Grow Castle autobot manual 06.07.2024 (1.39.6)

What is necessary:

- Windows 10 at least.

- Nox, configure it ([Nox setup](#)).

- the screen is at least 1520x865, if you have smaller screen, you will only be able to start in background mode. To check - you need to move the cursor to the lower right corner of the screen in the clickerman editor, and see what coordinates it shows: **X, Y: 1519, 864** or more. Also set Scale in display settings to 100% in **Scale and layout** section.

- The taskbar should be at the bottom of the screen (optionally).

- Clickermann app (**download from repository `Clickermann v4.13 x64.rar`, if you don't have one, because there are broken clickermanns on internet, that cannot call dlls**).

- my app (if it doesn't open, check if all dlls are in the same place as exe file). Also the app cannot be opened with windows below 10:
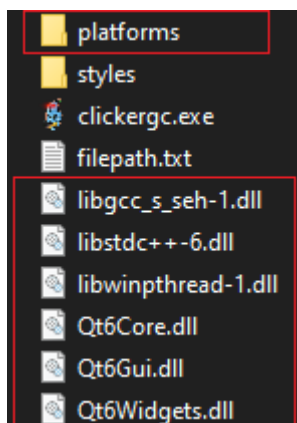
/platrofms
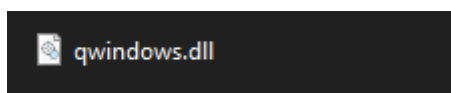
If you still cannot open app, you can write desired parameters directly inside variables in file **gc_bot_settings.txt**.

Links:

https://www.bignox.com/ - Nox

https://github.com/Silpux/GrowCastleAutobot- Clicker's github repo - clickermann can be found in there.

Nox setup

Grow castle must be here. When it restarts the game, it will click here
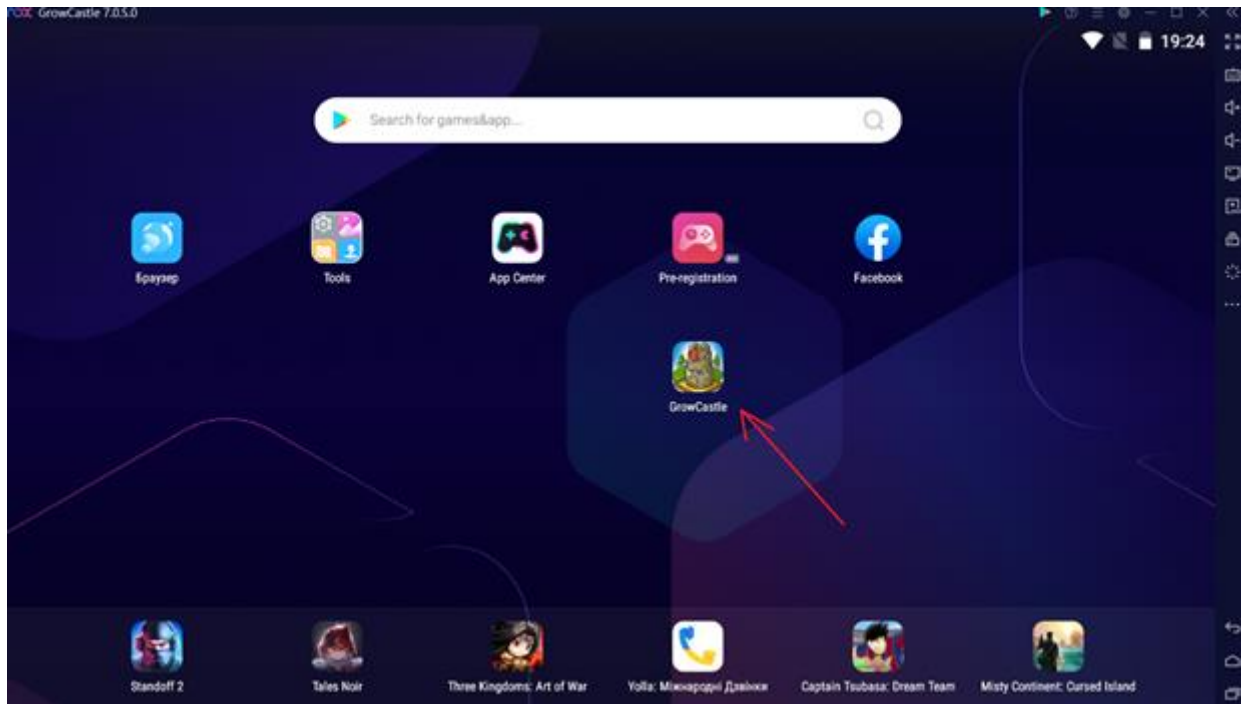
Move Cleanup button and Reset button here:
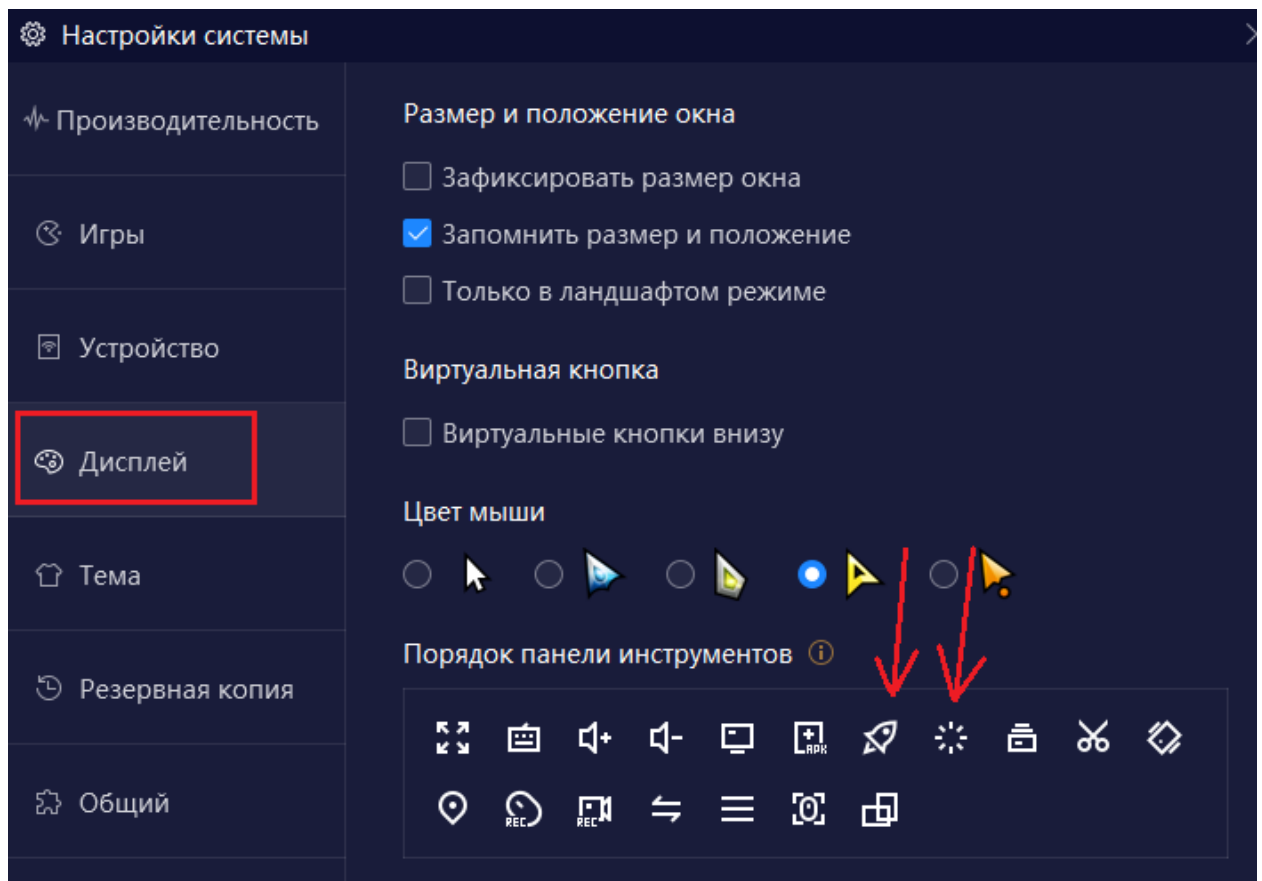
These buttons must be here (keep the side bar open always):

Set resolution to 1920x1080
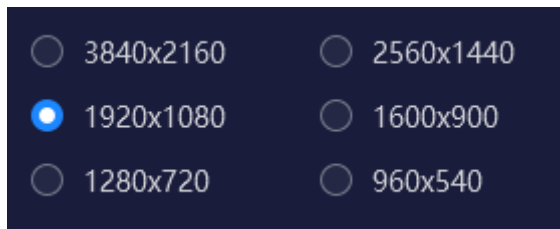
Set 60 fps

Check this thing to right-click to move backward (make sure that right click makes same thing as Esc button)
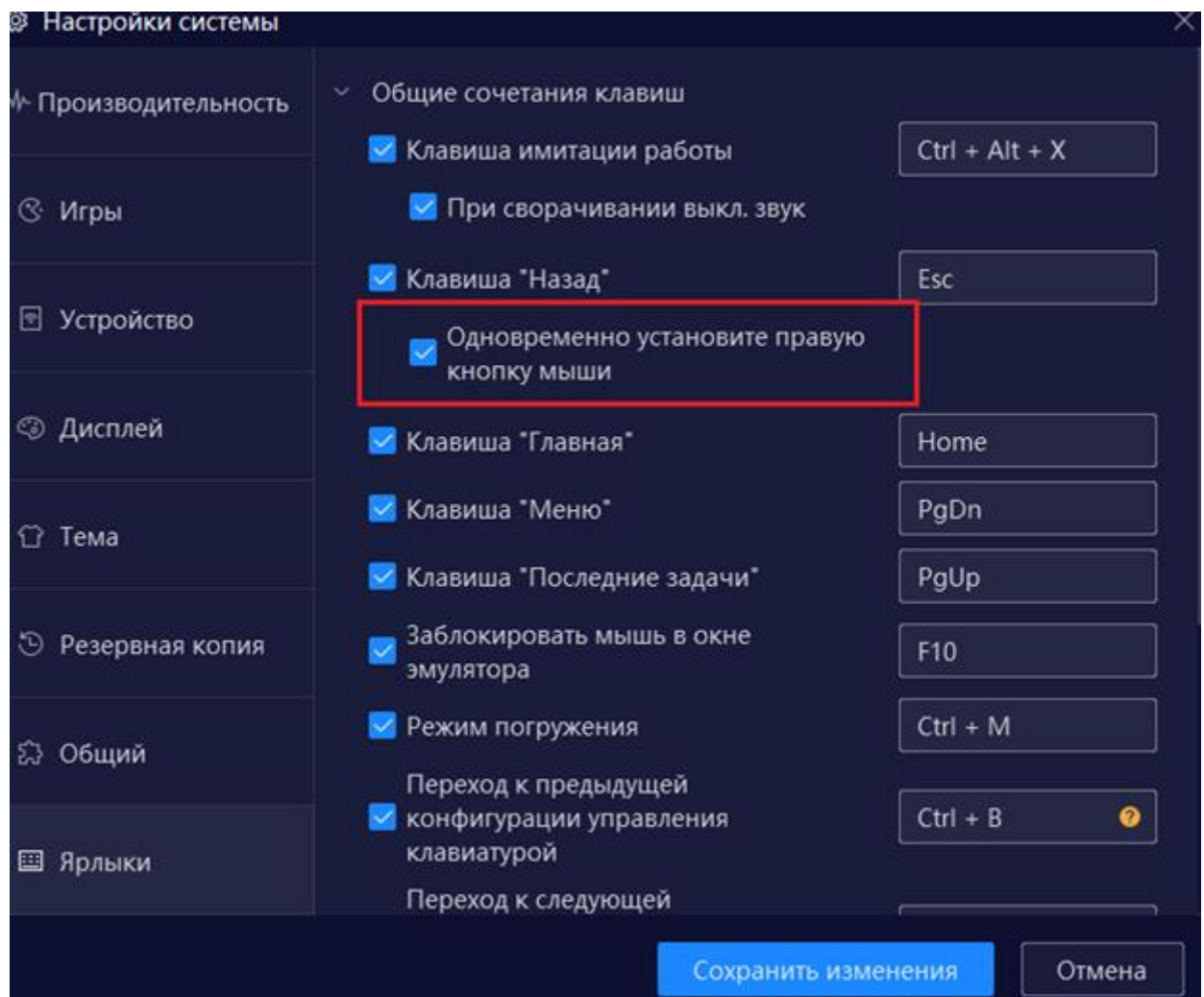
Only grow castle has to be in recent apps, otherwise clicker will not see clear all button when it restarts the game.
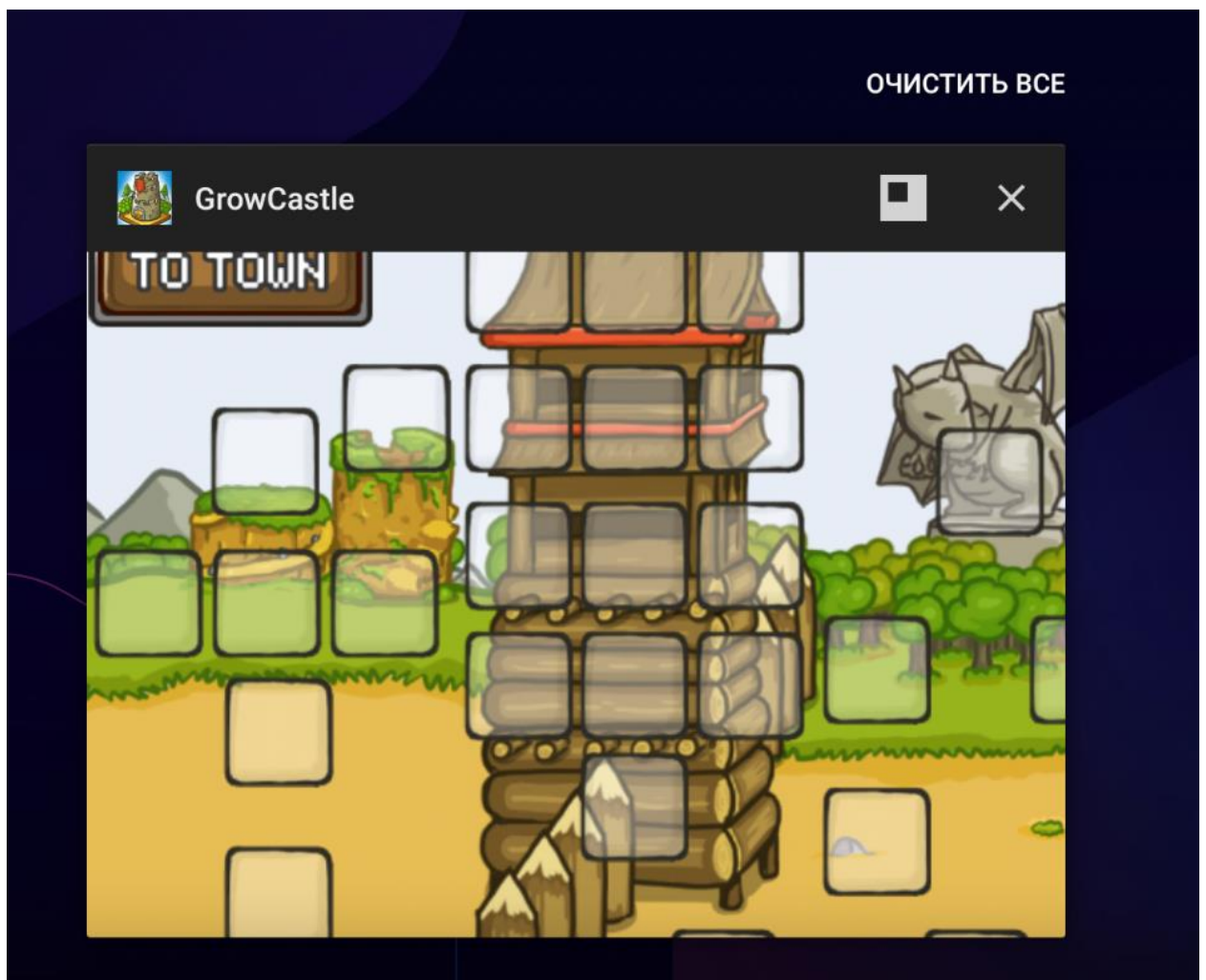
The language of "clear all" doesn't matter. It just has to have at least one pixel of clear white color – 16777215 inside (985,91,1101,131) area, and background shouldn't be white. It is better to leave default colors for nox.

Enable in editor and check. Also keep colormode at 0 in editor – the number under enlarged area on right (do it only when the window is in correct position, see [Set pos](#)):
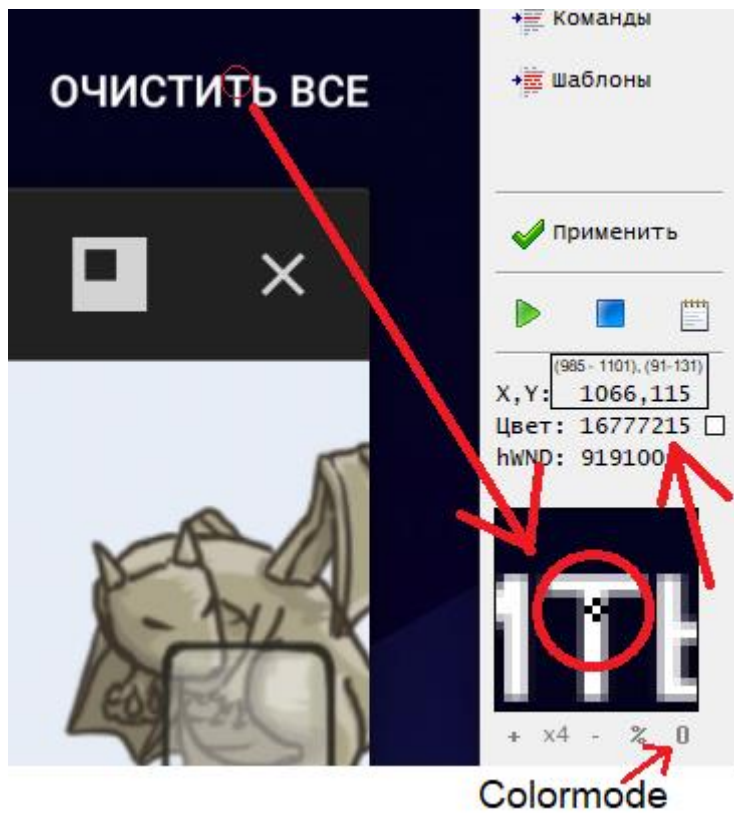
It is desirable that the computer normally plays at 50-60 frames per second. There may be problems due to lags.

Check the color of the sky in gc (for some reason some people had different color here). Skip this if you have 16182759, like me (don't forget to check colormode in editor – it has to be 0. Another colormode will give you wrong results):

If you have different color code, then you need to go to **the gc_bot_ settings.txt** file - and in it, at the very beginning, write the number shown by the editor into the **SKY_COLOR** variable:
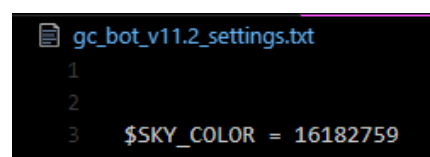
Also check some other colors to be sure that they have correct code:
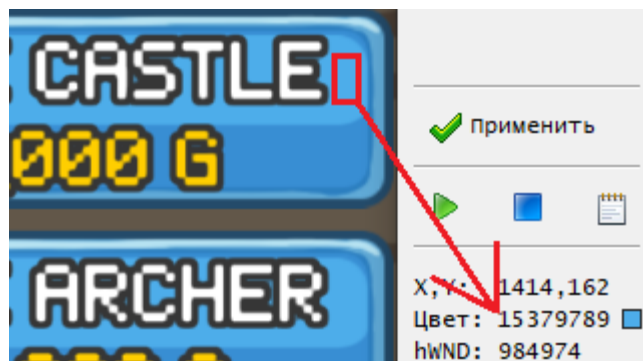
If you have the same values, everything should be ok.

When working with a clickerman, it is important to control that it does not open several times, otherwise problems will begin. Check here regularly:

 - ok not ok.
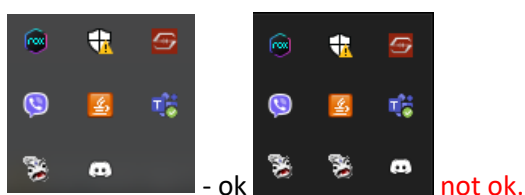
If several clickerman icons are displayed, it would be best to quit of all of them and open again.
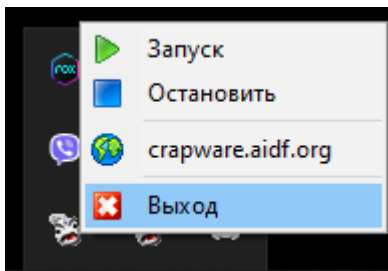
Be careful with minimizing clickermann window, because by default it goes in system tray and disappears from task bar. To fix this, go to clickermann's folder, open data, then config.ini file,

```
; Если 1, то в свернутом состоянии программа не будет отображаться на панели задач
; Только в системном трее
only_tray = 0
```

And put 0 to **only_tray**. If you have downloaded clickermann from github repo, it is already set up there.

Clicker setup

If you want to change language to english, go here:
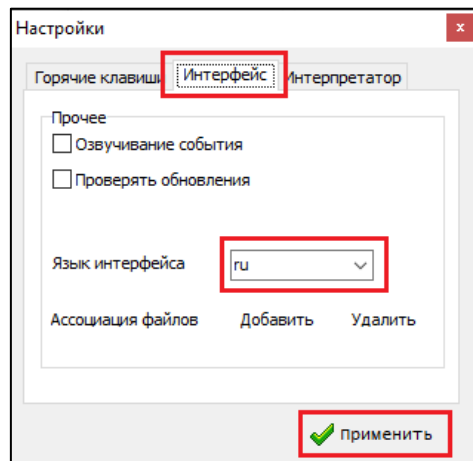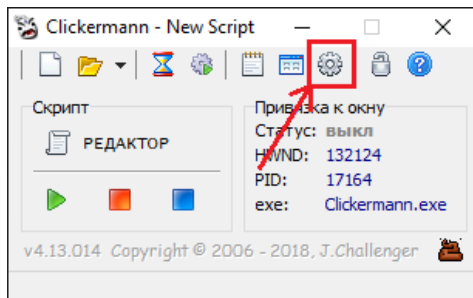
Set start and stop keyboard shortcuts and remember it. If you have downloaded clickermann from my github repo, I have **Alt + 1** for start, and **Alt + 2** for stop:
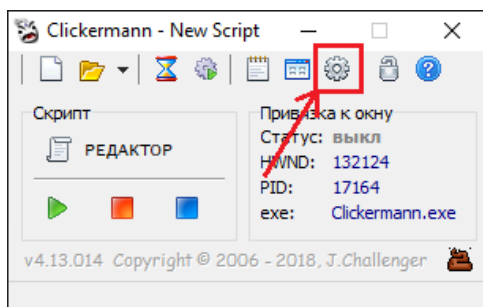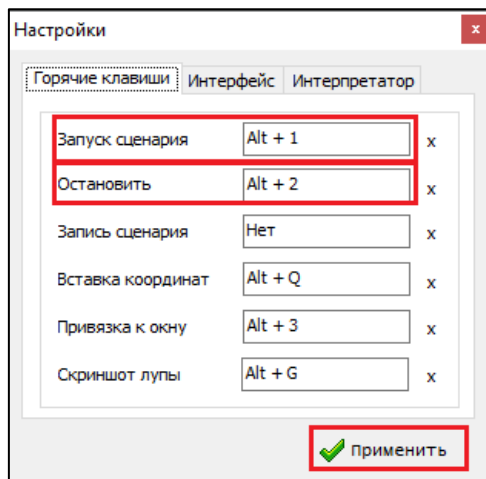
To set up clickermann to work, click on the folder:
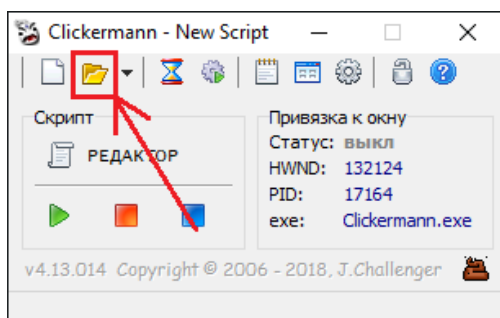
Then you need to select the file **__open_in_clickermann.cms** from the clicker's core folder. Then you need to check the editor. It should be like this:

```
 1
 2 #logfile "gc.log"
 3
 4 #include "colorlist.txt"
 5 #include "gc_bot_lib.txt"
 6 #include "gc_bot_settings.txt"
 7 #include "gc_bot_compiling.txt"
 8 #include "gc_bot_solvecaptcha.txt"
 9 #include "gc_bot_init.txt"
10 #include "gc_bot_main.txt"
11
```

After **Parameters setup**, you can run and the clicker should work. If it doesn't work, carefully check if the nox is set up correctly. Check sky color in gc. It must be specified correctly.

If you see this error when starting clicker:
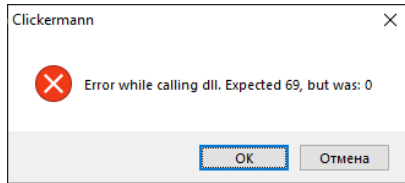
Check if **slpx.dll** and **opencv_world490.dll** are in clicker's core folder. If they are, it means that clickermann didn't call it for some reason. It can be because you use broken clickermann. Try downloading one from github repo. There is `**Clickermann v4.13 x64.rar**`, the clickermann is there. If the issue persists, I don't know what to do. All you can do is turn off solving gift.

Do not run clicker with clickermann's editor open, because it will be very slow. Close it every time before run.

Warn: If the clicker starts the wave and do nothing, it is possible that you configured for example build 2, and told clicker to play on build 1, which is empty.

Parameters setup

Open the **clickergc.exe** app, click on **Select main settings file**, then you need to select the **_main_settings.txt** file from the core folder.

After that it should be like this:
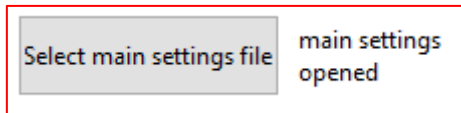
Now, to check if the changes are saved, you need to check the boxes somewhere and restart. If the checkboxes remain, then everything is fine. If not, then you need to re-select the settings file.

Window name

This is the name of the nox window. You need to enter its name there, without the version. Then, by pressing **Set pos**, the window should take the size that is needed for the clicker to work.
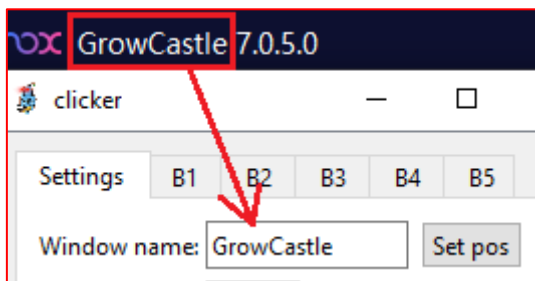
If it says **cant find window**, then there is an error in the title:
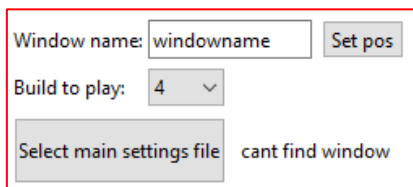
It is desirable to make sure that there are no windows with the same name as nox, as it can grab the wrong window, because of this there may be problems.

Build to play
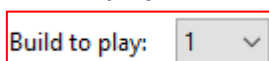
The number of the build that the clicker will play.

All 5 builds are configured here:
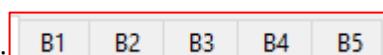
Example of a customized build

○ Pw
○ Smith
○ Orcband
○ Miliraty F
○ Chrono
○ clickable

Castle:

	Milit	X	Orc band
X	X	X	X
	X	Pw	X
X	Chrono	X	
X			

Pw – pure wizard. It should be in the center, it is pressed when there are no clickable heroes, which have a few pixels left for recovery.

Smith – it is pressed when there is not enough hp. It may not have time to press when the damage is too fast. There can be problem when the HP number is way too high, and the checked pixel is overlapped with that number.

Orc band and **Military F** – if you specify them, they will be pressed once at the beginning of the wave when 30 waves are skipped (you can set them to be pressed always on wave start here).

Chrono – will be pressed first at the beginning of the battle.

Clickable – other heroes that should be clicked. They are pressed if they have a blue line. Marked with an X.

Note: It doesn't work on heroes that need a scope, such as Ice Mage. For them, you need to go into the code and change the behaviour for a specific position (here), or simply remove them.

Farm dungeon – from the drop-down list you select a dungeon, if there is a checkmark, it will play there.

For dragons: whether to dust clothes of different grades.

For dungeons: whether to start cast heroes only when the boss came out with given delay.

Make replays if dungeon doesn't load: If the internet has gone, and clicker cannot open dungeons, if it is checked, then **until next gift** the clicker will play replays. Then will switch back to dungeons.

Warn: **If the inventory is full, then if this is enabled, will switch to replays, but if not, it will try to enter dungeon constantly. It is your responsibility to check if invertory is full.**

Note: All runes will be collected from dungeons.

  - make replays of the last wave. Cannot be combined with dungeon farming or skips.

Skips settings:

Skip waves – skip the waves obviously. If you are not skipping with oranges, the script will look on the white number of crystals to see if there are 30 crystals. It cannot be combined with **Upgrade castle**, because they both spend crystals.

Note: To get your crystals count, the clicker looks on the width of white number of crystals. If you have 1-9 crystals, it thinks you have 0 (minimal possible), 10-19 => 10, and if you have more, and the number does not contain 1 inside itself (because 21 and 12 have same width, and clicker chooses minimal possible, i.e. 10), then it will look on the first number to determine if it is 2 or not. If first number is 2, then you have 20 at least, otherwise you have 30, and clicker will skip. So,
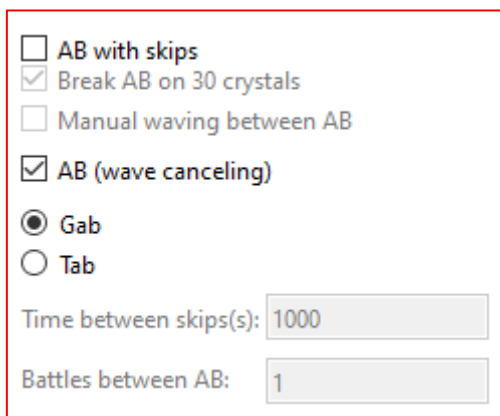
the clicker WILL NOT skip when you have 29-, 31, 41, 51… crystals.

(Except of situation when breaking AB waiting when 30 crystals are collected. In this case, will skip always).

5 waves between skips – 5 waves between skips. That is, after skipping 30 waves, the next 5 waves will definitely not skip. This is so that the orc band always gives another wave. In addition, a couple of fractions of a second each wave will not be spent counting the number of white crystals.

Info: Orc band and military are pressed only with a skip of 30 waves by default. It is useless to use them without the skip, as they eat the crystal, which is given as a reward for the wave. (can be changed in settings file here).

Skip with oranges - skip with oranges. It will always skip, no matter how many crystals you have. If the oranges are over, the store will open once and script will disable skipping with oranges automatically.

AB with skips. If enabled, the clicker will turn on AB, then skip waves, and in a circle. Specify whether to put on gab or tab, and the number of seconds between skips in seconds in the **Time between skips** field. All the time the clicker will watch the game, and if the wave goes on for more than 2 minutes, or something blocks it, then a re-start is made, and the timer is reset, as if this time has passed. Then it will pass the gift, if it comes out (if Solve gift is enabled), and play with skips as many waves as indicated in the **Battles between AB** field. Put on AB, and in a circle.

Note: When I say "with skips", it is not necessary oranges skip. It will decide to skip based on your Skip waves settings.

Break AB on 30 crystals. If enabled, clicker will look on crystals count while on AB. Once it sees that there is 30 crystals, it will drop the timer and wait for first cancel AB button, and start battles with skips as many waves as indicated in the **Battles between AB** field.

You can choose whether to play skips manually (with hands) – for this you have to enable **Manual waving between AB**.

Tip: you can set 999999 to **Time between skips**, and 0 to **Battles between AB**. In this case, the clicker will just look for the game, and if game crashes, or you got defeats and AB stopped automatically, it will start AB again. Don't forget to enable Solve gift.

Note: if gift appears during skipping between AB, it will solve gift, and exit from battle immediately to start again, because there are different parts of code in script, and its easier to keep it consistent this way. It will not affect of how many battles has already been started.

AB (wave canceling) – if enabled, then clicker will start wave, turn on AB, then wait for cancel AB button that appears between waves during AB, cancel AB and start next wave with AB. It will save time between waves in AB mode. Can be combined with Skip waves, Upgrade castle, Ad for coins.

You can set delays between clicks on AB button, switching to gab and start buttons to make it slower or faster in additional settings here.

Warn: Of course, you need to enable Solve gift, otherwise it will stop stupidly, and this functions will be useless.

Note: If ab loses constantly, the clicker will always restart autobattle.

Warn: when you want to change parameters, you have to restart clicker, because it reads parameters only when you start it.

Info: If you disable all playing modes like this:

The clicker will play waves manually.

Background mode – background mode. It will not steal the cursor with it and you can overlap the nox with another window, but you can't minimize the nox. Minimized windows are not rendered, and clicker will see just black screen.

Solve gift – solve the gift or not.

Restart on gift / 4 fails – if **Solve gift** is on, then with 4 fails in a row, it can either stop immediately with gift on screen, or restart grow castle and leave it so gold from colony will come. If you turn on this setting, it will restart, otherwise will stop with gift on screen. If **Solve gift** is off, do the same without trying to solve gift.

Info: The gift is solved as follows: after clicking start, the clicker saves 18 frames of heroes flying to their finish positions to the clicker's core folder: **"Gift_screen_X.bmp"**. After that, the clicker passes control to **slpx.dll** file, which takes these images and makes decision.

If you don't have lags during gift animation, the average success rate should be ~97%. You can enable saving screenshots of all solved gifts, for failed gifts, and enable logging for gift solver. All solved gifts are saved to **/Gifts/Gift_X/**. Failed gifts are saved to **/FailedGifts/Gift_X/** in core folder (it doesn't exist initially, and will be created automatically). There will be original frames (**Gift_screen_X.jpg**), and result frames (**Result_X.jpg**), where clicker will outline each hero on each frame, in clicker's opinion.

Max conflict count for fail number – When clicker solves gift, it initially solves all 8 heroes independently. Then it looks on all answers it got. If all answers are distinct, it means that there are 0 conflicts, and chances of success are high. But sometimes its hard to track heroes correctly, and there can be situation where solver thinks that different heroes ended up in same finish position. This is called **conflict**. When it happens, it means that at least one hero was solved incorrectly. After it, the conflict resolver comes in and searches for the optimal arrangement of finish positions to make them distinct. It reduces chances of wrong answer, and works good when there is 1-3 conflicts. But sometimes it can happen that 4 or more heroes are solved incorrectly initially, which can lead to 10+ conflicts in the end, and there is very high chance of wrong answer.

So, you specify the highest number of conflicts at which the clicker will make an attempt to solve gift for each number of the fail. If the number of conflicts exceedes it, the clicker will restart the game and will not make an attempt. In this case, the gift will reset, but there will be no fail.

<mark>Info</mark>: The maximal theoretical number of conflicts is 56.

On the example of the settings from the screen: as soon as the gift got out, we have 0 fails on our account, so the maximum conflict count at which the clicker will make an attempt to solve the gift is 100, that is, always absolutely. If this is a miss, then now there is 1 fail on the account and now if the solver shows more than 6 conflicts, then it will restart and will not make an attempt. In this case, the gift will reset, but it will only take 10 seconds to wait, and so it will restart until the conflict count is 6 or less. And so on: if it misses again, it will expect 2 or less conflicts, then 0.

On the right is indicated how many seconds you have to wait for the next attempt in case of a miss.

<mark>Info</mark>: I would recommend to start with 100, 6, 2, 0, and then edit like you want.
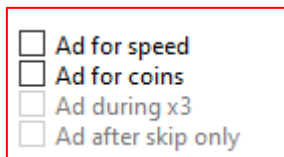
Ad for speed – watch ads for X3. It is called in the main menu of gc through store. Viewing ads is a separate algorithm. If the ad completed successfully, then the last time the ad was viewed will be written to the file **timerx3spd.txt** - this is necessary to save the last time the ad was viewed after the clicker stopped. Then after this hour the advertisement will not be checked. If the ad ends, it will not start and the clicker will automatically turn off viewing it.

<mark>Warn</mark>: If the time for watching ad has come, and clicker doesn't watch it, then write 0 inside **timerx3spd.txt.** It should solve the problem.

Ad for coins – ad for coins. After the wave, there is a delay of 0.4 seconds, after which it is checked whether the advertisement popped up. Viewed, and after that 5 waves are not checked, so as not to waste time between waves. If the ads are over, then each wave will be delayed by 0.4 seconds, and there is nothing you can do but disable viewing it.

Ad during X3 – whether to watch ads for coins during X3 with ads. You can turn it off to make the most of X3 from ads, or return it if you need to watch ads for coins as quickly as possible.

Ad after skip only – watch ads for coins only after skip 30 waves. There is more bonus gold, so there will be more gold from advertising. Cannot be activated without **Skip waves**.

<mark>Info</mark>: After the start of the ad, the clicker waits for a couple of seconds, and simply presses the right click once a second until it sees one of the options for the pause button that I entered into its algorithm, or until the game opens. If the advertisement has not closed in a minute, then a restart is made. I met only 3 options for pauses in advertising - and I entered them into the clicker. In fact, this is just checking the color of the pixels at the given coordinates, where the

continue button usually always appears - always the same color. If you have another ads UI, you can set **$fixed_ad_wait** setting to add constant wait after ad opening.

Heal altar – it will hit the altar when there is low hp, and the smith is either not in the build, or he is pressed already, if there is a smith, then it will press smith first.

Death altar – presses the altar on the dragon and dungeon - when the boss appears. Only works there.

Pw on boss – press pw on the bosses. When the wave reaches this point:
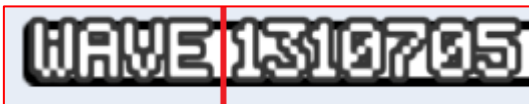
pw stops pressing, and when the hp bar of the bosses appears, it will be pressed after 200ms, as indicated on the screenshot. Then there will be higher chance to press all heroes at once on the bosses, and the wave will end faster.

Upgrade castle – upgrade castle parts for crystals. For correct operation, the upgrading parts must be on the floor specified in the **Floor**:

Info: By pressing, the tower that stands on that floor opens and it is upgraded, while there are >9 white crystals on the account, or until it has been upgraded 60 times in a row (if the tower has reached the maximum level). Tower upgrade is called every 10 battles. Check if it works correctly and doesn't spend oranges.

Warn: when you get closer to 10000 level, be careful, because clicker cannot determine which level it has right now, and it can potentially spend oranges, or the store will be constantly opening.

Screenshot:

☑ Items
☑ Runes

☑ On esc
☑ Long load
☑ Long wave
☑ After 10esc

☑ Nox load fail
☑ Nox main menu load fail
☑ Clear all fail

Screenshot.

In case of various errors, a screen is made. You can also screen the items and runes. All these screenshots are saved in the screens folder in the clicker's core folder.

Items / Runes: Items will be screened only those that go to inventory. This is done so that you can safely throw off the screen to someone, and not be suspected of a bot. The same goes for runes. Screenshots are saved with the name ItemB, ItemA…, Rune

On esc – if the sky is blocked in the game for 4s in a row, this situation is non-standard and the clicker starts pressing the right click. At this point, a **pressesc** screen will be made. Also with this setting, screenshots from errors in AB mode will be saved: **AB_error**, **AB_error2**. Such as a long wave or some sort of failure.

Right click is pressed until the sky opens, or 10 times - if after 10 right clicks did not open the sky - a screen is made from the checkbox After 10 esc.

Long load – after clicking on gc from the nox menu when re-entering the game, a timer starts for 20 seconds. If the gc menu has not opened in 20 seconds, a **restartgcloadfail / resetgcloadfail** screen is made (from the restart or reset functions, respectively). If the game always loads longer than 20 seconds, it would be necessary to somehow speed up this process, or change this moment in the code.

Long wave – In the **gc_bot_settings.txt file**, the $max_battle_length_ms variable specifies the maximum duration of the wave in milliseconds. If so much time has passed since the start of the wave, then a restart will be made and the **longWave** screen is saved (nox can freeze in the middle of the wave, I had it).

```
$max_battle_length_ms = 180000
```

Nox load fail – if after the reset nox did not start in 2 minutes, a **noxloadfail** screen is made.

Nox main menu load fail – nox menu loading error, that is, if it was not possible to open the main menu of nox on restart. The screen of **noxmainloadfail** is saved.
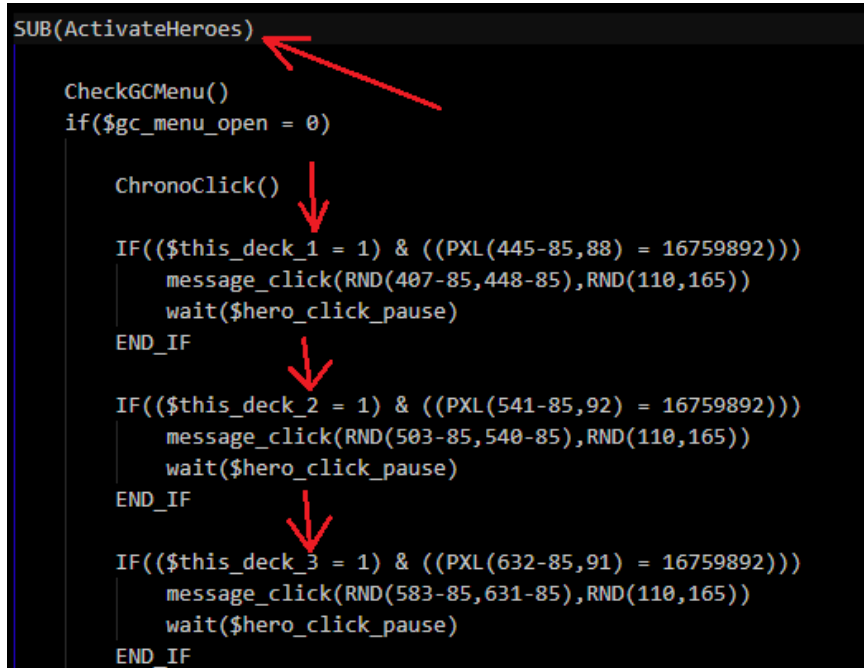
Clear all fail – could not find clear all button in nox on restart. The **clearallfail** screen is saved.

Aiming heroes setup

You need to go to gc_bot_lib.txt, find SUB(ActivateHeroes) or SUB(ActivateHeroesDun) there - normal or for dungeon.

There is a separate check for each hero. There are 15 of them:

```
SUB(ActivateHeroes)

    CheckGCMenu()
    if($gc_menu_open = 0)

        ChronoClick()

        IF(($this_deck_1 = 1) & ((PXL(445-85,88) = 16759892)))
            message_click(RND(407-85,448-85),RND(110,165))
            wait($hero_click_pause)
        END_IF

        IF(($this_deck_2 = 1) & ((PXL(541-85,92) = 16759892)))
            message_click(RND(503-85,540-85),RND(110,165))
            wait($hero_click_pause)
        END_IF

        IF(($this_deck_3 = 1) & ((PXL(632-85,91) = 16759892)))
            message_click(RND(583-85,631-85),RND(110,165))
            wait($hero_click_pause)
        END_IF
```

Numeration of **$this_deck_n** goes like that:

```
//Castle slots:

//                +-----+-----+-----+
//                |     |     |     |
//                |  1  |  2  |  3  |
//                |     |     |     |
//          +-----+-----+-----+-----+
//          |     |     |     |     |
//          | 13  |  4  |  5  |  6  |
//          |     |     |     |     |
//          +-----+-----+-----+-----+
//                |     |     |     |
//                |  7  |  8  |  9  |
//                |     |     |     |
//    +-----+     +-----+-----+-----+
//    |     |     |     |     |     |
//    | 14  |     | 10  | 11  | 12  |
//    |     |     |     |     |     |
//    +-----+     +-----+-----+-----+
//    |     |
//    | 15  |
//    |     |
//    +-----+
```

By doing this:

```
IF(($this_deck_6 = 1) & ((PXL(634-85,201) = 16759892)))
    message_click(RND(583-85,631-85),RND(203,276))
    wait($hero_click_pause)
END_IF
```
=>
```
IF(($this_deck_6 = 1) & ((PXL(634-85,201) = 16759892)))
    message_click(RND(583-85,631-85),RND(203,276))
    wait(0.1)
    message_click(1409-85,584)  // 5619175
    wait(0.05)
END_IF
```

After click the hero on position 6, the clicker will additionally click on the coordinates that are indicated in the second click. You have to pick them up yourself. The same is for each position. You can specify any combination of clicks. The main thing is to leave delays between clicks, otherwise the game will not respond on all of them.

Warn: Do not confuse ActivateHeroes and ActivateHeroesDun - the second one is intended for hit the heroes in the dungeon (there you need a more thorough check to see if the menu has opened or the timer on the dragon appeared).

Output files

Dungeon_statistics.txt – The number of items of different grades from each dragon. The numbers must be on their strings, otherwise the clicker will not be able to read them.

Informgift.txt – Each attempt to complete the captcha is recorded there:

```
+    h5p5, 24.06.2024 21:06:54, CC: 0, 0 restarts, 00:00:02:968
+    h2p8, 24.06.2024 21:07:12, CC: 0, 0 restarts, 00:00:02:979
+    h6p8, 24.06.2024 21:07:30, CC: 0, 0 restarts, 00:00:02:989
-    h4p8, 24.06.2024 21:07:47, CC: 5, 0 restarts
+    h5p5, 24.06.2024 21:08:00, CC: 0, 1 restarts, 00:00:10:110
+    h2p6, 24.06.2024 21:08:19, CC: 0, 0 restarts, 00:00:02:983
+    h5p8, 24.06.2024 21:12:19, CC: 2, 0 restarts, 00:00:03:298
```

+/- - whether the captcha passed. **h,p** - starting and finishing position of the hero, according to the clicker. Date, time, **CC** – conflict count, the higher **CC**, the higher chance of fail. The number of restarts, and the time from the appearance of the captcha on the screen. How gift is solved.

gc.log – Clicker log. Its actions are recorded there. From it you can understand the time, and possibly the cause of some errors. Its size is constantly growing, so you need to clean it periodically.

giftLog.log – Log from gift solver. There it prints top of probable finish positions for each hero, and result positions for them. You can turn off logging (**$gift_logOutput**).

Additional settings

The **gc_bot_settings.txt** file reads the settings from the **_main_settings.txt** file, and there are also variables that can only be set there. <mark>Warn</mark>: After changing values in the file, you have to open script editor and click apply.

$hero_click_pause – delay between hero clicks. If the value is too low, the game may not have time to respond to all clicks.

$battle_click_wait_ms – delay in milliseconds after clicking Battle, before removing the skip window, if too short, the clicker may either not remove it or hang in the menu and the exit window will pop up. You need to choose the most optimal value, or leave it as it is.

$wait_before_AB_open – if you set to play in AB mode with skips, or wave canceling mode, this is delay before pressing AB button (after pressing Battle button or closing skip window). If the AB button is pressed very quickly, and game couldn't response to this click, you can increase this value. Set a value that makes you feel confident that game always responds all clicks.

$wait_after_AB_open – this is wait before click on switch to gab button (or start button, if you set tab mode). Set a value that makes you feel confident that game always responds all clicks.

$wait_after_gab_open – this is wait before click start button after clicking switch to gab button – if you set to play with gab mode.

$orc_and_military_on_skip_only – if 1, then the orc band and military will only be able to be pressed on the 30 wave skip. If 0, then they will always be pressed.

$cleanup_interval – The interval between cleanups. When you play for a long time, the game starts lagging, and you have to make memory cleanup. Set how many seconds will be between cleanups (by default 10800 = 3 hours). The cleanup will be called between battles, when selected time has passed.

$I_have_3x – if 1, then clicker will not run if you want to watch ads for 3x, and if you are farming dragons, when item drops from the chest, will click on speed 3 times instead of 2.

$max_battle_length_ms – The maximum duration of a battle in milliseconds. If more time has passed since the start of any battle, the clicker will re-enter the game.

$max_restarts_in_row_for_reset – The maximum number of restarts attempts before nox reset.

$mimic_collect_percent – The chance of collecting mimic chest – 0 to 100 value.

$fixed_loading_wait – If you have very long loading of gc, write some constant wait that clicker will wait after opening gc.

$fixed_ad_wait – fixed delay after starting ad. If clicker cannot finish ad because it tries to exit it, increase this value.

$simulate_mouse_movement – if 1, the clicker will simulate mouse movement between clicks. Its unlikely that game is tracking mouse movement, but if you believe that it does, you can set this to 1. With it, it will run slower, because of cursor movement between clicks.

$log_CurrentBattleLength – if 1, it will constantly log the duration of the battle. Used for testing.

$gift_logOutput – print log from gift solver to **giftLog.log** or no.

$gift_saveScreenshotsAlways – if 1, will save screenshots from all attemts of solving gift to **/Gift/Gift_X/**. This folder is in clicker's core folder and will be created automatically if it doesn't exist.

$gift_saveFailedGiftScreenshots – if 1, will save screenshots from failed attemts of solving gift to **/FailedGift/Gift_X/**. This folder is in clicker's core folder and will be created automatically if it doesn't exist.

$gift_notFoundScreen – if 1, will save screenshots when clicker couldn't find crystal on gift. Screenshot will be saved with name "**giftNotFound**".

Pop-up messages

Popup messages only work with Background mode. For some events, you can enable or not:

$make_hints_gift – when gift appears.

$make_hints_cant_fight_dungeon – if cannot start dungeon battle.

$make_hints_clearallfail – cant see Clear all button when restarts.

$make_hints_emergstop – stopping. Happens on captcha, when Restart on gift is active.

$make_hints_escclick – if something blocks the game, and the clicker starts pressing right click.

$make_hints_long_gc_load – if the game takes too long to load (20 seconds + the value in **$fixed_loading_wait**). At the same moment, the clicker will quit and start the game again.

$make_hints_noxmainmenuloadfail – Nox main menu does not open. The main menu is determined by a white pixel in place of the strip from the play market, you shouldnt move it from there:

$make_hints_reset – If you had to reset nox. Nox reset is called if after maximum number of attempts to restart the GC in a row, the clicker has not open gc.

Info: try to check if popup messages appear at all by running this code in clickermann editor:
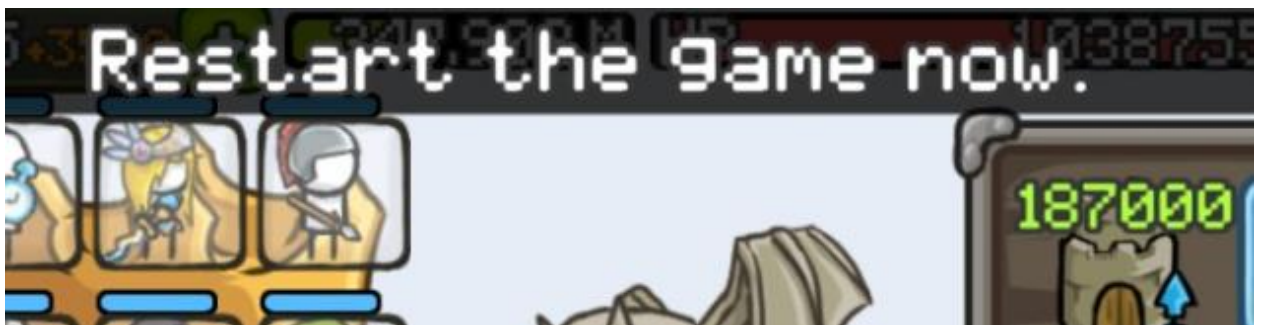
```
hintpopup("Test")
halt
```

Macro check (?) passing

Once such a message popped up. At this time, the script played and saved the image. It restarted the game, but account was banned after this.

And this is the most recent version of it:

The script will check this pixels:

if the top pixel is blocked by something, and the bottom one is free (has regular sky color), or there are green pixels on the top of game view, the script will consider it as a macro check, and will restart the game immediately. Then, it will keep making screenshots for some time (maybe something will happen then). After it, the script will continue playing. You can change this behavior. **gc_bot_lib.txt** -> **CHECK_ON_HINT()** function.

Problems solving
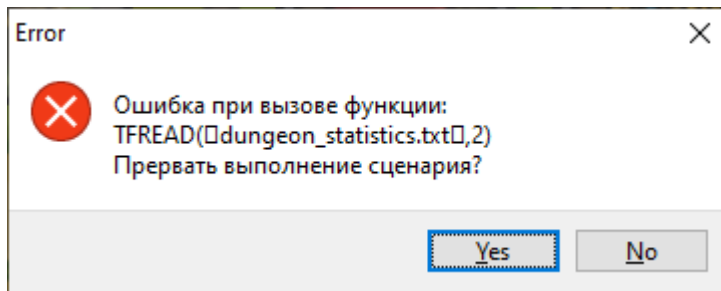
If you see this error while trying to start clicker or when item is dropped from dragon, it means that for some reason **dungeon_statistics.txt** was broken. You have to recover this file. For this just copy its initial contents from github.
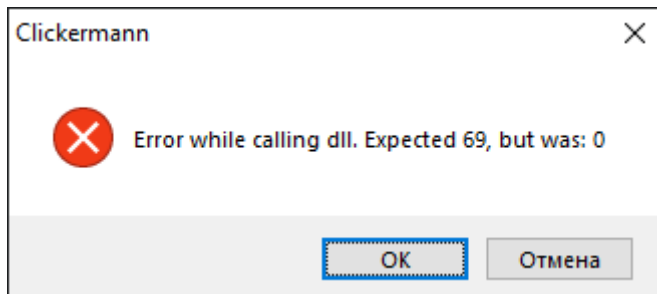
==

Check if **slpx.dll** and **opencv_world490.dll** are in clicker's core folder. If they are, it means that clickermann didn't call it for some reason. It can be because you use broken clickermann. Try downloading one from github repo. There is `**Clickermann v4.13 x64.rar**`, the clickermann is there. If the issue persists, I don't know what to do. All you can do is turn off solving gift.

==

If the clicker stops ads and cannot watch it to the end because of trying to exit it, it is probably because you have different ads UI that I didn't add to the clicker. To fix this, go to **gc_bot_settings.txt**, and change **$fixed_ad_wait** **variable** to 30 or 40, so the clicker will try to exit ad after this amount of seconds.

==

If the clicker doesn't watch ad for 3x: try to put 0 to the **timerx3spd.txt** file instead of the number that you have.

==

If the clicker cannot exit pause window or hero window, and clicks in some random place on screen, check if you have **this** enabled.

==

If the clicker keeps open exit window before start wave, or doesn't close skip window after starting wave, try to increase **$battle_click_wait_ms** value.

If the clicker cannot start autobattle correctly, try to increase **$wait_before_AB_open**, **$wait_after_AB_open** and **$wait_after_gab_open** values.

==

If the game opens too long, the clicker can consider it as gc freezing, and will keep re-entering grow castle. To fix this, set **$fixed_loading_wait** value.