Dev Diary 6

Salutations!

We are once again pleased to present you with our latest Dev Diary! While we haven't published a dev diary in a while, that isn't for lack of progress. We've completed a bunch of background work that, while mostly lacking pretty visuals, has laid a foundation for the game. First, we shall look at the prime focus of this Dev Diary:

User Interface (UI)

Victoria II's UI is a messy mix of customisable GUI and GFX text files and hard-coded backend behaviour. It is very complex, requiring a lot of work to interpret and convert into UI elements on your screen, and very restrictive, with hardcoded constraints on what can be added or removed.

Nevertheless, many mods include UI changes, for example HPM's political/social reforms menu and TGC's extra province building slot. In keeping with our goal to support any mod that Victoria II supports, we have implemented full GUI and GFX file loading and conversion to Godot UI scenes.

This system generates and combines together the component pieces of a tree of UI elements, including loading textures, positioning elements relative to their parents and anchor points, and providing custom control functions for special elements like pie charts and masked flags. However, rather than being defined in text files with the layout, the UI's behaviour is hardcoded in Victoria II. This means that for every UI panel we must write a control script that handles everything from setting label text to handling button presses, from updating progress bars to generating and arranging icon lists (e.g. cores and modifiers) - in short, all game data outputs, player inputs, and general UI state/visibility handling.

In these screenshots you can see two Victoria II UI panels loaded into OpenVic:

- the province overview panel, with a significant portion of its behaviour script implemented, including names, terrain image, liferating bar, controller flag, RGO and crime icons, total population, POP type and culture pie charts, and a working close button.
- the topbar, with its behaviour script covering the player country's flag and name, the current date and game speed/pause icon, and working pause/unpause and speed control buttons.

These UI panels have acted as “pipe-cleaners”, helping us build and iron out the UI generation system and behaviour script API around them. We will now be shifting focus to continuing the implementation of Victoria II’s UI, in particular anything which displays the game state data we had been focusing on loading up until now.

MAPMODES

Separate from the GUI/GFX file based UI work, we have also made some improvements to the map and mapmodes:

The terrain map has better colour tinting (previously it was tinted with a uniform shade of green/blue across all land/water), making terrain type transitions smoother, providing more variety across geographical regions, and, in the case of water tinting, improving the look of coastlines.

Support has been added for stripped province shading, currently used by the culture and religion mapmodes to display second largest cultures/religions which still represent over a third of the province’s population.

Dataloading

Over the last few months there's been a continuous effort to load Victoria II files (and mods too!) in a forward-facing and moddable manner. At this time, we're on the tail-end of the dataloading chapter of OpenVic's development, with only a few minor files left to load. It's been a massive amount of work and will undergird the entire rest of the game, with everything that is to come building on top of this carefully crafted foundation. For those interested, here's a high level list of what OpenVic currently loads from Victoria II defines files:

The Map

- Provinces
- Province history (owner, RGO, buildings)
- States & Regions
- Terrain types
- Adjacencies (Normal, Impassable, Canals & Straits)
- Province Model Positions

POPs

- POP Types
- Cultures
- Religions
- POP history

Countries

- General history info (primary/accepted cultures, capital province, political parties)
- Starting Alliances, Subjects, & Wars
- Starting Armies & Navies
- National Foci

Economy

- Buildings
- Goods
- Production Types (Factories)
- Technologies
- Crimes

Politics

- Ideologies
- Issues & Reforms
- Government Types
- National Values

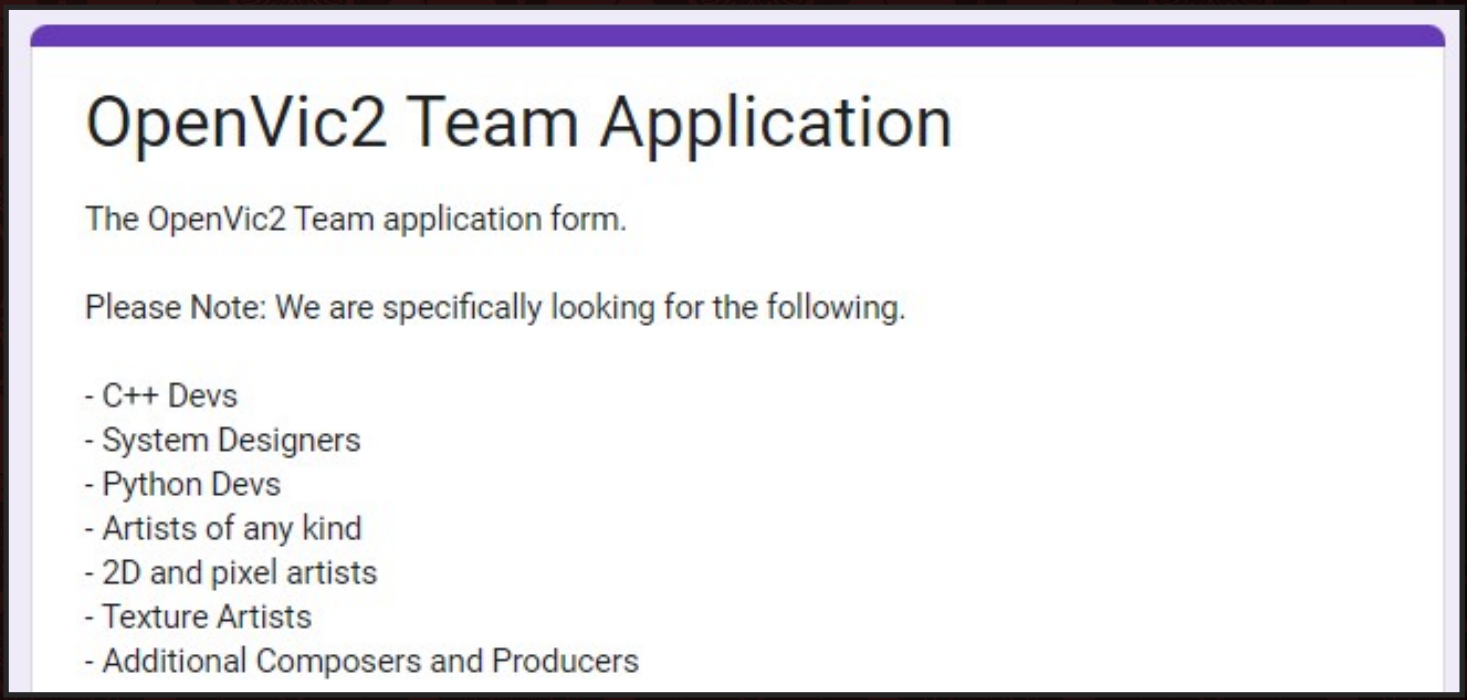
Military

- Units (Infantry, Guards, Hussars, Cruisers, etc)
- Leader Traits
- Casus Belli

Miscellaneous

- Lua Defines
- Modifiers
- UI (GUI + GFX file loading)
- Bookmarks/Alternate Start Dates

With so much work now complete, we're in a strong position to start Dev Cycle 5, which should bring plenty of pretty pictures for us to show off, as well as time to pick up some more team members to help out with the UI and simulation - so if you're interested and familiar with C++ or Godot, make sure you apply!

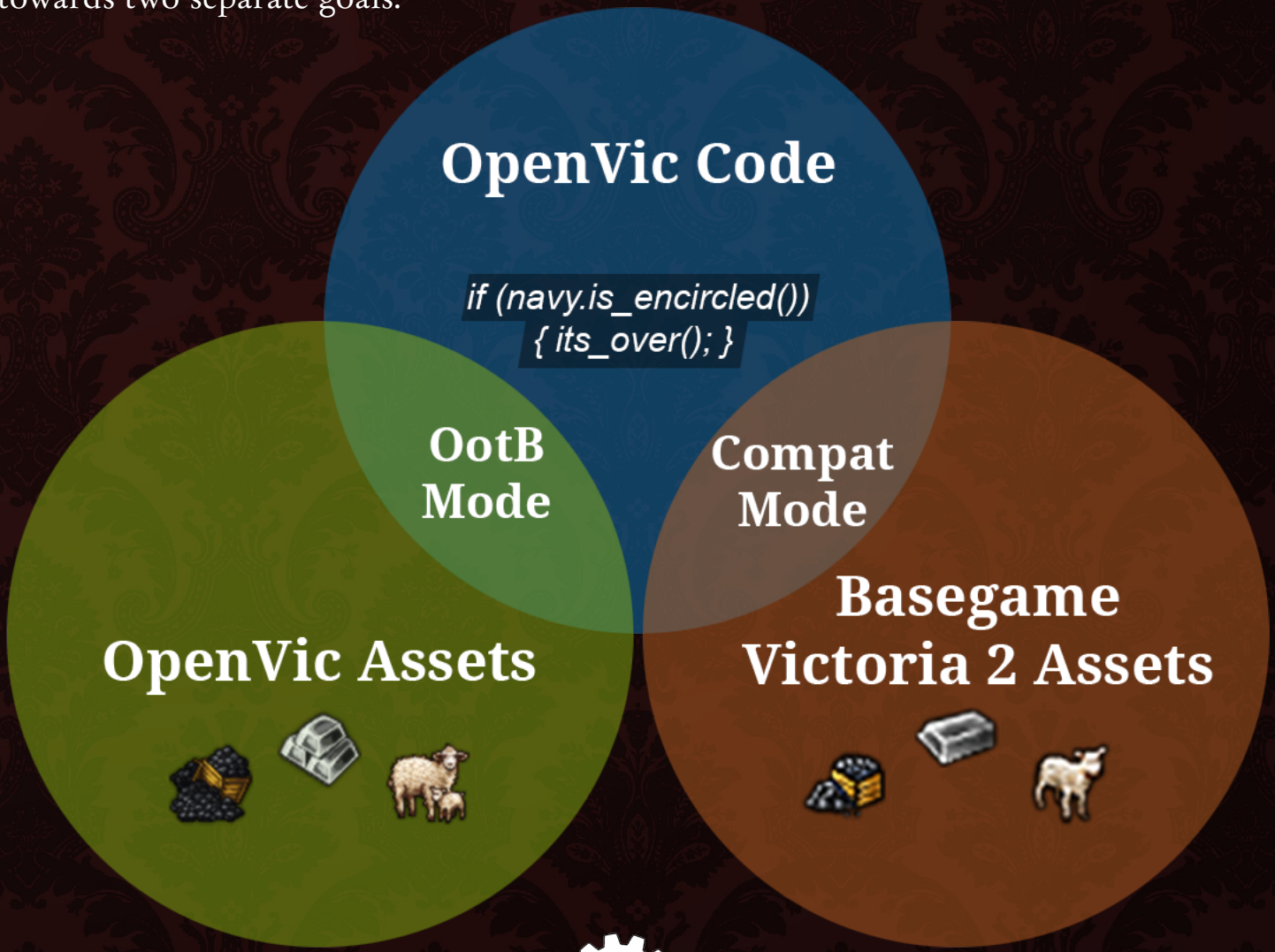


TESTING

Another area we have put a lot of work into is testing, both manual and automated. We've added basic automated testing, and it's been a lifesaver when checking our dataloading code against Victoria II & mod files, replacing manually checking our dataloading code against what we expect from Victoria II's assets. We hope soon to utilise a dedicated, redistributable OpenVic testing mod/set of defines, rather than local copies of the game, which will help with automatic cross-platform tests.

PROJECT GOALS

Moving on, I think it's important to reiterate our main goals for this project as it gives you a good indicator of what we're aiming for and what we are likely to be working on in the near future. As we continue to work on the project we have found the need to readjust and specify exactly what we intend to do - both for your sake and ours. Over the past few months we have fine-tuned and focused our scope towards two separate goals.



OpenVic Project Scope:

Core Objectives:

Improved General stability: The vanilla OpenVic should not crash or completely freeze at any time.

Improved Multiplayer Stability: The vanilla OpenVic should ideally not desynchronize under any circumstance. Robust hot joining and resyncing features should exist.

Centralised Match-Making: Players will be able to dynamically connect to each other's games over the internet without having to input each other's IP manually or use port-forwarding.

Performance Parity or Better: OpenVic should match or exceed the running speed of Victoria 2 without requiring a high end machine. People who can run Victoria 2 now should be able to run OpenVic at the same speed or faster.

Improved Modding System/Tools: The modding tools in OpenVic should:

- Be as extensive as possible, with logic operators, loops, variables, improved scoping, and other aspects missing from Victoria 2.
- Have as minimal impact on stability as possible.
- Have as low a bar to entry as possible, programming language skills should be optional, and the syntax should be easy to read and pick up as a new user.
- For more see **"OpenVic Modding Suggestions"** doccument.

Modernization: OpenVic should be able to:

- Run on all modern machines,
- Mainstream operating systems,
- Support any screen resolution within reason,
- Support options for modern graphical fidelity (such as higher resolution map files).

Backwards compatibility: OpenVic should have inbuilt support for all Victoria 2 mods, where Victoria 2 mods can be loaded without any changes.

High Degree of Quality of Life: OpenVic should have a high degree of quality of life improvements such as a better peace deal system, a tool to transfer provinces and states, vision treaties separate to alliances, a built in war analyzer, and of course a "macro builder" of some kind for armies as well as many more.

High Degree of Design Documentation: OpenVic should be accompanied by a resource that details all of the game's systems in detail and accurately. Gameplay equations for things like the economy should be easily accessible. Equally, the code itself should be highly documented, Each element of internal functionality within the software should be justified by supporting documentation.

OpenSource: The project must be licensed under GPL3 or another permissible Open Source License. Original Assets made by the OpenVic team or otherwise used in the project are not under an Open Source licence.

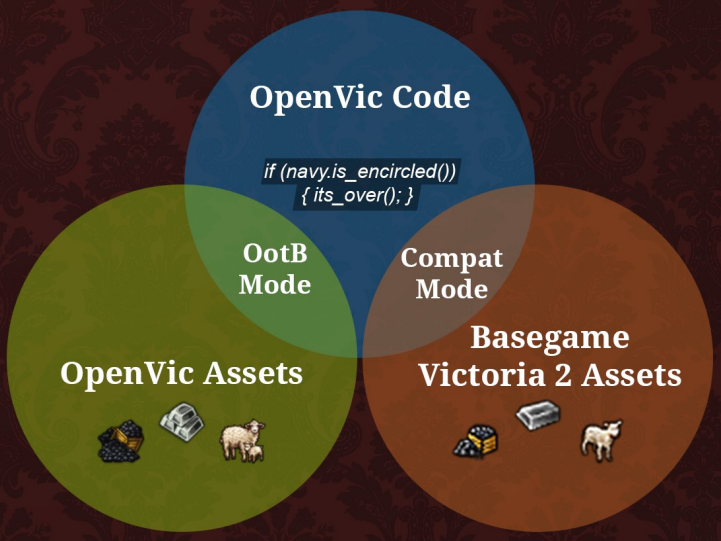
Greater User Customization Options: Users should be able to customise their gameplay experience to their preferences. This should include being able to mechanically enforce certain multiplayer rules such as multi-province retreating.

File Loading and "Out of the Box" mode: OpenVic will be able to load the Victoria 2 files for an authentic 1:1 Victoria 2 emulation, with none of the original paradox files (assets and defines included among others) being shipped with the game. However the game should feature its own original assets for use in an "Out of the Box" mode where the game does not require a Victoria 2 installation or file loading from external sources to run.

Key game modes:

Compatibility mode/File Loading Mode:
Compatibility mode will load files from a local legally obtained and installed copy of Victoria 2. In this mode OpenVic will replace the Victoria 2 executable file primarily with nothing past that.

Out of the Box Mode:
In Out of the Box (OotB) Mode all assets, including events, are completely custom made and not derived from Victoria 2. The assets will however still be inspired by Victoria 2. In OotB mode OpenVic replaces all Victoria 2 files. These assets should be of a high quality, exceeding that of Victoria 2 in depth, breadth, and fidelity.



Extended Objectives:

Modding suite: OpenVic may have an associated modding suite to enhance the modding experience and make it easier to make mods, which contributes to fulfilling the core improved modding system objective.

Improved AI: OpenVic may have an improved AI to Victoria 2 that incorporates the many years of tactical knowledge accumulated by the community since Victoria 2's last update.

Resource and Shader Packs: OpenVic may have support for visual packs that affect the graphical fidelity of the game.

Extended Features: OpenVic may feature extended features - such as multi-monitor and pop out window support for certain in game menus, an improved ledger, even dynamic provinces that can appear/move/change shape/add/delete during the game. These features are mostly community sourced and can be found in the **"Suggested diversions from Victoria 2"** doccument.

Compatibility mode will load files from a **local and legal copy** of Victoria II. Through this, OpenVic will provide an experience as close as possible to the original Victoria II game. Compatibility mode will also have the ability to load Victoria II mods, just the same as the original game. To make it abundantly clear, every mod that works in Victoria II will work in OpenVic no changes, no compatibility patches, just plug & play.

Out of the Box mode will, on the other hand, use completely original assets and events inspired by the original Victoria II experience. The major difference is that they are not Victoria II-derived, and so will deviate significantly in specifics, while staying similar overall due to being based off of the same historical record. We aim to exceed this experience with higher quality and fidelity assets, modernising the Victoria experience while staying true to the original's look and feel.

Initially, the vast majority of our efforts will be focused on Compatibility Mode. This is because attempting to juggle both goals at the same time is a good way to get neither done.

However, once we get to a stage where Compatibility Mode is functional, we can then bring the knowledge and experience gained from implementing it to confidently develop the best game we can for Out of the Box Mode, bringing on further improvements to the Victoria experience which we previously could only have dreamt of. Here, we will be very excited to showcase our improved custom graphics and artwork, as well as explore new mechanics and algorithms. All very lofty stuff.

On a more sombre note, our community manager, Catylist, has taken the difficult decision to step back from the project. As one of its leaders, Catylist has been with us since the project's inception and has been instrumental in building the organisation and infrastructure setup which will see us right through to completion. Thanks to him we have an active and thriving community whose support we are incredibly thankful for.

As ever, thanks again for following our motley project and see you in the next one!

The OpenVic Team