

NuVote!

An alternative voting system where majority rules!

This addon offers a replacement for the vanilla voting system, which has a few key benefits:

- Programmable majority rule settings
- Runtime rotation modifications - aka "Hot Soccing"
- Frequency controls for select maps
- Map popularity tracking
- Hell and encore likelihood tuning
- Faster roulette

All features are also available when the next map rule is set to Random (advancemap=Random).

This addon works with Hostmod 16 - 18+, Kart VS, NVJR Hostmod, ATO, Juggys Jug, and commandqueue ([which you should totally check out](#)) out of the box.

NuVote is not to be confused with NuMetal



Ch1 - Player controls

All controls relevant for normal players

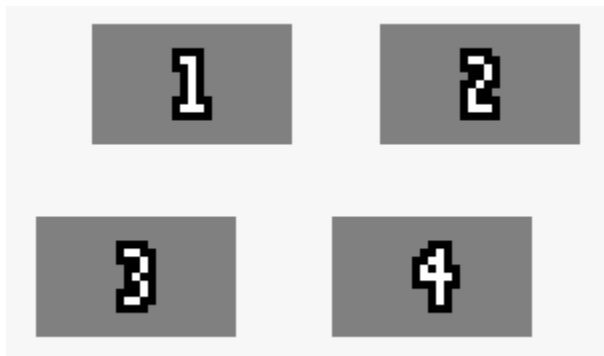
Making your choice

In order to cast your vote, you must not be spectating. NuVote will allow you to join during the voting phase.

In summary, casting your vote works just like in vanilla:

- Aim/Steer to move your cursor
- Accel to confirm your choice
- Spectate to rescind your choice

You can also type in chat your preference. Just type the number corresponding to your choice



Finally, you can also send one of the 4 commands which also corresponds to the graphic:

- `vote1`
- `vote2`
- `vote3`
- `vote4`

Vote mechanics

Pay attention to the top right. These numbers will affect the outcome of the vote:



FOR MAJORITY: If a choice reaches this many votes, it is guaranteed to be chosen

FOR ELIGIBILITY: A choice must reach this many votes in order to be chosen at all (i.e. during roulette)

Server owners: See chapter 5 to control this

The vote result logic goes as follows

- Choose the map which reaches the majority requirement
- If no map satisfies above, randomly choose any map which reaches the roulette requirement
- If no map reaches the roulette requirement, choose the map with the most votes
- If there is a tie among 2 or more maps, randomly choose 1 among them

Get info about the server

With this addon, server owners can fine tune a lot of parameters of the vote logic and algorithm. You can get a summary of these settings by typing !nvhelp in chat anytime other than intermission. Values and explanations are (hopefully) self explanatory.

Though I do want to explicitly call out:

FORCED HELL IS BACK

By default it's off, but beware, server owners can choose to turn it on. Check nvhelp to find out if it's enabled.

For the uninitiated, when this is enabled, if everyone in the lobby chooses random, the result will always be hell. Happy Halloween!!

Setting the clock

The clock on the bottom left, if available, can be adjusted to your preference



pcvar*	args	description
nv_showtime	On/Off; default=On	Toggles whether to show the clock at all.
nv_ampm	On/Off; default=On	Toggles showing a 12hr clock vs a 24hr clock
nv_tzoffset	[-12:59 - 12:59]; default=-4:00	<p>Adjusts the timezones for the clock. 0 is the UTC time zone, -4 is 4 hours behind UTC. Go to https://greenwichmeantime.com/current-time/ to find out your offset. To input minute adjustments, add a colon, then your minute adjustment. E. G. 9:30 adjusts to 9 hours, 30 minutes ahead of UTC.</p> <p>If you are in a place where daylight savings is observed, you may need to adjust when the biannual transition occurs.</p>

*- stands for psuedo-console variable. Settings are saved on your local storage, and persists between sessions and servers

If you see a clock, thank your server admin! The clock requires some work on part of the admin in order to show up. chatting !nvclockhelp will just repeat the info above

Strike your name from the record

NuVote helps server admins track the popularity of available maps by tracking the results for each voting round, veto, and rtv. If you prefer to remove your name from this data, you can with the following pcvar

pcvar	args	Description
nv_donottrackme	On/Off; default=Off	Toggles whether to strike your player name from popularity tracking log events

In order to strike your name from all log events, reach out to your server contact.

Check Map Status (advanced)

NuVote maintains a table for which maps are hell, preferred*, or can appear at all. This is separate from the status ordained by any loaded soc files. As a result, there is a separate console command for examining the status of a map:

```
nv_show_mapstatus <query>
```

type in place of <query> put the name of your map. This command will tell you if the map is delisted (out of rotation), in hell, and/or preferred


*- preferred maps will show up more frequently than normal maps. Think of them as the opposite of hell maps.

Cases when no vote will happen

If your server admin has either ATO or commandqueue installed, and there are maps yet to be played, NuVote will display which maps are coming up, and no vote will occur.

Ch2 Server setup

Out of the box, NuVote will work without any issues. By default, it is ON, but can be turned off at any point with `nv_enableNV`. Though there are a few things all admins should know before anything else:

- **exitlevel has slightly changed, please use exitlevel2 instead**
 -  If you have Hostmod, and have rtv enabled (Hostmod<=16), you will want to read [kartserv.cfg considerations](#).
 - Why: Vanilla's exitlevel will skip over NuVote's voting screen, exitlevel2 will not
- **Do not put any [commands which manage maps](#) in `kartserv.cfg`.** [More info](#)
- NuVote can display and play the graphics and music from the vanilla vote screen, and any replacements. More info in [Setting up branding](#)
- NuVote produces a file called `kartevents.txt` in lua which tracks the results for map voting. Good for tracking map popularity
 - Setting up the time coprocess is recommended for this - as a timestamp is placed for each event. See [Setting up time](#)
- If hostmod version <=16 is loaded, `hm_specbomb_antisoftlock` will be forced off

Master Controls

cvar	args	description
<code>nv_enableNV</code>	On/Off; default=On	Toggle for operation of the entire addon. Disabling will turn off all features, and vanilla vote will take place instead.
<code>nv_enableTime</code>	On/Off; default=On	Toggle specific for the time tracking feature. Toggle this off if you do not plan on taking advantage of any features that uses time (such as popularity tracking, which uses timestamps for events.) Note: NuVote will continue keeping track of in-game tics regardless of this toggle for internal state tracking.
<code>nv_enablePopTrack</code>	On/Off; default=On	Toggles operation of popularity tracking, which records vote results and rtv

		calls to <kartinstall>/luafiles/kart events.txt
--	--	-------------------------------------------------------

kartserv.cfg considerations

If you have `exitlevel1` as a votable command whether thru RTV or otherwise, it will not work by default. NuVote introduces the command `exitlevel2`, which essentially does the same thing, but allows for NuVote's **map election** to take place. A map election is the sequence that NuVote is replacing - the voting by players for which map to play next, out of a choice of four.

To enable `rtv` again, place the following command in your `kartserv.cfg`:

```
hm_votable exitlevel2
```

Similar to `hostmod`, NuVote will automatically issue `vote exitlevel2` for any player which types 'rtv' in chat.

Forbidden commands in kartserv.cfg

Most commands for NuVote can be placed in `kartserv.cfg`, and is encouraged. However, do not place any of the following commands:

- `nv_delist`
- `nv_relist`
- `nv_hellify`
- `nv_unhellify`
- `nv_neutralizehell`
- `nv_prefer`
- `nv_unprefer`

If any of these commands are put in `kartserv.cfg`, there is a potential for all other settings for map management to be lost. NuVote automatically manages map configurations, so there should be no reason to use these commands here.

Setting up branding

NuVote offers the option to emulate the voting screen by displaying the vanilla background, and playing the `vote/roulette/result` music during map elections. Servers which have replaced these graphics and sound files can enjoy the same branding benefits without changing much, as the same assets are used. To enable this feature, turn ON the following cvars:

cvar	args	description
nv_show_vote_bg	On/Off; default=Off	Toggles whether to display INTERSCR - the background for the vote screen in vanilla during voting phases
nv_play_vote_music	On/Off; default=Off	Toggles whether to play the music associated with all voting phases - voting/roulette/result

Setting up time

NuVote keeps track of the current approximate time for popularity tracking and other functions. However, due to some limitations of kart, NuVote will need help from a coprocess from outside of the game. Hence, the second attachment to the MessageBoard post: NuVoteScripts.

Time tracking can be disabled with `nv_enableTime`

Setup is relatively simple. To get started:

1. Open the NuVoteScripts.pk3 file, and navigate to the folder of your OS
2. Copy the files in the folder to a location of your choosing
3. Open the file you copied in. You will need to make some edits. The file will provide more details within.
 - a. If you are using linux, start with kartmon.sh
 - b. If you want to make the coprocess a linux service, a unit file is provided, but still begin with kartmon.sh

Do not add NuVoteScripts to your mod list! Nothing bad will happen, but it will add no value to your game in this way.

What the coprocess does

It just writes the current utc time according to your machine to `<kartinstall>/luafiles/currentutc.txt`. Efforts are made so that it only does this during certain conditions - e.g. only when `<kartinstall>/luafiles/ping.txt` is modified in some way. NuVote the doom addon will modify this file when it deems appropriate e.g. when a player joins the server for the first time in a while.

Disclaimer about coprocess installs

I have not tested these procedures on any Linux distro other than Ubuntu. I don't claim to be a unix/linux/windows expert. These scripts were made to fulfill a niche usecase, and are **NOT** foolproof. I've provided the following disclaimer in each file, and I'll repeat it here:

DISCLAIMER: Use these scripts at your own risk.

These files are provided as-is, without any express or implied warranties. The author(s) disclaim all liability for direct, indirect, or consequential damages resulting from your use of these scripts, including but not limited to loss of data or damage to your system. It is your responsibility to review and test the script to ensure it meets your requirements before using it in any production environment.

The author(s) make no guarantees regarding the accuracy, reliability, or suitability of this script for any purpose. Use caution and exercise due diligence when running scripts from untrusted sources.

Ch3 Vote Configurations

One of NuVote's core features is controlling the voting experience for all players. This section covers the following controls offered by NuVote:

- Majority rules logic
- The algorithm which decides the maps to be voted on

Voting Logic

The result of a vote election is controlled by two cvars:
`nv_supermajority_pct` and `nv_superminority_pct`

These control the majority, and minority (aka roulette) requirements, respectively. These numbers are what you see as FOR MAJORITY and FOR ELIGIBILITY during the voting phase.

For convenience, this is the vote logic again. It follows these steps:

- Choose the map which reaches the majority requirement
- If no map satisfies above, choose randomly any map which reaches the roulette requirement
- If no map reaches the roulette requirement, choose the map with the most votes
- If there is a tie among two or more maps, randomly choose one among them

The calculation for the FOR MAJORITY and FOR ELIGIBILITY numbers is as follows:

```
FOR MAJORITY = ceiling(nv_supermajority_pct * # non-spectators)
FOR ELIGIBILITY = ceiling(nv_supermajority_pct * # non-spectators)
```

Below are some common vote rulesets, and the config required to achieve them

Result logic	Settings to put in
Default NuVote logic: Choose a map when almost all players vote for it, and do not choose maps which almost no players vote for	<code>nv_supermajority_pct = 75</code> <code>nv_superminority_pct = 20</code>
Default vanilla logic: Any map which is voted for can be chosen	<code>nv_supermajority_pct = 100</code> <code>nv_superminority_pct = 0</code>
Always choose the map with the most votes (roulette as tie breaker)	<code>nv_supermajority_pct = 51</code> <code>nv_superminority_pct = 50</code>

Warning

Setting `majority_pct` to 51 is NOT recommended. Under this setting, the outcome of a map election will more often fall to the decision of the last voter. With this in mind, some players may deliberately withhold their vote for longer, in order to let the outcome be the result of their decision. The default `nv_supermajority_pct` setting is at 75 in order to discourage this behavior.

The Choices Algorithm

As a server admin, you have the ability to change many aspects about which maps are served up for voting. The aspects include:

- How often a choice is converted to encore
- How often hell maps appear
- How often preferred maps appear
- How many rounds to keep a recently played map ineligible to be voted for again

All you need to know is covered under [Vote Behavior Settings](#). But keep reading the next section to get primed on how preferred maps work.

Introduction to preferred maps

Preferred maps is a novel feature of NuVote which allows server owners to promote a select number of maps. When a map is preferred, they will appear more often in map elections. This is great for showing off maps by members of your community, or maps that are just so good you want to play them more.

Here's how it works:

Indicate to NuVote which maps you prefer with `nv_prefer <map>***`

Every 20* rounds, NuVote will pull 2** maps from the pool of preferred maps for a map election. Preferred maps still can show up during normal elections too. If your next map rule (`advancemap`) is Random, a preferred map will load every 20* rounds.

*- Can be controlled with `nv_preferred_every`

** - Can be controlled with `nv_preferred_ct`

***-Can be undone with `nv_unprefer <map>` (See [Map management commands](#))

Considerations for preferred maps

When deciding which maps will be preferred, pay mind to the number of maps you choose. The more maps you choose, the less likely all other preferred maps will become as a result of dilution.

Also consider the amount of activity your server gets. The default setting is "2 every 20", which roughly translates to one map election per hour. This setting will hit different if your server has a

session daily which lasts 5 hours. As of writing, it's unclear if 2 every 20 is too much or too little. Adjust to your taste.

Vote behavior settings

Here's a list of all CVARS which control how voting works, and the choice algorithm:

cvar	args	description
nv_supermajority_pct	[51 - 100]; default=75	Percent of players required to vote for a particular map for majority rule to take effect
nv_superminority_pct	[0 - 50]; default=20	Percent of players required to vote for a particular map for it to be eligible in the roulette
nv_hell_as_random_pct	[0 - 100]; default=1	Percent chance for the random choice in a map election to be hell
nv_hell_as_nonrandom_pct	[0 - 100]; default=0	Percent chance for a non-random choice in a map election to be hell. Players will be able to discern a map as hell
nv_hell_as_randommap_pct	[0 - 100]; default=1	Percent chance for a hell map to be the destination after calling randommap in console. Percent chance for a hell map to be rolled when <code>advancemap=random</code>
nv_encore_pct	[0 - 100]; default=5	Percent chance for a map choice to be turned into encore
nv_encore_round_pct	[0 - 100]; default=0	Percent chance for an "encore round" to occur. An encore round will modify the choices in a map election where 3 maps are encore, and 1 is not. Choices are not affected by <code>nv_encore_pct</code> when an encore round is active

nv_preferred_every	[0 - MAXINT]; default=20	How often preferred maps will show up in map elections. When set as <i>n</i> , NuVote will pull from the preferred map pool every <i>n</i> map elections. 0 means they won't show at all.
nv_preferred_ct	[0 - 4]; default=2	How many maps from the preferred map pool shall be pulled, when it's time to serve preferred maps. 0 means they won't show at all.
nv_recent_maps_size	[0 - 1030]; default=20	<p>NuVote will remember the last <i>n</i> maps which were recently played on. These maps will not show up in the next map election.</p> <p>nv_recent_maps_size controls <i>n</i></p> <p>Note: NuVote may adjust this setting to allow for at least 5 maps to show up for a map election.</p> <p>Note: NuVote will forget all recent maps if no one is on the server for at least 30 minutes, if time is set up correctly.</p>
nv_show_smr	On/Off; default = On	Toggles whether to show the FOR MAJORITY/FOR ELIGIBILITY indicators. Might be helpful in deterring vote shenanigans.

Ch4 Managing Maps

NuVote adds the capability of adding and removing maps from rotation, all without needing to create a soc file, nor restarting the server. Aside from this, NuVote can also:

- Move a map into hell
- Move a map out of hell
- Make a map preferred
- Make a map not preferred

To keep track of all of these settings, NuVote maintains a table in memory, and disk. Every time a map setting is changed, the table is written to disk. This table is written as a csv to the following location

```
<kartinstall>/luafiles/NuVote Config.txt
```

NuVote is compatible with soc files out of the box, and will automatically convert your soc settings into a single, easy to manage, csv file. Gone will be the days of managing rotations by messing with soc files 🎉

Map management commands

Below listed are all the commands to manage every map. You can just type the name of the map you want to manage for any commands which take <query> as the argument. Alternatively, you can type the “extended map name” - e.g. “MAP01” or just “01” for the map name. The query will not work if there is more than one match, or 0 matches.

command	args	description	example
nv_delist	<query> [-r <reason>]	<p>Queries for a map. The result is delisted from all NuVote map elections. It won't show up for votes.</p> <p>Text after optional -r flag will be recorded, and available in NuVoteConfig.txt, and nv_show_mapstatus. Use this to remind yourself why a map was delisted</p>	<p>nv_delist Green Hills -r Not yet sand hills</p>

nv_relist	<query>	Puts a map back into rotation, if it was ever delisted. If it was not delisted, no effect. Reason is deleted	nv_relist Green Hills
nv_hellify	<query>	Puts a map into hell rotation; The map will appear as often as the hell cvars will specify.	nv_hellify Vanilla Lake 2
nv_unhellify	<query>	Takes a map out of hell rotation; The map will appear as often as any normal map	nv_unhellify Crystal Abyss
nv_prefer	<query>	Marks a map as preferred. Preferred maps show up at least as often as nv_preferred_every specifies. See Introduction to preferred maps .	nv_prefer Blue Mountain
nv_unprefer	<query>	Unmarks a map as preferred, if it was previously preferred, else no effect.	nv_unprefer Blue Mountain
nv_save_mapconfig		Saves the current map configuration - which are delisted, which are in hell, which are preferred - to: <kartinstall>/luafiles/ NuVoteConfig.txt	
nv_load_config		Loads the full map configuration file. This may take a few seconds to fully complete. Please see warning	
nv_show_mapstatus	<query>	Will show the current status of a map in	nv_show_mapstatus Green Hills

		regards to NuVote. Will show if it's delisted, hell, or preferred	
--	--	----------------------------------------------------------------------------	--

Warning on nv_load_config

While it is generally safe to run a map election while nv_load_config is running, it is **NOT** safe to run any of the other [commands which modify the status of a map](#) (e.g. commands above nv_save_mapconfig) while this is happening. The following message will appear when it is complete, and it is safe to issue the aforementioned commands again:

```
Finished loading x map configurations
```

x = number of map configs present

nv_load_config is always run when NuVote is loaded (usually at server start). This is why you should not put any of these commands in your kartserv.cfg file.

Managing maps from the config file

When the game program first starts, it pulls all map configurations from

```
<kartinstall>/luafiles/NuVoteConfig.txt
```

The game can reload this file at any time with the command nv_load_config at any time. This file is a csv, so it can be loaded into a spreadsheet program. Simply save your changes in the program before loading, then run the command for the changes to go live.

Below is the schema for the NuVoteConfig.txt file

Field	Description	Example	Purpose
rawid	DOOM's internal ID for the map (numeric: 1 - 1035). ⚠ Do not modify this field. The configuration for the map will be lost.	915	Primary Key
extid	Short for Extended ID - the more common ID for a particular map	MAPWN	Informative

name	The name for the map that is human readable.	Smash Kart Circuit	Informative
gametype	The gametype for the particular map, according to the available soc files. battle/race	race	Informative
delisted	1 = is delisted 0 = not delisted Delisted means it will not show up for elections	0	Configurable
hell	-1 = Let available SOC files dictate whether map is in hell (default) 0 = map is not in hell 1 = map is in hell	-1	Configurable
preferred	0 = not preferred 1 = preferred Preferred maps show up more often for votes	1	Configurable
reason	The written out reason for delisting a map	Too much violence against smash characters	Configurable

Ch5 Popularity Tracking

NuVote helps you keep track of the popularity of each map by recording:

- all votes made during each map election
- Any calls for hostmod rtvs/vetos

All data is written to storage at

`<kartinstall>/luafiles/kartevents.txt`

Cvars associated with this function:

cvar	args	description
nv_enablePopTrack	On/Off; default=On	Toggle for popularity tracking. When disabled, no records will be recorded.
nv_include_total_state_in_tracking	On/Off; default=On	The schema for popularity tracking includes 3 large columns. These columns record the whole state of which maps are enabled/hell/preferred. This is a toggle to disable recording this specific data as it is large.

If a user asks you to remove their username from popularity tracking, it's this file which you will need to change. All that you will need to do is remove the username and skincolor for each line where the username exists. More on how to manage below.

Managing kartevents.txt

The kartevents.txt file can be easily managed as the file is actually a .csv file (kart only allows the .txt extension to be written). To load it in a spreadsheet program, simply make a copy and rename the extension to .csv. If you need to make changes, be sure to overwrite the .txt file so that changes persist.

There are several event types which can be recorded:

Event Type	Description
ballot cast	When a map election finishes, a ballot cast event is made for each person who voted. It encapsulates which map they voted for, and some context around it
rtv	Captures when someone RTV/Vetos a particular map. It also captures when someone agrees or dissents a particular call for rtv.

As a result of this, some fields may not be populated in all rows

kartevents.txt schema

Below is an explanation for each field

Field	description	example	Field number	eventtype
schemaver	Schema version for the record. This should only contain one value throughout the kartevents file. See [If schemaver has multiple values] if not.	1.0	1	ALL
eventtype	Indicates what kind of tracking event the record is. Will indicate whether certain fields will be relevant	ballotcast	2	ALL
eventid	An identifier for a particular record. Use this for the primary key, if loading into a DB or dataframe	"ABCDF"	3	ALL
timestamp	UTC Epoch timestamp in seconds. Good for plotting events over time	1542359569	4	ALL

servername	The name of the currently running server	SRB2Kart Server	5	ALL
name	Username of the player that is creating the event	cglitcher	6	ALL
skin	Skin name (aka character) of the player	curlybrace	7	ALL
color	Skin color of the player	peach	8	ALL
speed	Speed setting of the player	5	9	ALL
weight	Weight setting of the player	6	10	ALL
speedweight	Speed and weight, concatenated	56	11	ALL
playerchoiceidraw	Rtv: the map id the user is RTV-ing ballotcast: the map id the user is voting for Note: This is the raw id, which does not start with MAP*	1035	12	ALL
playerchoicefullraw	Same as playerchoiceidraw, but instead of map ids, it's the name of the map in human readable format	Unused Podium Map	13	ALL
playerchoiceid	Same as playerchoiceidraw, except if the user voted for random, the ID will be 0 instead. Players won't know they voted for a particular map, so we may want to discount it for popularity tracking	0	14	ballotcast
playerchoicefull	Same as playerchoicefullraw, except if the user voted for random, the map will show up as 'RANDOM'	RANDOM	15	ballotcast
playerchoicewasencore	True/False on whether the map in question was encore (1=true,0=false)	1	16	ALL
playerchoicenum	Choice number 1 - 4. 1 is top left. 4 is for the random choice	4	17	ballotcast
playerwasafk	True if the vote timer reached 0 before they could vote. Will be False if the player voted for a map, then rescinded, but couldn't vote for another map in time. Can be used to determine if player actually dislikes the other maps	False	18	ballotcast

choicesids	Pipe separated list of the choices, represented by raw ID (not MAP*). Choice mappings: 1 2 3 4	463 932 915 1035	19	ballotcast
choicesfullform	Same as choiceids, except the title is human readable.	Sky Babylon Windy Canyon Smash Kart Circuit Unused Podium Map	20	ballotcast
choiceencorecapacities	Pipe separated list of whether the choices is encore. Follows same format as choiceids	1 0 0 0	21	ballotcast
choicepoolwaspreferred	True if for the voting round, maps were pulled from the preferred map pool. Filter any trues in analysis to get cleaner popularity tracking	0	22	ballotcast
wantsrtv	During an rtv - records whether player wants rtv. In most cases will be true, except when they explicitly vote no to an RTV call. IDCs are not counted at all	1	23	rtv
isinitiator	During an rtv, indicates whether player started the call for rtv or veto. Player must be the first to call veto to get this, or first to type rtv in chat	1	24	rtv
fulldeliststate	Base 64 encoding on which maps are eligible to be voted for. Encoding is Big Endian. Bit 1 represents the state for map01, etc. 1 = map is votable, 0 = map is delisted	/+/////r//fgPYAgAIAAE AH/////4AAgfAAB/BR gIAAAegAGAANgAiA AIM3//4ACAAAAAA AAAAAH/9gHA6GAA D5wAAfFADL8BQAA AABACQAF+iAAAAA	25	ballotcast

		AAACQLU AAAvAACI AAUEACdI AADJuAA AYAAAAIB ET5gAAjP gAAAYf/gB oo		
fullhellstate	Same as fulldeliststate, except indicatates if map is in hell. 1 = hell, 0 = not hell	AAAAAAAA QAAAAAI AgAAAAA AAAAAAAA gAAAAAA AABAAQA AAAgAAA QIAAAAA AAD/9YAA AAAAAAA/ /AAAAAAA AAYAAAAI AAAIAEAA AAAAAAA BAAAABG CAAAAAA AAAABEA AAACAAI AAAAAAA AAAAIIAA AQAAAAQ BAAAAAA YAAAAAA b/gAAI	26	ballotca st
fullpreferredstate	Same as fulldeliststate, except indicatates if map is preferred. 1 = preferred, 0 = not preferred	AAAAAAAA AAIAAAAA AAAAAAAA AAAAAAAA AAAAAAAA AAAACAA AAAAAAAA AAAAAAC AAAAAAAA AAAAAAB	27	ballotca st

		AAAAAAA ABgAAAA AAAAQAA AAAAAAA AAEIAAAA AAAAAAA AAAAAAA AAAAAAA AAAAAAA AAAAAAA AAAIAAAA AAAgBgA AAAAAAB gA		
--	--	-----------------------------------------------------------------------------------------------------------------------------------------------------	--	--

Ch6 Limitations

NuVote was built to alleviate a few gripes I had with the voting system for race servers. Less time was devoted to a few other aspects, or not at all. Here's a few aspects to watch for:

- Battle has not been thoroughly tested
- NuVote currently does not offer the option to show race and battle maps during the same election
- The behavior for NuVote is undefined when the next map rule (advancemap) is `Same` or `Next`
- NuVote tries to always have it possible for at least 5 maps to be eligible for rotation, however, if too many maps are delisted (like such that only a few are left), behavior is undefined
- No efforts were made to be compatible with local multiplayer
- Most testing was done using a dedicated kart server. It should work ok with a non-dedicated, though testing is limited
 - NuVote will not work properly if all maps are not yet considered unlocked by the game

If this addon becomes popular enough, I may have some motivation to fix some of this

Disclaimer on code readability

TLDR: I know my code is a mess, but I don't have the time nor motivation to fix it

I've spent too much time on this project, and at this point I'd like to move on to different projects. I knew this project would be big starting out, but scope creep got me anyway. The result: you'll see foundations for systems which only get used once, duplicate implementations, mind boggling implementations, 15 different coding styles, and functions more fragile than the tail of a [Prince Rupert's drop](#).

To help with this, I'm making the lua of this project reusable. If there's enough interest, I'll make a github so that any changes from the community will get to everyone.

Acknowledgements

Thank you to all of the following people who helped make this possible:

Haya - Design for roulette

Snu, Freeman - end of round detection and help with hostmod intermission interop

Tyron - Help with hostmod veto interop

Ashnal - General lua and linux help

!Not - Code structure for HUD code

Juggy - Juggy's Jug of Tools interop

cfiancee for all her support!!

Special thanks to all members of the Stray Banana community for keeping me going each week, and bearing with all pre-release testing. Thank you to all players on CG's Smugbox!

Statement on reusability

All code is reusable* for SRB2Kart on the condition that I'm (cglitcher) credited. Graphics are not reusable at this time. If you have an improvement you'd like to be added to the official release, please contact me on discord.

* - This addon may NOT be ported to Dr Robotnik's Ring Racers. Before starting, please contact me.