Phobos

The Phobos Contributors

Mar 18, 2021

PROJECT INFO

CHAPTER

ONE

BUILD TYPES

There are three main types of Phobos builds:

- *stable builds* - those are numbered like your regular versions (something close to semantic versioning, e.g. version 1.2.3 for example) and ideally should contain no bugs, therefore are safe to use in mods;
- *development builds* - those are the builds which contain functionality that needs to be tested. They are numbered plainly starting from 0 and incrementing the number on each release. Mod authors still can include those versions with their mods if they want latest features, though we can't guarantee lack of bugs;
- *nightly builds* - bleeding edge versions which can include prototypes, proofs of concepts, scrapped features etc., in other words - we can't guarantee anything in those builds and they absolutely should NOT be used in mod releases and should only be used to help with development and testing.

1.1 Disabling development build warning

DISCLAIMER: We understand that everyone wants to try and use the new features as soon as they're released, but we can't do all the testing ourselves, so we only test the functionality on a basic level. We ask everyone who uses the new development build first to **test the new changes in every possible way first before disabling the development build warning** and proceeding to include the build in your mod release. This would allow us to concentrate on implementing the actual features, which is the most complex task.

You can hide the warning by specifying the build number after `-b=` as a command line argument (for example, `-b=1` would hide the warning for development build #1 of Phobos).

CHAPTER

TWO

SAVED GAMES FILTERING

Phobos fully supports saving and loading thanks to prototype code from publicly released Ares 0.A source and it implements it's own filtering which shouldn't conflict with Ares save filtering. Save games between different versions are incompatible due to changes to Phobos extension classes which are present in almost every build release. The filtering mechanism, hovewer, doesn't apply to nightly versions - those use latest development build number on which this nigthly is based on. While different nightly version saves may be listed, they are most likely incompatible in case there were changes to extension class fields.

CHAPTER

THREE

COMPATIBILITY

While Phobos is standalone, it is designed to be used alongside Ares and CnCNet5 spawner. Adding new features or improving existing ones is done with compatibility with those in mind.

While we would also like to support HAres we can't guarantee compatibility with it due to the separation of it's userbase and developers from international community. We welcome any help on the matter though!

CHAPTER

FOUR

CONTRIBUTING

This page describes how to help or contribute to Phobos and all the different ways to do so.

Engine modding is a complicated process which is pretty hard to pull off, but there are also easier parts which don't require mastering the art of reverse-engineering or becoming a dank magician in C++.

4.1 Research

You can observe how the stuff works by using the engine and note which other stuff infliuences the behavior, but sooner or later you would want to see the innards of that. This is usually done using such tools as disassemblers/decompilers (IDA, Ghidra) to decipher what is written in the binary (`gamemd.exe` in case of the binary) and debuggers (Cheat Engine's debugger is pretty good for that) to trace how the binary works.

Note: Assembly language and C++ knowledge, understanding of computer architecture, memory structure, OOP and compiler theory would certainly help.

4.2 Code development

When you found out how the engine works and where you need to extend the logic you'd need to develop the code to achieve what you want. This is done by declaring a *hook* - some code which would be executed after the program execution reaches the certain address in binary. All the development is done in C++ using YRpp (which provides a way to interact with YR code and inject code using Syringe) and usually Visual Studio 2017/2019 or newer.

Note: You'd benefit from C++ experience, knowledge of programming patterns, common techniques etc. Basic assembly knowledge would help to correctly write the interaction with the memory where you hook at. Basic understanding of Git and GitHub is also needed.

4.3 Testing

This is a job that any modder can do. Look at a new feature or a change, try to think of all possible cases when it can work differently, try to think of any possible logic flaws, edge cases, unforeseen interactions or conditions etc., then test it according to your thoughts. Any bugs should be reported to issues section of this repo, if possible.

Warning: **General stability** can only be achieved by extensive playtesting of new changes, both offline and online. Most modders have beta testing teams, so please, if you want the extension to be stable - contribute to that by having your testers play with the new features! Also the check-list below can help you identify issues quicker.

4.3.1 Testing check-list

- **All possible valid use cases covered**. Try check all of the valid feature use cases you can think of and verify that they work as intended with the feature.
- **Correct saving and loading**. Most of the additions like new INI tags require storing them in saved object info. Sometimes this is not done correctly, especially on complex stuff (like radiation types). Please, ensure all the improvements work **identically** before and after being saved and loaded (on the same version of Phobos, of course).
- **Interaction with other features**. Try to use the feature chained or interacting with other features from vanilla or other libs (for example, mind control removal warhead initially was crashing when trying to remove mind control from a permanently MC'ed unit).
- **Overlapping features not working correctly** (including those from third-party libs like Ares, HAres, CnCNet spawner DLL). Think of what features' code could overlap (in a technical sense; means they modify the same code) with what you're currently testing. Due to the nature of the project some features from other libs could happen to not work as expected if they are overlapping (for example, when implementing mass selection filtering Ares' `GroupAs` was initially broken and units using it weren't being type selected properly).
- **Edge cases**. Those are the cases of some specific cases usually induced by some extreme parameter values (for example, vanilla game crashes on zero-size `PreviewPack` instead of not drawing it).
- **Corner cases**. Those are similiar to edge cases but are hard to reproduce and are usually induced by a combination of extreme parameter values.

Note: Knowledge on how to mod YR is the only requirement to do this, but having an inquisitive mind, being attentive to details would also help.

4.4 Writing docs

No explanation needed. If you fully understand how some stuff in Phobos works you can help by writing a detailed description in this wiki, or you can just improve the pieces of wiki you think are not detailed enough.

Note: OK English grammar and understanding of docs structure would be enough.

4.5 Providing media to showcase features

Those would be used in docs and with a link to the respective mod as a bonus for the mod author. To record gifs you can use such apps as, for example, GifCam.

Note: Please, provide screenshots, gifs and videos in their natural size and without excess stuff or length.

CHAPTER

FIVE

LICENSE

CHAPTER

SIX

NEW / ENHANCED LOGICS

This page describes all the engine features that are either new and introduced by Phobos or significantly extended or expanded.

6.1 Buildings

6.1.1 Extended building upgrades logic

Upgrading own and allied Power Plants in CnC: Final War

- Building upgrades now can be placed on own buildings, on allied buildings and/or on enemy buildings. These three owners can be specified via a new tag, comma-separated. When upgrade is placed on building, it automatically changes it's owner to match the building's owner.
- One upgrade can now be applied to multiple buildings via a new tag, comma-separated.
  - Currently Ares-introduced build limit for building upgrades doesn't work with this feature. This may change in future.

In `rulesmd.ini`:

```
[UPGRADENAME]       ; BuildingType
PowersUp.Owner=Self ; list of owners (Self, Ally and/or Enemy)
PowersUp.Buildings= ; list of BuildingTypes
```

6.2 Warheads

6.2.1 Generate credits on impact

TransactMoney used in Rise of the East mod

- Warheads can now give credits to its owner at impact.

In `rulesmd.ini`:

```
[SOMEWARHEAD]   ; Warhead
TransactMoney=0 ; integer - credits added or subtracted
```

6.2.2 Reveal map for owner on impact

SpySat=yes on [NUKE] warhead reveals the map when nuclear missile detonates

- Warheads can now reveal the entire map on impact.
- Reveal only applies to the owner of the warhead.

In `rulesmd.ini`:

```
[SOMEWARHEAD] ; Warhead
SpySat=no     ; boolean
```

6.2.3 Shroud map for enemies on impact

- Warheads can now shroud the entire map on impact.
- Shroud only applies to enemies of the warhead owner.

In `rulesmd.ini`:

```
[SOMEWARHEAD] ; Warhead
BigGap=no     ; boolean
```

CHAPTER

SEVEN

FIXED / IMPROVED LOGICS

This page describes all ingame logics that are fixed or improved in Phobos without adding anything significant.

7.1 Bugfixes and miscellanous

- Fixed the bug when deploying mindcontrolled vehicle into a building permanently trasferred the control to the house which mindcontrolled it.
  - Currently doesn't work with superweapons attached to the deployed building or similiar logics.
- SHP debris shadows now respect the `Shadow` tag.

7.2 Vehicles

7.2.1 Customizable disk laser radius

- You can now set disk laser animation radius using a new tag.

In `rulesmd.ini`:

```
[SOMEWEAPON]           ; WeaponType
DiskLaser.Radius=38.2 ; floating point value
                       ; 38.2 is roughly the default saucer disk radius
```

7.2.2 Remember target when deployed

Vehicle keeping target after deployed to building in C&C: Reloaded

- Vehicle to building deployers can now keep their target when deploying.

In `rulesmd.ini`:

```
[SOMEVEHICLE]              ; TechnoType
Deployed.RememberTarget=no ; boolean
```

CHAPTER

EIGHT

USER INTERFACE

This page lists all user interface additions, changes, fixes that are implemented in Phobos.

8.1 Audio

- You can now specify which soundtrack themes would play on win or lose.

In `rulesmd.ini`:

```
[SOMESIDE]              ; Side
IngameScore.WinTheme=   ; soundtrack theme ID
IngameScore.LoseTheme=  ; soundtrack theme ID
```

8.2 Bugfixes and miscellanous

- Enabled ability to load full-color non-paletted PCX graphics of any bitness. This applies to every single PCX file that is loaded, including the Ares-supported PCX files.
- You can specify custom `gamemd.exe` icon via `-icon` command line argument followed by absolute or relative path to an *.ico file (f. ex. `gamemd.exe -icon Resources/clienticon.ico`).
  - Currently doesn't work with CnC-DDraw as it overrides the icon too.

8.3 Battle screen UI/UX

8.3.1 Low priority for box selection

Harvesters not selected together with battle units in Rise of the East mod

- You can now set lower priority for an ingame object (currently has effect on units mostly), which means it will be excluded from box selection if there's at least one normal priority unit in the box. Otherwise it would be selected as normal. Works with box+type selecting (type select hotkey + drag) and regular box selecting. Box shift-selection adds low-priority units to the group if there are no normal priority units among the appended ones.

In `rulesmd.ini`:

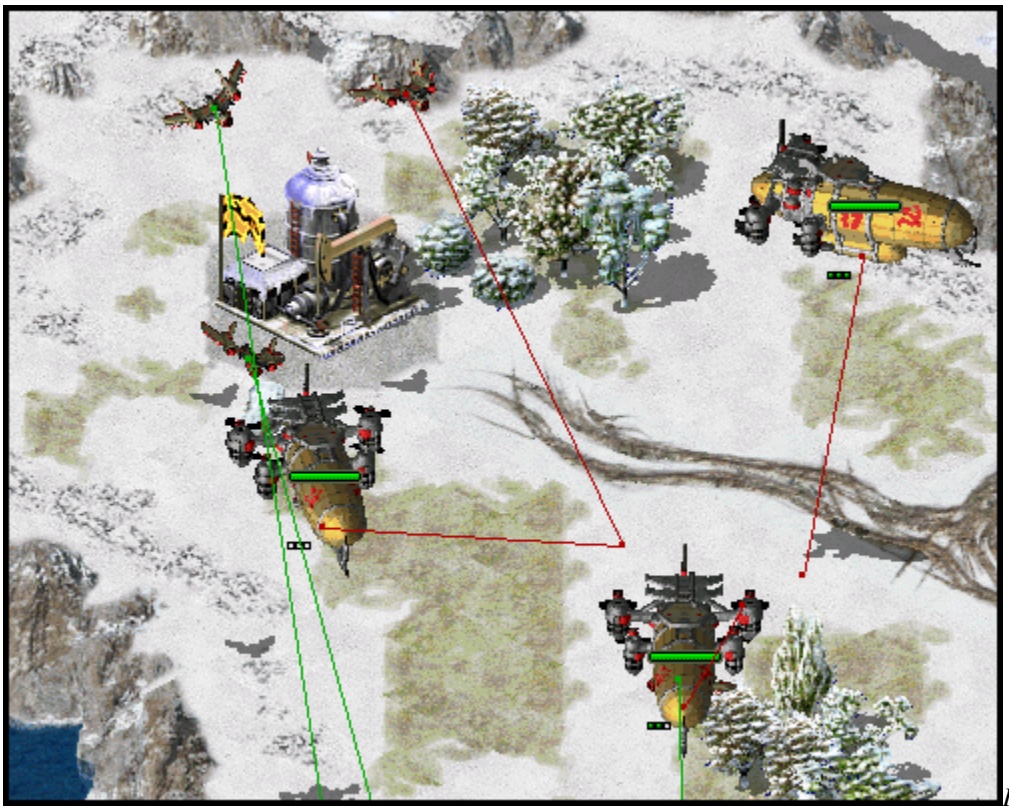
```
[SOMETECHNO]            ; TechnoType
LowSelectionPriority=no ; boolean
```

- This behavior is designed to be toggleable by users. For now you can only do that externally via client or manually.

In `RA2MD.ini`:

```
[Phobos]
PrioritySelectionFiltering=yes ; bool
```

8.3.2 Hide health bars



Health bars hidden in CnC: Final War

- Health bar display can now be turned off as needed, hiding both the health bar box and health pips.

In `rulesmd.ini`:

```
[SOMENAME]          ; TechnoType
HealthBar.Hide=no   ; boolean
```

8.4 Loading screen

- PCX files can now be used as loadscreen images.
  - You can specify custom loadscreen with Ares tag `File.LoadScreen`.
- The loadscreen size can now be different from the default `800x600` one; if the image is bigger than the screen it's centered and cropped.
  - This feature works in conjunction with CnCNet5 spawner DLL which resizes loadscreen window to actual monitor size and places the image in center. If there's no CnCNet5 spawner loaded, the window resolution will be always `800x600`.
- You can now disable hardcoded black dots that YR engine shows over empty spawn locations, which allows to use prettier and more correctly placed markers that are produced by Map Renderer instead.

In `uimd.ini`:

```
[LoadingScreen]
DisableEmptySpawnPositions=no ; boolean
```

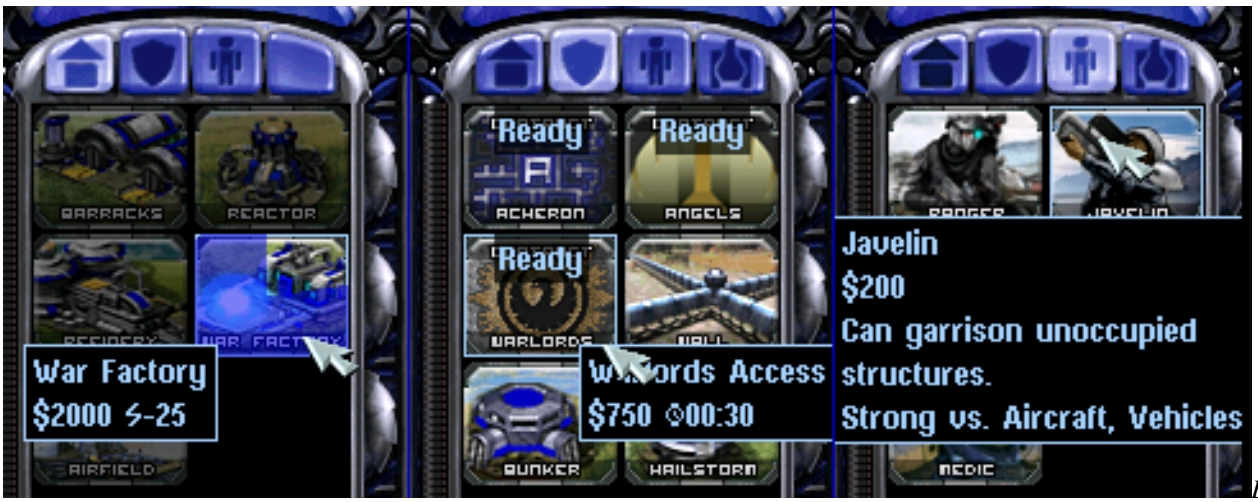
8.5 Sidebar / Battle UI

- It's now possible to switch hardcoded sidebar button coords to use GDI sidebar coords.

In `rulesmd.ini`:

```
[SOMESIDE]            ; Side
Sidebar.GDIPositions= ; boolean
                      ; default values are:
                      ; yes for the first side
                      ; no for others
```

8.6 Tooltips



Extended tooltips used in CnC: Final War

- Sidebar tooltips can now display extended information about the TechnoType/SWType when hovered over it's cameo. In addition the low character limit is lifted when the feature is enabled via the corresponding tag, allowing for 1024 character long tooltips.
- TechnoType's tooltip would display it's name, cost, power and description (when applicable).
- SWType's tooltip would display it's name, cost, and recharge time (when applicable).
- Extended tooltips don't use `TXT_MONEY_FORMAT_1` and `TXT_MONEY_FORMAT_2`. Instead you can specify cost, power and time labels (displayed before correspoding values) with the corresponding tags. Characters `$` `U+0024`, `U+26A1` and `U+231A` are used by default.
- Fixed a bug when switching build queue tabs via QWER didn't make tooltips disappear as they should, resulting in stuck tooltips.
- The tooltips can now go over the sidebar bounds to accomodate for longer contents. You can control maximum text width with a new tag (paddings are excluded from the number you specify).

Note: If you use the vanilla font in your mod, you can use the improved font (v3 and higher) which among everything already includes the mentioned icons. Otherwise you'd need to draw them yourself using WWFontEditor, for example.

In `uimd.ini`:

```
[ToolTips]
ExtendedToolTips=no ; boolean
CostLabel=<none>    ; CSF entry key
PowerLabel=<none>   ; CSF entry key
TimeLabel=<none>    ; CSF entry key
MaxWidth=0          ; integer, pixels
```

In `rulesmd.ini`:

```
[SOMENAME]           ; TechnoType or SWType
UIDescription=<none> ; CSF entry key
```

- The descriptions are designed to be toggleable by users. For now you can only do that externally via client or manually.

In `RA2MD.ini`:

```
[Phobos]
ToolTipDescriptions=yes ; bool
```

Phobos is a WIP community project providing a set of new features and fixes for Yuri's Revenge based on modified YRpp and Syringe to allow injecting code. It's meant to accompany Ares rather than replace it, thus it won't introduce incompatibilities.

For now you can discuss the project at a dedicated channel on C&C Mod Haven (which is my C&C modding server).

CHAPTER

NINE

BUILDING AND USAGE (WINDOWS)

1. Install Visual Studio with "Desktop development with C++" workload and "C++ Windows XP Support for VS 2017 (v141) tools" individual component and clone this repo recursively (that will also clone YRpp).
2. Open the solution file in VS and build it (`DevBuild` build config is recommended).
3. Upon build completion place the resulting `Phobos.dll` from folder named identical to the used build config in your YR directory and launch Syringe targeting your YR executable (usually `gamemd.exe`).

You can also get test a nightly version for a specific commit which is built automatically with a GitHub Actions workflow, just press on a green tick, open the workflow, find and download the build artifact (a ZIP containing the extension's DLL). Those versions have build information (commit and branch/tag) in them which is displayed ingame and can't be turned off. **Those versions are bleeding edge, do not redistribute them outside of testing!**

CHAPTER

TEN

DOCUMENTATION

The documentation can be found at here @ Read the Docs and is split by a few major categories (similiar to Ares docs), each represented with a page on the sidebar. Each page has it's contents grouped into multiple subcategories, be it buildings, technotypes, infantry, superweapons or something else.

You can switch between versions in the bottom left corner, as well as download a PDF version.

CHAPTER

ELEVEN

HOW TO READ CODE SNIPPETS

```
; which section the params should be in
; can be a freeform name - in this case the comment would explain what it is
; if no comment to be found - then it's a precise name
[SOMENAME] ; BuildingType
; KeyName=DefaultValue ; accepted type with optional explanation
; if there's nothing to the right of equals sign - the default value is empty/absent
UIDescription=<none> ; CSF entry key
```

CHAPTER

TWELVE

CREDITS

- Belonit aka Gluk-v48, Metadorius aka Kerbiter - project authors
- misha135n2 - YRpp edits
- tomsons26, CCHyper - all-around help, assistance and guidance in reverse-engineering, YR binary mappings
- Ares developers - YRpp and Syringe which are used, save/load code from Ares;
- DCoder - unused deployer fixes that are now included in Phobos
- CCHyper - current project logo
- ZPHYUS - win/lose themes code
- ayylmao, SMxReaver, 4SG, FS-21 - help with docs
- wiktorderelf, Metadorius aka Kerbiter - overhauled Unicode font

Thanks to everyone who uses Phobos, tests changes and reports bugs! You can show your appreciation and help project by displaying the logo (monochrome version can be found here) in your client/launcher, contributing or donating to us via links on the right and the `Sponsor` button on top of the repo.

CHAPTER

THIRTEEN

LEGAL AND LICENSE

The Phobos project is an unofficial open-source community collaboration project to extend the Red Alert 2 Yuri's Revenge engine for modding and compatibility purposes.

This project has no direct affiliation with Electronic Arts Inc. Command & Conquer, Command & Conquer Red Alert 2, Command & Conquer Yuri's Revenge are registered trademarks of Electronic Arts Inc. All Rights Reserved.