the site, but the Internet Archive still had it.

Red Baron was the one the log already named, but which Red Baron? Not the sequels, and not "Snoopy vs. the Red Baron" either, that one is in another key. A fingerprint search found YouTube recordings of the title music of the 1990 DOS game by Dynamix that clearly matched the tune, but not the recording. So I booted the game in DOSBox with AdLib sound and listened to the title screen. Same melody, same tempo, a different arrangement. Somebody arranged it, and we'll probably never know who. When I showed Armin, he also thought it might come from that 1990 game.

And then there's ext6, a rock loop for the subgame called Tron, which was later rewritten into a snake game (that name sent me down the wrong path for a while). I searched every OCRemix, current and withdrawn, the old versions on archive.org, the whole VGMix archive, a long list of game music bands from Minibosses to Machinae Supremacy, the music of free games in Debian, Tron soundtracks, Armagetron. Even Shazam. Nothing. If you recognise it, please tell us.

OverClocked ReMix only allows unmodified, non-commercial copies, and the other sounds have no known author. So the original tunes and the seven subgame sound effects moved to a `nonfree/` folder, and the game got free replacements with the same file names. I picked them by ear, over three rounds, to fit each subgame: CC0 and CC-BY tracks from OpenGameArt by Eric Matyas, HydroGene, Cleyton Kauffman, OwlishMedia and Jonathan Shaw. For Tetris there's Korobeiniki, the old folk tune, generated by a small script, so it can be rebuilt from source. The sound effects I made from scratch with a little synthesizer script. If the original files are installed the game plays them, and if not it plays the free ones. PROVENANCE.md lists where every file comes from.

Everything else has a clear licence. The code is MIT. McSebi's graphics, sounds, music and levels are under the Artistic License he granted in 2005, in answer to the first ITP. David Gowers put his 2007 graphics under the Artistic License 2.0, and Armin confirmed the Python levels are MIT. That leaves three fan-made level packs with no licence we can find. The Debian package ships only the free part, and a separate package in contrib, bubbros-nonfree-installer (#1153895), downloads the original tunes and those levels for you. If you made one of those level packs, or know who did, please tell us.

Come and play

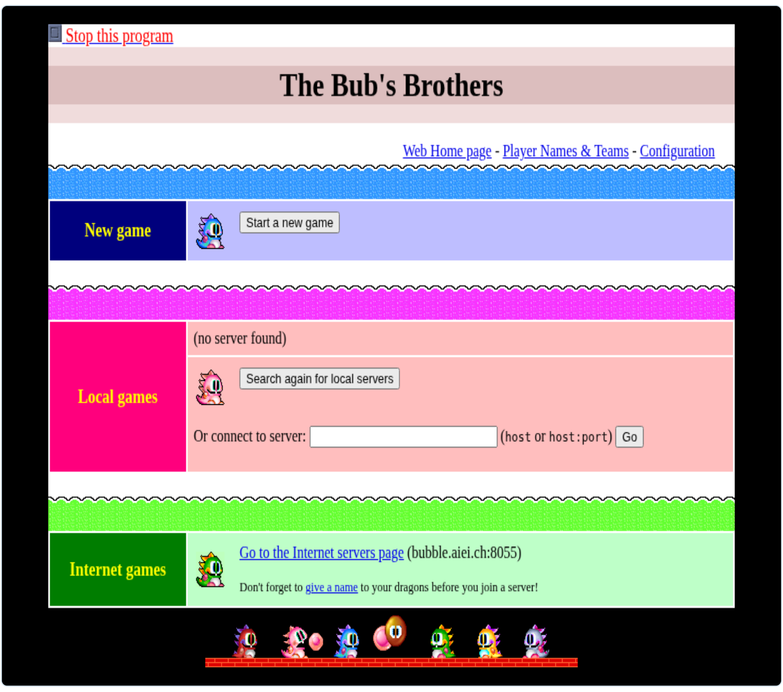
We need testers. You need Python 3 and pygame (`python3-pygame` on Debian):

```
git clone https://github.com/alexmyczko/bubbros.git
cd bubbros
BROWSER=firefox BUBBROS_METASERVER=bubble.aiei.ch python3 BubBob.py
```

The game opens a control page in your browser. From there you start a game or join one from the Internet list, and the game window opens. Click a dragon, then use the keys for left, right, jump and bubble.

bubble.aiei.ch is Alex's own server, and he plans to keep it running.

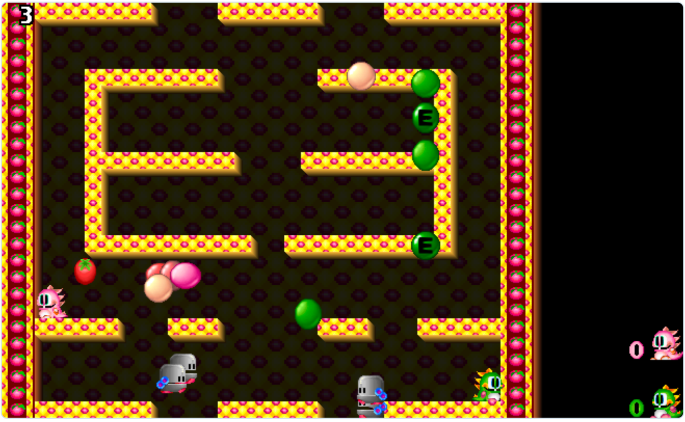
Tell us what breaks, and on which desktop, in the issues. And if you know your way around Debian, the package is waiting to get back into the archive (ITP #1153735).


Stop this program
The Bub's Brothers
Web Home page - Player Names & Teams - Configuration
New game
Start a new game
Local games
Search again for local servers
Or connect to server:
(host or host:port)
Go
Internet games
Go to the Internet servers page (bubble.aiei.ch:8055)

(no server found)

Don't forget to give a name to your dragons before you join a server!

The control page that opens in your browser.



Two players on board 3, on Python 3.

What I take away from it

An old ITP is a kind of promise, and this one was kept 21 years later. McSebi's two lines from 2005 are the only reason his graphics can ship in Debian today, so if you write something, say what licence it's under. The game worked fine on X11 and failed on the desktops people actually use now, and only testing on those desktops showed why. And above all, this is a game made by people. Armin, McSebi, David, Odie, paxed, Gio and Michel-Stéphane made it. Now Alex, anzz1, Christopher Bock and I have picked it up where they left it.

If you like this kind of story, Conquer is the long one: a 1987 Usenet game, and twenty years of chasing its authors until it became free software.

What game would you bring back, and who would you want to play it with?

COMMENT ON MASTODON ↗

TAGS

#debian #planet-debian #bubbros #games #software-archaeology

OLDER →
Hello, Planet Debian

vejeta .

© 2026 Juan Manuel Méndez Rey. All rights reserved.

GitHub

@vejeta@mastodon.sdf.org

debian.vejeta.com

wolfi.vejeta.com

conquer.vejeta.com

Universidad Quantum

CC BY-NC-SA 4.0

MADE WITH DEBIAN

Built with Astro. Typefaces Space Grotesk,
Geist Mono and Silkscreen.