

Games

Get your game on!

- [Wine](#)
- [Steam](#)
- [DOSBox](#)
- [ScummVM](#)
- [OpenRCT2](#)
- [CorsixTH](#)
- [ioquake3](#)
- [DXX-Rebirth](#)
- [Minecraft](#)
- [SimCity 3000 Unlimited](#)
- [UT2004 \(Atari DVD Release Version\)](#)

Wine

Install Wine to play (almost) any Windows game on Linux. For a compatibility list refer to [ProtonDB](#).

Install Wine from the repositories:

```
pacman -S wine wine-gecko wine-mono winetricks
```

DXVK

DXVK is a Vulkan-based translation layer for Direct3D 9/10/11 which allows running 3D applications on Linux using Wine.

Install DXVK from AUR:

```
yay -S dxvk-bin
```

NOTE: If you set up Wine with a non-default prefix (i.e. your Wine "installation" does not reside under `~/.wine`) you will need to supply it as a temporary environment variable:

```
WINEPREFIX=your-prefix setup_dxvk install
```

WARNING: DXVK overrides the DirectX 10 and 11 DLLs, which may be considered cheating in online multiplayer games, and may get your account **banned**. Use at your own risk!

Install DXVK libraries into your Wine prefix:

```
setup_dxvk install
```

VKD3D-Proton

VKD3D-Proton aims to implement the full Direct3D 12 API on top of Vulkan.

Install VKD3D from AUR:

```
yay -S vkd3d-proton-bin
```

NOTE: If you set up Wine with a non-default prefix (i.e. your Wine "installation" does not reside under `~/.wine`) you will need to supply it as a temporary environment variable:

```
WINEPREFIX=your-prefix setup_vkd3d_proton install
```

WARNING: VKD3D-Proton overrides the DirectX 12 DLLs, which may be considered cheating in online multiplayer games, and may get your account **banned**. Use at your own risk!

Install VKD3D-Proton libraries into your Wine prefix:

```
setup_vkd3d_proton install
```

MIDI Playback

Some Windows games still use MIDI playback for music. In order for this to work in Wine, a sequencer has to be installed, e.g. `fluidsynth`:

NOTE: FluidSynth uses soundfonts to render MIDI music.

```
pacman -S fluidsynth soundfont-fluid
```

FluidSynth comes with a systemd user unit to run it in daemon mode. Edit the file `/etc/conf.d/fluidsynth` and uncomment the lines with the environment variables. Point the `SOUND_FONT` variable to a soundfont file in `*.sf2` format (refer to [DOSBox](#) for a list of available soundfonts for installation). Furthermore, adjust the `OTHER_OPTS` variable to use the appropriate audio backend that you are using, e.g. set parameter `-a pipewire` if you're using PipeWire instead of PulseAudio:

```
# Mandatory parameters (uncomment and edit)
SOUND_FONT=/usr/share/soundfonts/FluidR3_GM.sf2

# Additional optional parameters (may be useful, see 'man fluidsynth' for further info)
OTHER_OPTS='-a pipewire -m alsa_seq -p FluidSynth\ GM -r 48000'
```

After you've set everything up, enable/start the systemd user unit with:

ATTENTION: Enable/start the unit as regular user, i.e. do not use `sudo` !

```
systemctl --user enable --now fluidsynth
```

Steam

The popular game distribution platform and library management client from Valve.

NOTE: activate the `multilib` repository in `/etc/pacman.conf` to install Steam/Proton.

```
pacman -S steam
```

Where are my game saves for Proton-enabled games?

Games that are running via Steam Proton on Linux all use their own Wine prefix. The location of game data is here:

```
~/.steam/steam/steamapps/compatdata/[game ID]/pfx
```

You can look up a game's ID by accessing its properties in the steam client:

1. Right-click on the game
2. Select "Properties..."
3. Go to "Updates"
4. Note down the "App ID"

Then navigate the directory structure in the `drive_c/` subdirectory that replicates a Windows directory structure. Personal files are under the `drive_c/users/steamuser/` directory.

DOSBox

Install DOSBox Staging for a more enhanced DOS gaming experience:

```
yay -S dosbox-staging
```

General MIDI/Soundfonts

The integrated FluidSynth MIDI sequencer has issues with some soundfont files, resulting in minor to major music playback issues in games. Timidity++ does not have these issues.

To install simply:

```
pacman -S timidity++
```

A list of available soundfonts to install from the AUR, [sorted by votes](#):

AUR Package Name	Description
<code>soundfonts-aur-meta</code>	Installs all the soundfont packages in the AUR
<code>soundfont-unison</code>	A lean and clean GM/GS soundbank
<code>soundfont-sgm</code>	A balanced, good quality GM soundbank
<code>soundfont-titanic</code>	A public domain, high quality MIDI soundfont by Luke Sena
<code>soundfont-generaluser</code>	A small and well balanced GM/GS soundbank for many styles of music.
<code>soundfont-zeldamcsf2</code>	Legend of Zelda: Minish Cap soundfont for MIDI playback
<code>soundfont-zelda3sf2</code>	Legend of Zelda: Link to the Past soundfont for MIDI playback
<code>soundfont-fatboy</code>	A free GM/GS SoundFont for classic video game MIDI, emulation, and general usage
<code>soundfont-arachno</code>	GM/GS soundbank courtesy of Maxime Abbey.
<code>soundfont-sso-sf2</code>	The Sonatina Symphonic Orchestra by Mattias Westlund. (SF2 format)
<code>soundfont-toh</code>	Don Allen's Timbres of Heaven soundfont
<code>soundfont-opl3-fm-128m</code>	A SoundFont designed to simulate the classic MIDI sound of the Sound Blaster 16 (and other YM262 enabled hardware).

AUR Package Name	Description
<code>soundfont-sunshine-perc</code>	Five drum/percussion soundfonts from Sunshine Studios. Non-commercial use only.
<code>soundfont-realfont</code>	GM soundbank by Michel Villeneuve
<code>soundfont-personalcop</code>	A large free SoundFont.
<code>soundfont-jeux</code>	Jeux organ soundfont

Configure Timidity++ to use the soundfont of your choosing in its global config file

`/etc/timidity++/timidity.cfg`:

```
soundfont /usr/share/soundfonts/FluidR3_GM.sf2
```

Set up `timidity++` to run in daemon mode and start with user login:

```
systemctl --user enable --now timidity
```

You need to tell DOSBox which MIDI Port to send MIDI data to. Install the `alsa-utils` package and list the available MIDI ports with `aconect`:

```
pacman -S alsa-utils
aconect -o
```

The output might look something like this:

```
client 14: 'Midi Through' [type=Kernel]
  0 'Midi Through Port-0'
client 128: 'TiMidity' [type=User,pid=89573]
  0 'TiMidity port 0 '
  1 'TiMidity port 1 '
  2 'TiMidity port 2 '
  3 'TiMidity port 3 '
```

In the configuration file for DOSBox, pass the client ID of the sequencer and the port to use on the `midiconfig` setting. The mididevice needs to be `default`. The syntax is `[client]:[port]`:

```
[midi]
mpu401      = intelligent
mididevice = default
midiconfig = 128:0
```

Gravis UltraSound (GUS)

The Gravis UltraSound cards were technically advanced soundcards with sample-based music synthesis ("wavetable") and hardware-mixing. DOSBox can emulate a Gravis UltraSound card for games that support it.

To enable GUS emulation, set the following options in your DOSBox configuration file:

IMPORTANT: The `ultradir` references a directory **within** DOSBox, not your local filesystem!

```
gus      = true
gusbase  = 240
gusirq   = 5
gusdma   = 3
ultradir = C:\ULTRASND
```

Depending on where you mount your `C:` drive (e.g. `~/DOS`), the `ULTRASND` directory needs to be placed inside it.

Installing GUS drivers

NOTE: Assumptions being made in this guide:

- The `C:` drive is mounted from `~/DOS`
- The `X:` drive is mounted from `~/Downloads/GUS Install` and contains the GUS setup files

IMPORTANT: Make sure you turn on GUS emulation in DOSBox **before** starting the setup procedure!

Preparations

GUS emulation needs the original install disks for the Gravis UltraSound, which can be downloaded [here](#).

Create an `autoexec.bat` at the root of DOSBox's `C:` drive:

```
touch ~/DOS/autoexec.bat
```


In DOSBox

Extract the contents into a directory and mount it as drive `X:` in DOSBox:

```
mount x ~/Downloads/GUS Install
```

Change directory to the `GUS410` directory and start the installer:

```
X:
cd X:\GUS410
INSTALL.EXE
```

Setup procedure:

1. Choose `Restore`, **NOT** `Install`
2. When asked what to restore, provide the glob pattern `*.*`
3. Keep the default target drive letter
4. Keep the default target directory
5. Start the installation process

Back at the main menu:

1. Choose `Install` (since it is restored, the installation should be quick)
2. Keep the defaults for the drive and directory
 - If it can't find Windows, provide `C:\ULTRASND\WINDOWS`
3. When the installation completes successfully exit out
 - Don't run Express or Custom Setup

Change directory to the `GUS411` directory and start the installer:

```
cd X:\GUS411
INSTALL.EXE
```

Repeat the installation steps above.

Testing

To test if setup was successful restart DOSBox, change into `C:\ULTRASND` and start `MIDIDEMO.BAT`.

If you hear music being played, the installation was successful.

Games with CD Audio

You can use [CBAE](#) to save some space with games that use CD audio tracks by compressing them.

`cbae` is a NodeJS package that is installed via `npm`. If you don't have NodeJS already installed:

```
pacman -S nodejs
```

Then install the `cbae` package **globally**:

```
npm i cbae --location=global
```

`cbae` takes `.bin/.cue` images as input and uses the information of the `.cue` file to determine what the CD audio tracks are.

To convert a `.bin/.cue` image:

```
cbae e KEEPER.cue -o ./ -enc OPUS:64 -p $(nproc)
```

This achieves the following:

- `e KEEPER.cue`: encodes CD audio tracks of the image `KEEPER.cue`
- `-o ./`: outputs the resulting files into a sub-directory of the current directory, e.g.:

```
KEEPER.bin
KEEPER.cue      <-- input file
KEEPER [e]      <-- sub-directory
├─ KEEPER.cue      <-- new .cue file by cbae
├─ KEEPER - Track 01.bin  <-- binary game data
├─ KEEPER - Track 02.opus <-- CD audio track
├─ KEEPER - Track 03.opus
├─ KEEPER - Track 04.opus
├─ KEEPER - Track 05.opus
├─ KEEPER - Track 06.opus
└─ KEEPER - Track 07.opus
```

- `-enc OPUS:64`: encodes audio tracks with Opus at 64 kbps (see `cbae --help` for available codecs)
- `-p $(nproc)`: specifies how many CPU cores are used for encoding (`$(nproc)` assigns the maximum number of cores available)

Mount the newly created `.cue` file with DOSBox's `imgmount` command, e.g. as the D: drive:

```
imgmount d ~/DOSGAMES/KEEPER [e]/KEEPER.cue -t cdrom
```

ScummVM

Dabble in some adventure games of yore with ScummVM:

```
pacman -S scummvm
```

OpenRCT2

A legitimate copy of the game is required to play, either bought from

- [Steam](#)
- [GOG.com](#)
- a retail CD version

Instructions on how to extract game assets and setting up the game can be found on the [official OpenRCT2 website](#) (`innoextract` is needed for extraction).

OpenRCT2 is an open-source re-implementation of RollerCoaster Tycoon 2 (RCT2), expanding the game with new features, fixing bugs and raising game limits.

NOTE: OpenRCT2 also supports RollerCoaster Tycoon 1 game files.

```
pacman -S openrct2 innoextract
```

For the development version install the AUR package:

```
yay -S openrct2-git
```

CorsixTH

A legitimate copy of the game is required to play, either bought from

- [GOG.com](https://www.gog.com)
- a retail CD version

Instructions on how to extract game assets and setting up the game can be found on the [CorsixTH GitHub Wiki](#) (`innoextract` is needed for extraction).

CorsixTH aims to re-implement the game engine of Theme Hospital, and be able to load the original game data files.

```
yay -S corsix-th-git innoextract
```

ioquake3

A legitimate copy of the game is required to play, either bought from

- [Steam](#)
- [GOG.com](#)
- a retail CD version

Instructions on how to extract game assets and setting up the game can be found on the [ioquake3 Website](#).

ioquake3 is a free software first person shooter engine based on the *Quake 3: Arena* and *Quake 3: Team Arena* source code.

```
yay -S ioquake3
```

Copy the `pak0.pk3` file from your copy of the game:

```
# single user
cp pak0.pk3 ~/.q3a/baseq3/

# system-wide
cp pak0.pk3 /opt/quake3/baseq3/
```

DXX-Rebirth

Legitimate copies of the games are required to play, either bought from

- GOG.com
 - [Descent 1](#)
 - [Descent 2](#)
- the retail CD versions

Further info can be found on the [DXX-Rebirth Website](#) (`innoextract` is needed for extraction).

DXX-Rebirth is a source port of the Descent and Descent 2 Engines so you won't need DOSBox to play the games. Additionally, it offers OpenGL graphics and effects, advanced Multiplayer, many improvements and new features.

```
yay -S dlx-rebirth d2x-rebirth innoextract
```

Copy the Descent 1 game files:

```
# single user
cp descent.(hog|pig) ~/.dlx-rebirth/

# system-wide
cp descent.(hog|pig) /usr/share/dlx-rebirth/
```

Copy the Descent 2 game files:

```
# single user
cp *. (ham|hog|pig|s11|s22) ~/.d2x-rebirth/

# system-wide
cp *. (ham|hog|pig|s11|s22) /usr/share/d2x-rebirth/
```

Minecraft

The official Minecraft Launcher package maintained by Microsoft:

```
yay -S minecraft-launcher
```


SimCity 3000 Unlimited

Widescreen hack

SimCity 3000 does only support a set amount of resolutions, widescreen resolutions are not natively supported. You can work around that by editing the main executable file in a hex editor.

Open the main executable file `SC3U.exe` in a hex editor and search for the following byte sequence:

```
8b 4c 24 04 8b 44 24 08 53
```

Overwrite the first four bytes with:

```
c2 08 00 90
```

Next search for the byte sequence:

```
8b 4c 24 04 8b 54 24 08 81 f9
```

Overwrite the first four bytes with:

```
c2 08 00 90
```

The game will now allow much higher resolutions. Keep in mind, however, that the interface will not scale with the resolution and on really high resolutions the UI of the game might be displayed *really* small and might get hard to read.

Scrolling speed fix

On modern systems, SimCity 3000 exhibits a scrolling bug in which the game scrolls way too fast.

To fix this, a mod can be installed, called cnc-ddraw. It limits the game's tickrate and adds a couple more neat features.

Download cnc-ddraw from the project's [GitHub releases page](#). Drop the files `ddraw.dll` and `ddraw.ini` into the installation directory of the game, next to the `SC3U.exe` file. Then, start the game with the environment variable `WINEDLLOVERRIDES=ddraw=n,b`. If you have the game on Steam, this can be easily achieved by opening the game's settings and adding the following line to the launch options:

```
WINEDLLOVERRIDES=ddraw=n,b %command%
```

Shaders for upscaling

cnc-ddraw supports scaling with GLSL shaders. Download them from the [libretro GitHub](#) (Code → Download ZIP) and extract it into a sub-directory in the same folder as the `ddraw.dll` file you copied earlier, e.g. `~/steam/steam/steamapps/common/SimCity 3000 Unlimited/Apps/Shaders`.

Then edit `ddraw.ini` and set the following options (shortened, Ctrl-F the setting keys):

```
[ddraw]
shader=Shaders/interpolation/pixellate.glsl
renderer=opengl
```

UT2004 (Atari DVD Release Version)

Install from the DVD. Navigate to the location the DVD was mounted at and run:

```
sudo sh ./linux-installer.sh
```

Follow the installation steps.

After installation completes **do not run the game immediately after install**. Patch first!

Patch to latest version

The patch can be downloaded here: ↓ [UT2004 Mega Pack Linux + LinuxPatch 3369.2 - utzone.de](https://utzone.de/ut2004-mega-pack-linux-linuxpatch-3369.2/)

Extract contents from the archive and copy them to the install location, overwriting all the files:

NOTE: If `cp` is asking for confirmation on every file, it is likely there is an alias to `cp -i`. Prepend a `\` before the `cp` command to temporarily ignore this alias. Alternatively, `unalias cp` to undefine the alias.

```
tar -xf ut2004megapack-linux -C /tmp/  
cp -R /tmp/UT2004MegaPack/* /usr/local/games/ut2004
```

Launch in 64-bit

By default, the `ut2004` command will launch the 32-bit version of the game. To make it launch the 64-bit version, edit the start script `/usr/local/games/ut2004/ut2004` and change **line 49** from

```
exec "./ut2004-bin" $*
```

to

```
exec "./ut2004-bin-linux-amd64" $*
```

Missing libraries

The game tries to load quite some old libraries, e.g. it will complain about `libSDL-1.2.so.0` or `libstdc++.so.5` missing.

In the case of `libSDL-1.2.so.0` simply install the SDL 1.2 compatability package:

```
pacman -S sdl12-compat
```

Then, remove the `libSDL-1.2.so.0` the game came with and put a symbolic link to your system's `libSDL-1.2.so.0`

```
sudo rm /usr/local/games/ut2004/System/libSDL-1.2.so.0
sudo ln -sf /usr/lib/libSDL-1.2.so.0 /usr/local/games/ut2004/System/libSDL-1.2.so.0
```

To fix the error about `libstdc++.so.5`, an AUR package with the files is available:

```
yay -S libstdc++5
```

No sound

UT2004 expects there to be a `/dev/dsp` sound device to access the sound card directly. This goes back to OSS (Open Sound System) which has long been deprecated in favor of ALSA and contemporaries.

This can easily be fixed in one of two ways:

1. Further edit the start script of `ut2004`, prepending the execution of the game binary with `padsp` to route all audio through PulseAudio:

```
exec padsp "./ut2004-bin-linux-amd64" $*
```

2. Edit your user's UT2004 config file `~/.ut2004/System/UT2004.ini`, go to the section `[ALAudio.ALAudioSubsystem]` and change the value of `UseDefaultDriver=True` to `UseDefaultDriver=False`

Screen resolution

Some higher resolutions might not show up in the game's configuration screen. To set a resolution manually, edit your user's UT2004 config file, go to section `[SDLDrv.SDLClient]` and set the following parameters, e.g.:

ATTENTION: The config file contains two sections for various graphical settings, one for Windows and one for Linux. The one for Linux comes after the one for Windows.

```
[SDLDrv.SDLClient]
FullscreenViewportX=2560
FullscreenViewportY=1440
```

Proper wide-screen support

A mod is available that expands UT2004's wide-screen support: [foxWSFix-UT2k4](#)