

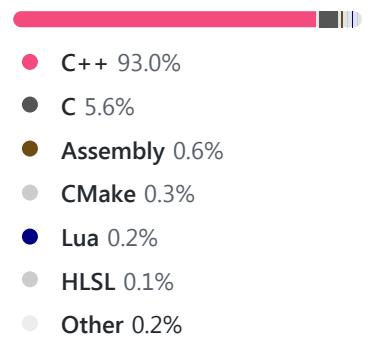
miami ▾

[Go to file](#)[Add file ▾](#)[Code ▾](#)This branch is 1758 commits ahead, 405 commits behind master. [Contribute ▾](#)**About**

GTA III, Vice City

[Readme](#)
 **Sergeanur fix** ... ✓ 16 hours ago ⏲ 4,334

 .github	fix cmake build.	6 days ago
 .vscode	.vscode/settings: fix include path	9 months ago
 cmake	Cmake version for hash	5 months ago
 codewarrior	swap out codewarrior reVC.mcp with ...	5 days ago
 gamefiles	Add russian support	4 months ago
 src	fix	16 hours ago
 utils/gxt	Sync aspect ratio setting with master	5 months ago
 vendor	update librw	4 months ago
 .clang-format	Add basic clang format file	7 months ago
 .gitattributes	Add .gitattributes for better language...	9 months ago
 .gitignore	swap out codewarrior reVC.mcp with ...	5 days ago
 .gitmodules	Move stuff to vendor	11 months ago
 CMakeLists.txt	Cmake version for hash	5 months ago
 CODING_STYLE....	Separate Coding Style out into its ow...	9 months ago
 README.md	reVC: Update Linux 64bit build downl...	5 days ago
 conanfile.py	Port conan to miami	6 months ago
 logo.png	reVC logo	5 months ago
 logo.svg	reVC logo	5 months ago
 premake-vs2015...	improvements (?) to premakefile	14 months ago
 premake-vs2017...	improvements (?) to premakefile	14 months ago
 premake-vs2019...	improvements (?) to premakefile	14 months ago
 premake5.exe	premake: add startrproject; add lto op...	6 months ago

Releases 2
 **First Release ...** Latest on 14 Feb
[+ 1 release](#)**Contributors** 36[+ 25 contributors](#)**Languages**

	premake5.lua	fix tabs in premake	5 days ago
	premake5Linux	premake: add startrproject; add lto op...	6 months ago
	printHash.bat	Remove version text, add toggleable ...	5 months ago
	printHash.sh	More compatible she-bang line	5 months ago

☰ README.md



Intro

In this repository you'll find the fully reversed source code for GTA III ([master branch](#)) and GTA VC ([miami branch](#)).

It has been tested and works on Windows, Linux and FreeBSD, on x86, amd64, arm and arm64.

Rendering is handled either by original RenderWare (D3D8) or the reimplementation [librw](#) (D3D9, OpenGL 2.1 or above, OpenGL ES 2.0 or above).

Audio is done with MSS (using dlls from original GTA) or OpenAL.

We cannot build for PS2 or Xbox yet. If you're interested in doing so, get in touch with us.

Installation

- reVC requires game assets to work, so you **must** own [a copy of GTA Vice City](#).
- Build reVC or download the latest build:
 - [Windows D3D9 MSS 32bit](#)
 - [Windows D3D9 64bit](#)
 - [Windows OpenGL 64bit](#)
 - [Linux 64bit](#)
 - [MacOS 64bit](#)

- Extract the downloaded zip over your GTA VC directory and run reVC. The zip includes the gamefiles and in case of OpenAL the required dlls.

Screenshots





Improvements

We have implemented a number of changes and improvements to the original game. They can be configured in `core/config.h`. Some of them can be toggled at runtime, some cannot.

- Fixed a lot of smaller and bigger bugs
- User files (saves and settings) stored in GTA root directory
- Settings stored in `reVC.ini` file instead of `gta_vc.set`
- Debug menu to do and change various things (Ctrl-M to open)
- Debug camera (Ctrl-B to toggle)
- Rotatable camera
- XInput controller support (Windows)
- No loading screens between islands ("map memory usage" in menu)
- Rendering
 - Widescreen support (properly scaled HUD, Menu and FOV)
 - PS2 MatFX (vehicle reflections)
 - PS2 alpha test (better rendering of transparency)
 - Xbox vehicle rendering
 - Xbox world lightmap rendering (needs Xbox map)
 - Xbox ped rim light
 - Xbox screen rain droplets
 - More customizable colourfilter
- Menu
 - More options
 - Controller configuration menu
 - ...
- Can load DFFs and TXDs from other platforms, possibly with a performance penalty

- ...

To-Do

The following things would be nice to have/do:

- Fix physics for high FPS
- Improve performance on lower end devices, especially the OpenGL layer on the Raspberry Pi (if you have experience with this, please get in touch)
- [PS2 port](#)
- Xbox port (not quite as important)
- reverse remaining unused/debug functions
- compare CodeWarrior build with original binary for more accurate code (very tedious)

Modding

Asset modifications (models, texture, handling, script, ...) should work the same way as with original GTA for the most part.

Mods that make changes to the code (dll/asi, CLEO, limit adjusters) will *not* work. Some things these mods do are already implemented in re3 (much of SkyGFX, GInput, SilentPatch, Widescreen fix), others can easily be achieved (increasing limis, see `config.h`), others will simply have to be rewritten and integrated into the code directly. Sorry for the inconvenience.

Building from Source

When using premake, you may want to point `GTA_VC_RE_DIR` environment variable to GTA Vice City root folder if you want the executable to be moved there via post-build script.

Clone the repository with `git clone --recursive -b miami https://github.com/GTAmodding/re3.git reVC`. Then `cd reVC` into the cloned repository.

- ▶ Linux Premake
- ▶ Linux Conan
- ▶ FreeBSD
- ▶ Windows

i premake has an `--lto` option if you want the project to be compiled with Link Time Optimization.

i There are various settings in [config.h](#), you may want to take a look there.

i reVC uses completely homebrew RenderWare-replacement rendering engine; [librw](#). librw comes as submodule of re3, but you also can use LIBRW environment variable to specify path to your own librw.

If you feel the need, you can also use CodeWarrior 7 to compile reVC using the supplied codewarrior/reVC.mcp project - this requires the original RW34 libraries, and the DX8 SDK. The build is unstable compared to the MSVC builds though, and is mostly meant to serve as a reference.

Contributing

As long as it's not linux/cross-platform skeleton/compatibility layer, all of the code on the repo that's not behind a preprocessor condition (like `FIX_BUGS`) are **completely** reversed code from original binaries.

We **don't** accept custom codes, as long as it's not wrapped via preprocessor conditions, or it's linux/cross-platform skeleton/compatibility layer.

We accept only these kinds of PRs;

- A new feature that exists in at least one of the GTAs (if it wasn't in III/VC then it doesn't have to be decompilation)
- Game, UI or UX bug fixes (if it's a fix to original code, it should be behind `FIX_BUGS`)
- Platform-specific and/or unused code that's not been reversed yet
- Makes reversed code more understandable/accurate, as in "which code would produce this assembly".
- A new cross-platform skeleton/compatibility layer, or improvements to them
- Translation fixes, for languages original game supported
- Code that increase maintainability

We have a [Coding Style](#) document that isn't followed or enforced very well.

Do not use features from C++11 or later.

History

re3 was started sometime in the spring of 2018, initially as a way to test reversed collision and physics code inside the game. This was done by replacing single functions of the game with their reversed counterparts using a dll.

After a bit of work the project lay dormant for about a year and was picked up again and pushed to github in May 2019. At the time I (aap) had reversed around 10k lines of code and estimated the final game to have around 200-250k. Others quickly joined the effort (Fire_Head, shfil, erorcun and Nick007J in time order, and Serge a bit later) and we made very quick progress throughout the summer of 2019 after which the pace slowed down a bit.

Due to everyone staying home during the start of the Corona pandemic everybody had a lot of time to work on re3 again and we finally got a standalone exe in April 2020 (around 180k lines by then).

After the initial excitement and fixing and polishing the code further, reVC was started in early May 2020 by starting from re3 code, not by starting from scratch replacing functions with a dll. After a few months of mostly steady progress we considered reVC finished in December.

Since then we have started reLCS, which is currently work in progress.

License

We don't feel like we're in a position to give this code a license. The code should only be used for educational, documentation and modding purposes.

We do not encourage piracy or commercial use.

Please keep derivate work open source and give proper credit.