VNFrame 2.0 Tutorial for Non-Programmer

- From The Beginning -

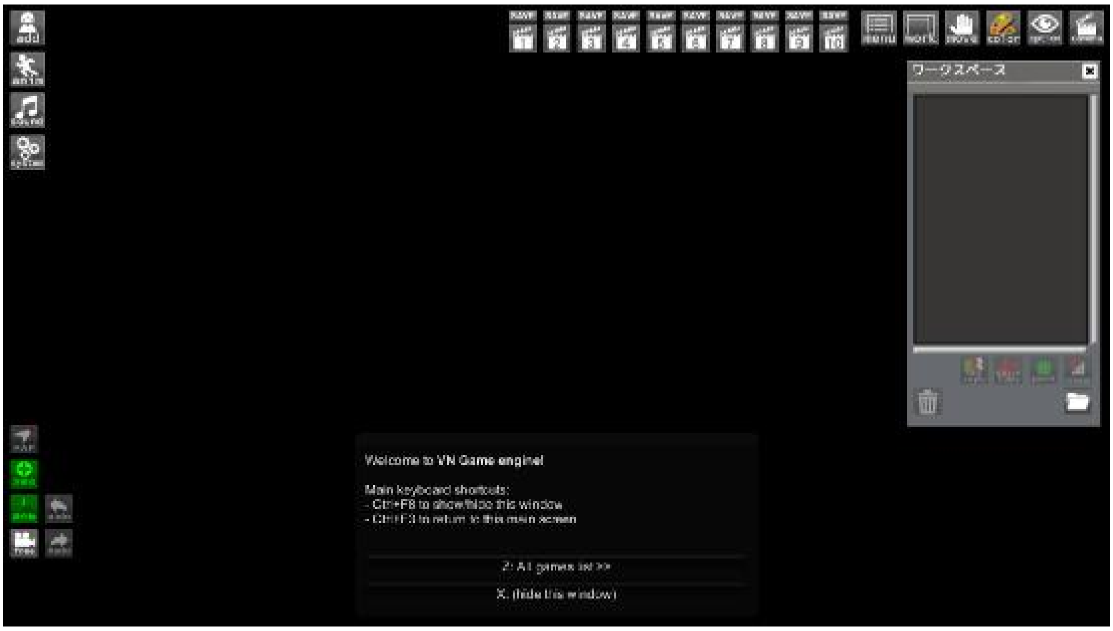
Introduce:

VNFrame is a framework base on vngameengine. It is used to control character and items in the scene. Since V2.0, an auto script template is included, that makes non-programmer can build strict-story-line game just use the 'Script Helper' provided by VNFrame.

In this document, a step-by-step example will show you how to make a game just like the "Demo: simple game" bundled with vngameengine. We choice HoneySelect Studio Neo here but you can use PlayHome Studio or Koikatu CharaStudio too, the operations are the same.

Preparation:

Install vngameengine and VNFrame of course! If you installed them properly, then start studio, there should be a window of vngameengine appears at the bottom of studio.



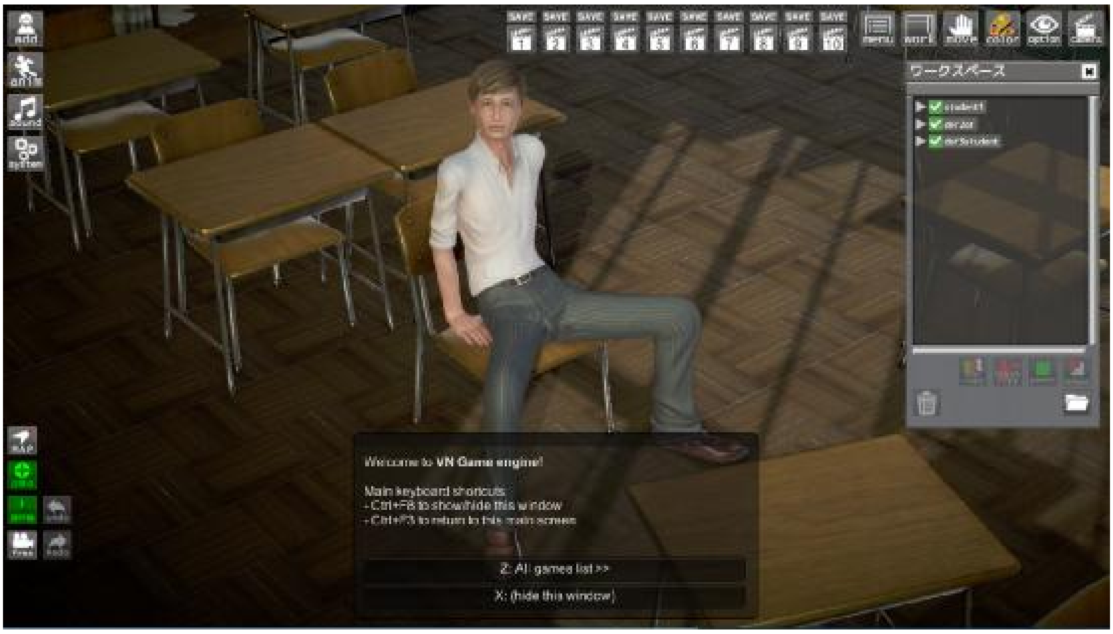
Welcome to VN Game engine!

Main keyboard shortcuts:
- Ctrl+F8 to show/hide this window
- Ctrl+F3 to return to this main screen

Now you can check the "Demo: simple game", it is in the All games list, to get an image of what kind of game we are going to make.

Setup:

First, we should prepare the game scene. As we plan to build a game just like "Demo: simple game", so we can just use the same scene file. Go to the scene folder, copy the "scene1.png" from "gamedemo" sub-folder to scene folder so you can load it as normal scene file. After load the scene, it should looks like this:



Welcome to VN Game engine!

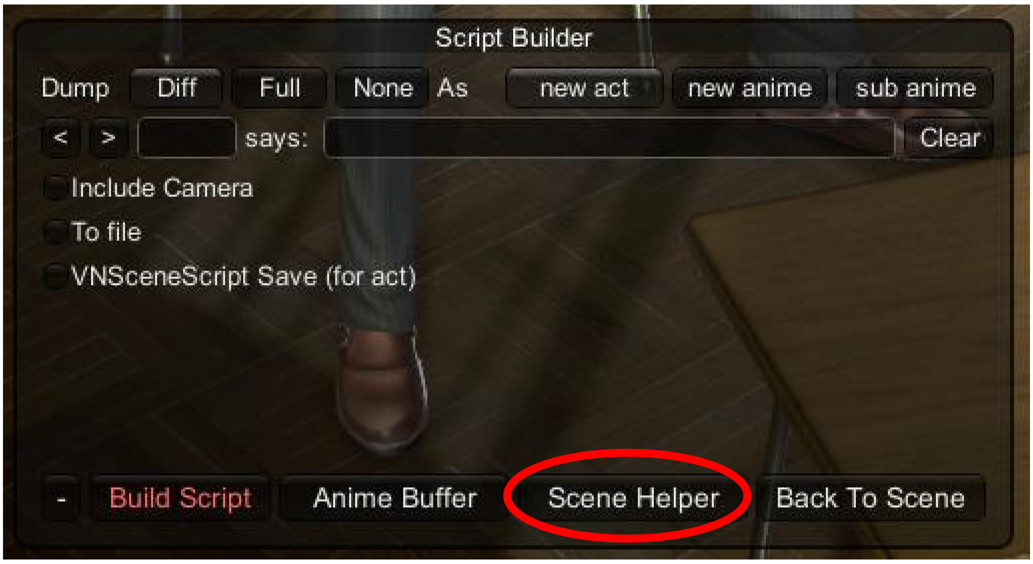
Main keyboard shortcuts
- Ctrl+F8 to show/hide this window
- Ctrl+F3 to return to this main screen

Z: All games list >>

X: (hide this window)

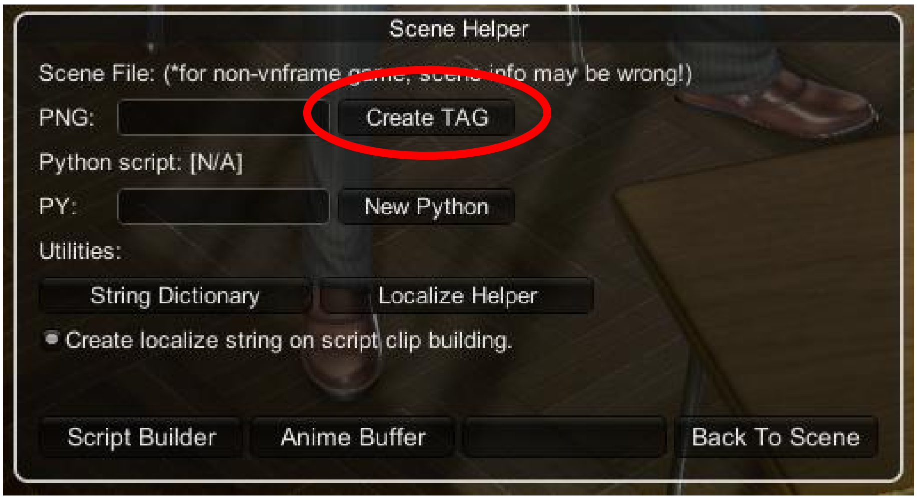
To use VNFrame, we need to "TAG" the actors now. As you can see there are three characters in the scene now. We should told VNFrame that we want to control them.

Press Ctrl-F5(by default), the vngameengine window should change to this:



This is the "Script Helper" window, it has three main pages: "Script Builder", "Anime Buffer", "Scene helper". Upon picture is "Script Builder" page by default. You can switch page using buttons at the bottom of window. If the "Script Helper" window did not show up, VNFrame may be not installed properly.

To tag character, we should use "Scene Helper". So click the button to switch to the "Scene Helper" page, it should like this:



Scene Helper

Scene File: (*for non-vnframe game, scene info may be wrong!)

PNG:

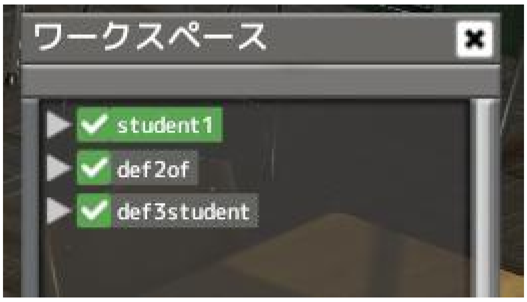
Python script: [N/A]

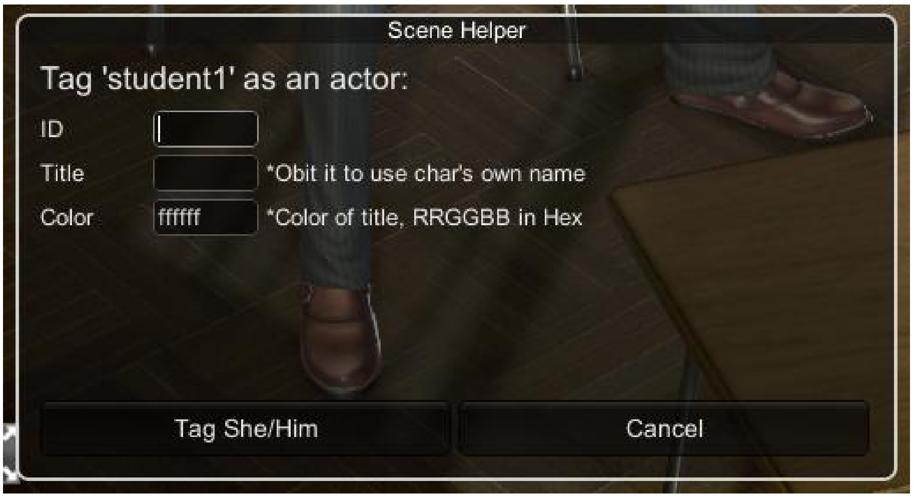
PY:

Utilities:

● Create localize string on script clip building.

Now, select the "student1", the male, in workspace tree view of studio. Then click the "Create TAG" button, a new page appears like this:





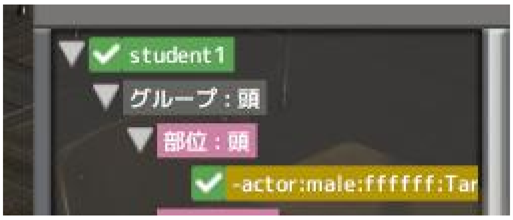
Scene Helper

Tag 'student1' as an actor:

ID

Title *Obit it to use char's own name

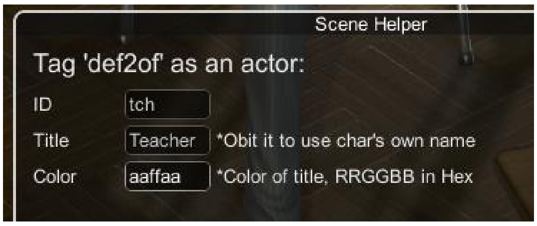
Color ffffff *Color of title, RRGGBB in Hex



Now set the ID as "male", so the VNFrame will know him as "male". Set Title as "Taro" or any name you like. If the title is omitted, the char's own name will be used as title, in the case it will be "student1". It is not a good name, so let's set a new name for him. The Color setting set a color of the title when displayed, RRGGBB in HEX format. If you don't know what that code means, you can pick a color from https://htmlcolorcodes.com/ and copy the HEX code (without #), or just leave it to "ffffff" (which means white).

After click "Tag She/Him", a tag folder will be added into workspace tree view as the picture above. If you want to change his name, you can edit this folder, or delete it and then TAG him again.

Do the same operation for "def2of" and "def3student", as the picture shows below:



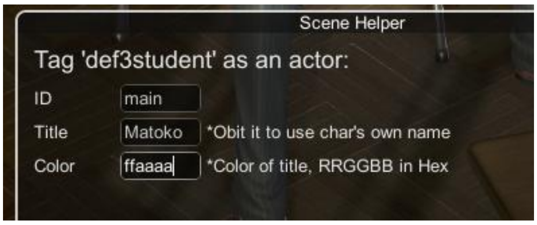
Scene Helper

Tag 'def2of' as an actor:

ID tch

Title Teacher *Obit it to use char's own name

Color aaffaa *Color of title, RRGGBB in Hex



Scene Helper

Tag 'def3student' as an actor:

ID main

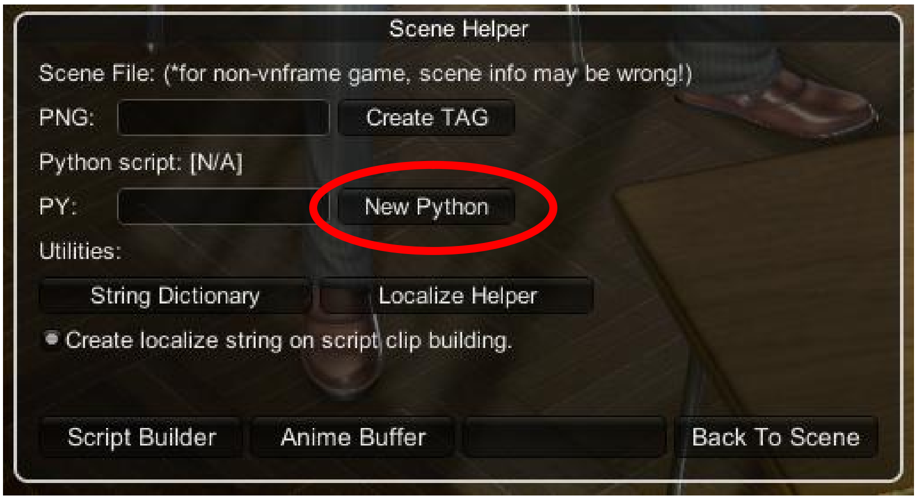
Title Matoko *Obit it to use char's own name

Color ffaaaa *Color of title, RRGGBB in Hex

After this, all characters are registered to VNFrame. We call them Actors now.

Create the game:

When actors are ready, let's create a new game by click "New Python" button in "Scene helper" page. Don't worry, you wouldn't need to edit it.



Scene Helper

Scene File: (*for non-vnframe game, scene info may be wrong!)

PNG: Create TAG

Python script: [N/A]

PY: New Python

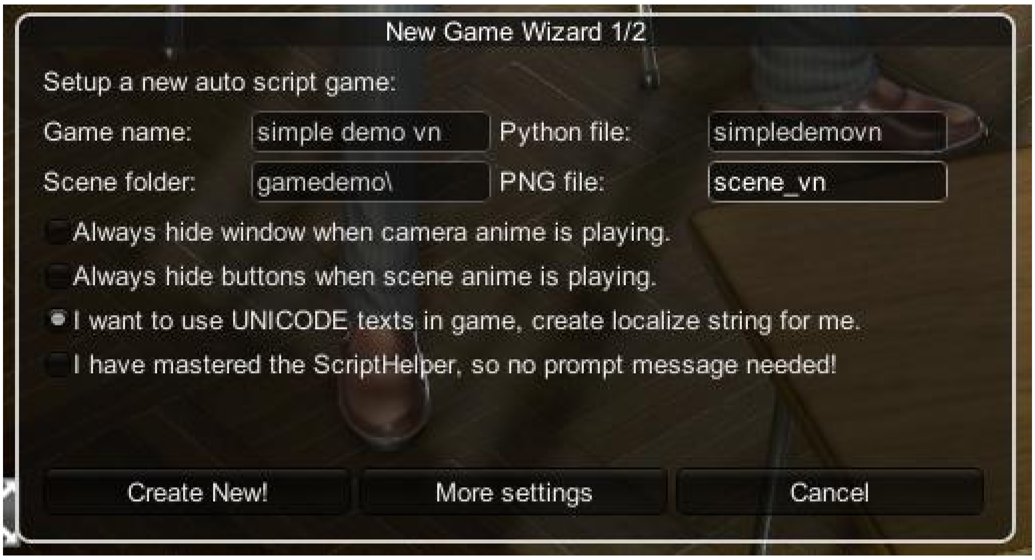
Utilities:

String Dictionary Localize Helper

● Create localize string on script clip building.

Script Builder Anime Buffer Back To Scene

And a new game wizard will show up:



New Game Wizard 1/2

Setup a new auto script game:

Game name: simple demo vn Python file: simpledemovn

Scene folder: gamedemo\ PNG file: scene_vn

○ Always hide window when camera anime is playing.

○ Always hide buttons when scene anime is playing.

● I want to use UNICODE texts in game, create localize string for me.

○ I have mastered the ScriptHelper, so no prompt message needed!

Create New! More settings Cancel

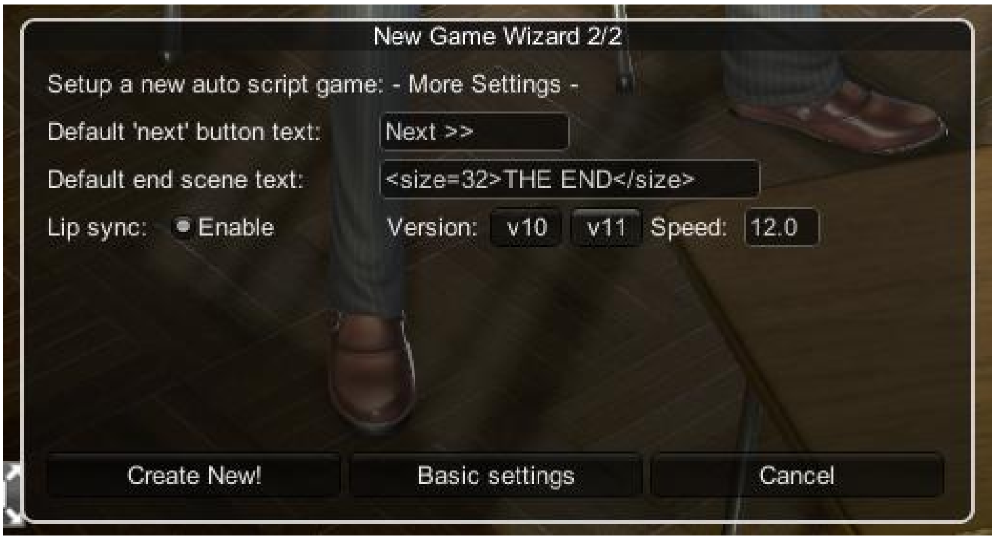
Name your game in Game name box, and set the path and name of scene file and python file. In the picture we set Scene folder to "gamedemo", but it is not a good practice. Please, build your own sub-folder before you create the game.

Always hide window when camera anime is playing and Always hide buttons when scene anime is playing check boxes are the global setting for hide window or button. If you don't want player interrupt you anime, you can enable them.

For some unknown issue, UNICODE text is not supported in PlayHome Studio and CharaStudio. So there is a workaround mechanism provided by VNFrame. If you need to use UNICODE, check the third checkbox. For details about UNICODE support refer to the end of this tutorial.

The forth checkbox will make the Script Helper run in Master Mode. Some prompt message will not show again. It is not for beginner.

Hit the "More Setting" button to next:



New Game Wizard 2/2

Setup a new auto script game: - More Settings -

Default 'next' button text: Next >>

Default end scene text: <size=32>THE END</size>

Lip sync: ● Enable Version: v10 v11 Speed: 12.0

At here, you can set the Default 'next' button text and Default end scene text. You can use UNICODE texts here, and if you do so, the UNICODE setting will be enabled automatically.

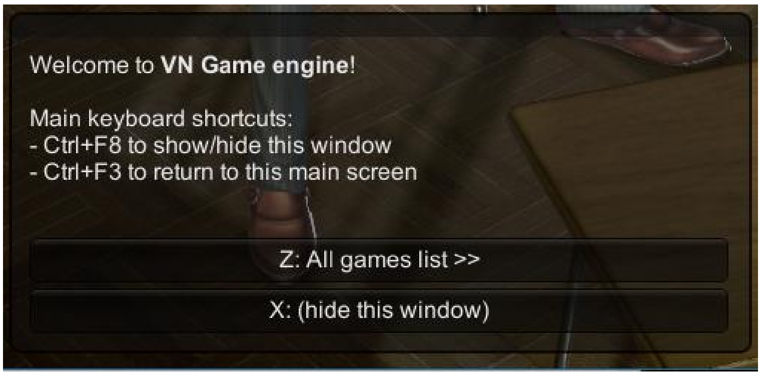
Lip sync makes your actor's mouth open and close when they are speaking. Recommended in most cases. v10 makes mouth open little, and V11 opens wider. Speed decides how fast the actor read. Default value 12.0 looks good for English, but if reading Chinese or Japanese, 6.0 may be better.

As all setting ready, click the "Create New!" button. If everything is OK, the VNFrame will tell you your PNG file and PY file were saved on disk.

Load our game:

Before we can go on to build game contents, we must load our game once. Load the game will let VNFrame initialize itself to make auto-script properly.

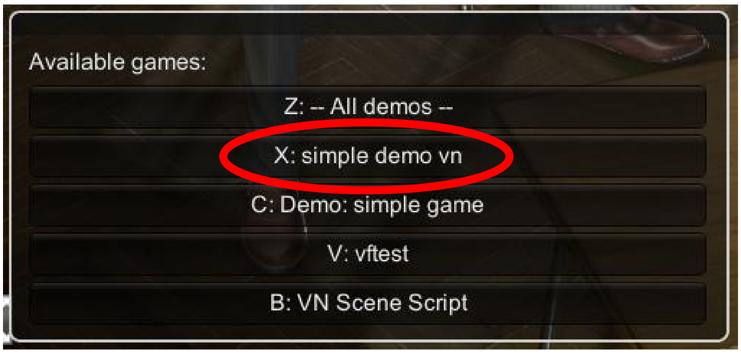
Hit the "Back to Scene" button or press Ctrl-F5 to return to vngameengine main window. Since we have not do anything in main window, it should looks like just loaded:



Welcome to **VN Game engine**!

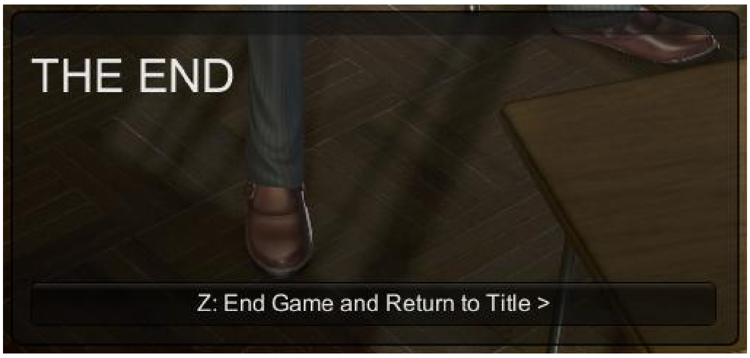
Main keyboard shortcuts:
- Ctrl+F8 to show/hide this window
- Ctrl+F3 to return to this main screen

If you loaded some game, press Ctrl-F3 to return to the main screen.
Hit the "All games list >>" button, and our new created game should be in the list now.



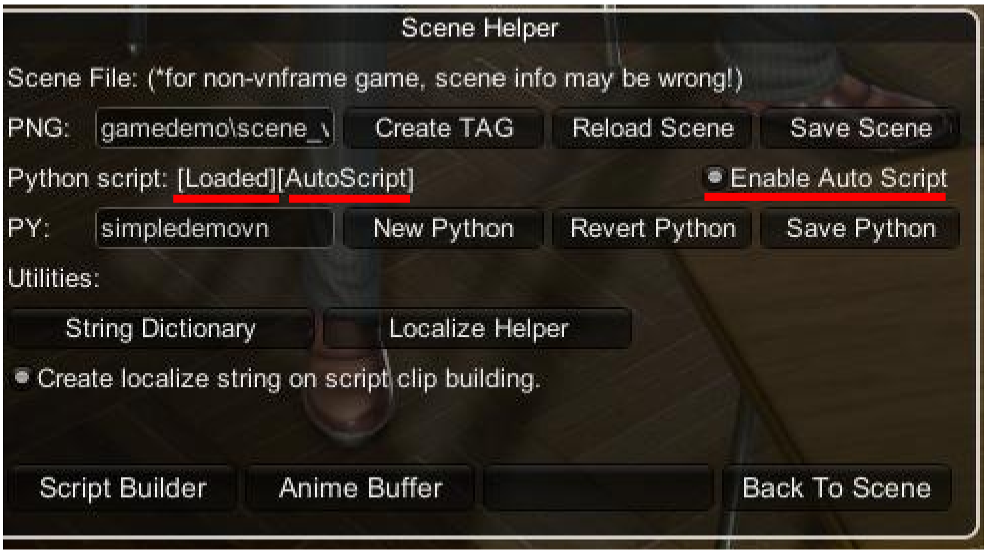
Available games:

So we just click our game and it should start normally, and shows THE END???



THE END

Yes, because we did not build game contents yet, so it jumps to the end scene directly.
Now open the script helper (CTRL-F5), switch to "Scene Helper" page and now it looks like this:



Scene Helper

Scene File: (*for non-vnframe game, scene info may be wrong!)

PNG: gamedemo\scene_\

Python script: [Loaded][AutoScript]

● Enable Auto Script

PY: simpledemovn

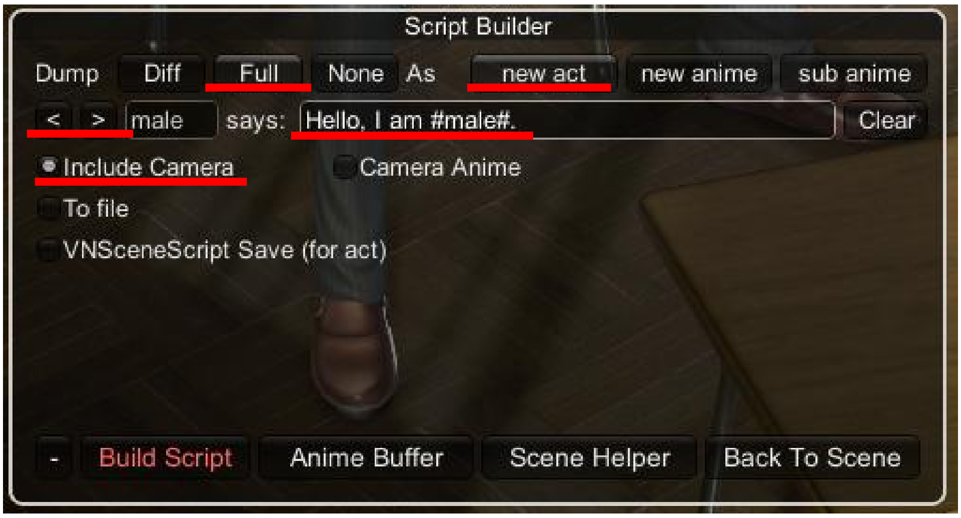
Utilities:

● Create localize string on script clip building.

Notice that PNG and PY file info are correctly set. And there are two label [Loaded] and [AutoScript]. [Loaded] means python contents were loaded into memory and can be edit. [AutoScript] means this game is build by auto script.
After them, make sure the Enable Auto Script checkbox is checked (it will be checked by default), and now we can start make the game.

Build game contents:

1st Scene: The scene just loaded is the first scene we need. So nothing need to be adjusted in scene. Press Ctrl-F5 to show script helper, switch to script builder page and set it like this, and then click the "Build Script" button.



male says: Hello, I am #male#.

● Include Camera

Tips: Dump Full for the first scene is a good practice. It can make the game start at a fix state, which can be different from the PNG file's load state.

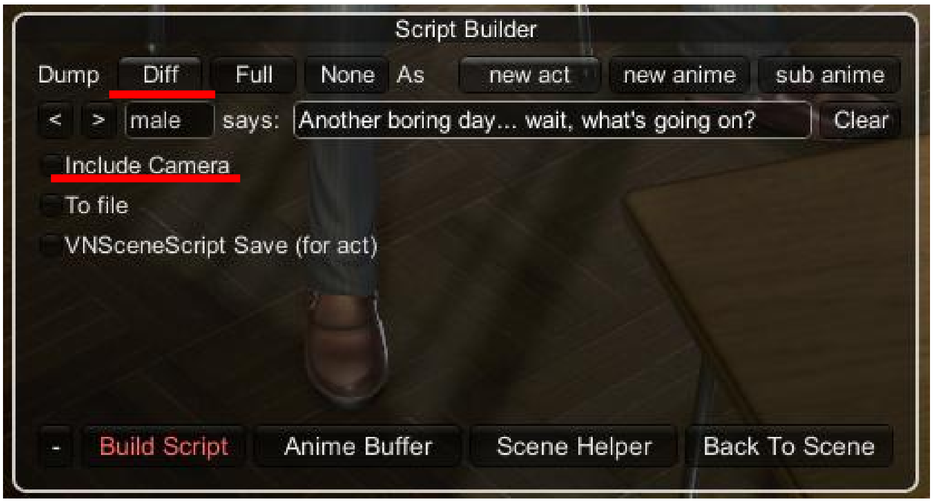
As new act means we will create an act here. An act will be performed immediately. In this simple demo we only create act.

Use the < > button in the second line can choose an actor to set dialogue, or choice 's' for system. Here we choice 'male', which is the id for the male student.

In the says box we can input the dialogue. Remember we enabled UNICODE support for our game, so we can use UNICODE texts if you want. Another merit of using UNICODE support is we can use *Quote* in texts now, as the *#male#* here. It will be replaced by the males name in game.

Check Include Camera will save the camera position in the script. This is the first scene and we don't need to do camera anime.

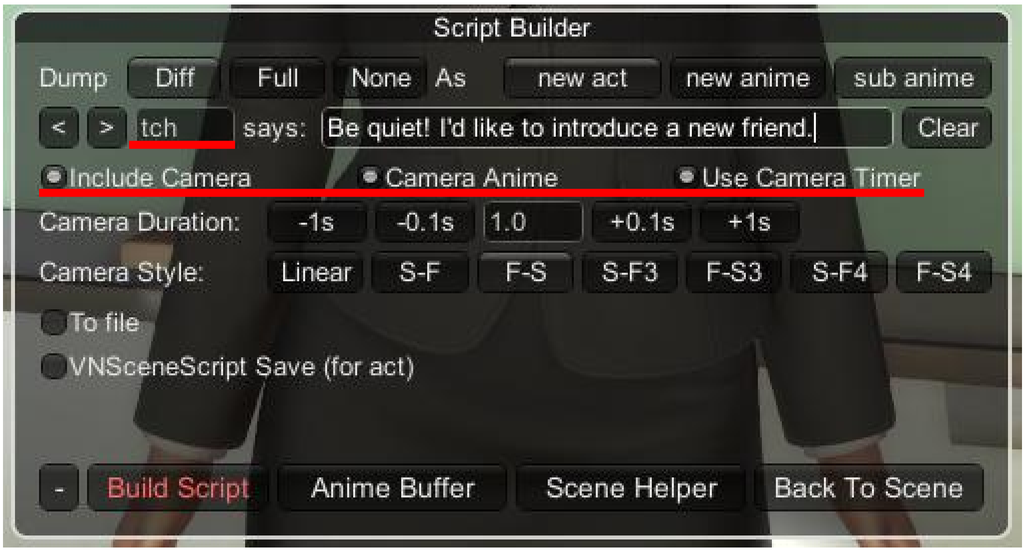
2nd Scene: Nothing need to be change, only our male student says something else. So it is very easy, set the Script Builder like this, and then click the "Build Script" button.



male says: Another boring day... wait, what's going on?

Tips: We use Diff instead of Full from now. Diff will set the VNFrame to detected what is changed and perform them for us, not set everything in the scene, so it should be faster. And for this scene, nothing is changed actually, so the camera is not necessary too.

3rd Scene: Now let's change the camera to aim at the teacher, and make teacher say something. Set the script builder like this and hit "Build Script".



Script Builder

tch says: Be quiet! I'd like to introduce a new friend.

● Include Camera ● Camera Anime ● Use Camera Timer

Camera Duration: 1.0

○ To file

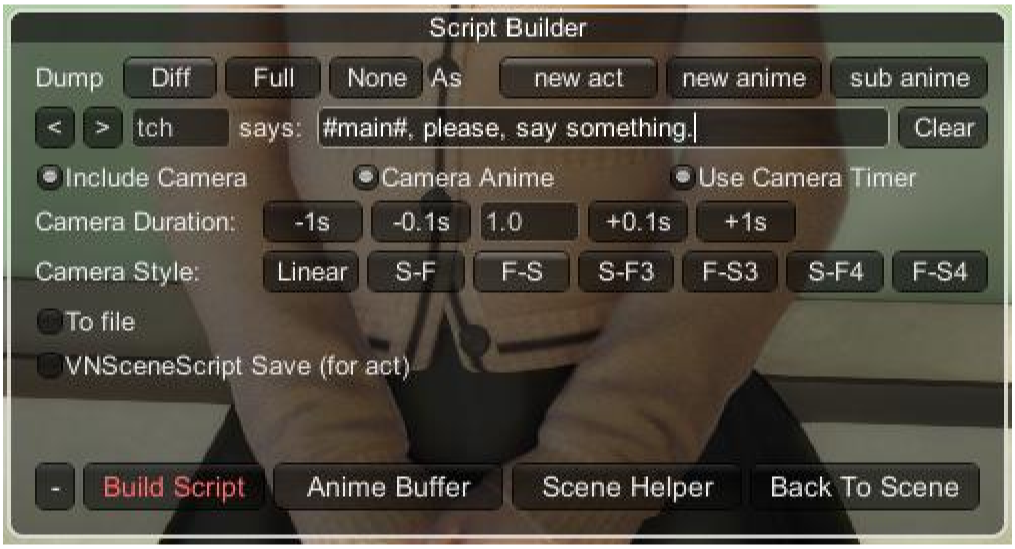
○ VNSceneScript Save (for act)

Tips: This time we changed the camera, so don't forget to include camera. And we want the camera move to teacher smoothly, not jump to her, so check the Camera Anime option.

Camera can be animated by scene anime timer or camera timer. Here we dump as new act but not anime, so the scene anime timer is not available now, and you cannot uncheck Use Camera Timer option.

When using camera timer, you can set camera duration and camera style. Duration is how long the anime lasts in seconds. Style can be choose from Linear, Slow-Fast, Fast-Slow, Slow-Fast3, Fast-Slow3, Slow-Fast4, Fast-Slow4. Just try them out by yourself.

4th Scene: Camera move to our heroine, but still teacher speaking. Set it like this and hit "Build Script".



Script Builder

tch says: #main#, please, say something.

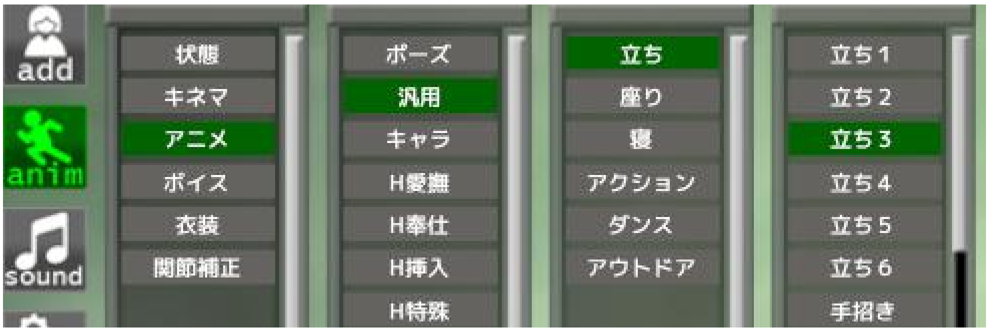
● Include Camera ● Camera Anime ● Use Camera Timer

Camera Duration: 1.0

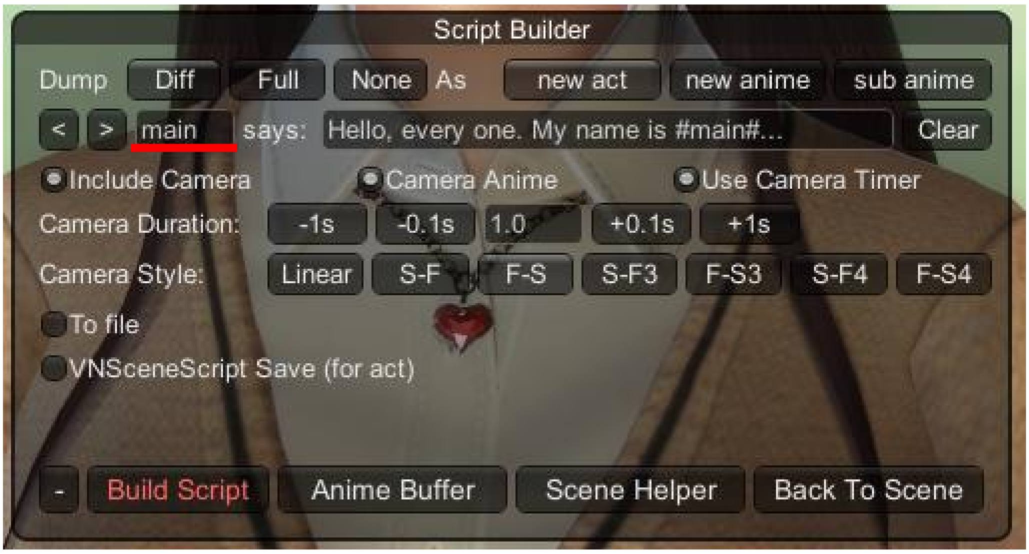
○ To file

○ VNSceneScript Save (for act)

5th Scene: Our heroine speaks now. It is boring to just speaking, so let's change the pose of her. Select her and change the anime to something like this:



Then change the camera and do another camera anime, set like this and hit "Build Script"



main says: Hello, every one. My name is #main#...

● Include Camera ● Camera Anime ● Use Camera Timer

Camera Duration: 1.0

Camera Style:

○ To file

○ VNSceneScript Save (for act)

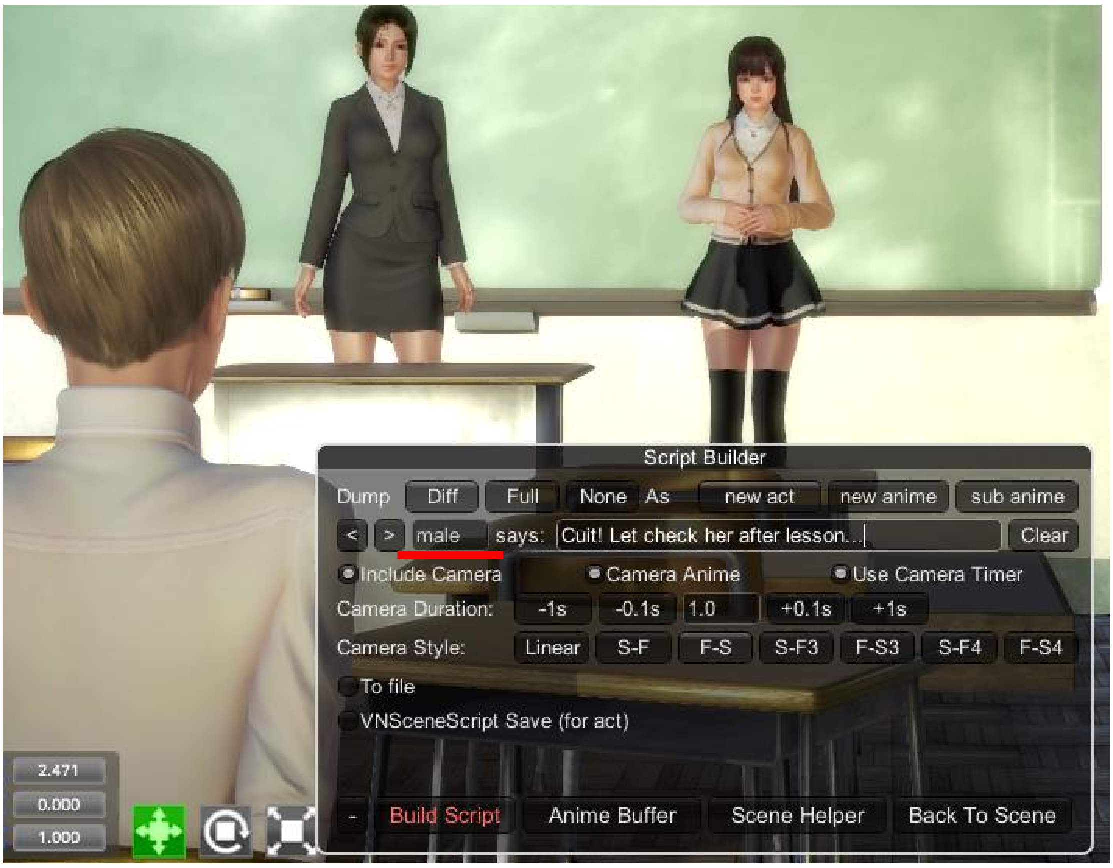
Tips: Now, when the female student starts saying her texts, she will change the pose at the same time. Again, we choose to dump as new act, so every change we made will be performed at the same time. If we choose dump as new anime, will her change pose smoothly? Sorry, the answer is NO, not that simple, additional work needed. We will talk about how to make that happen in another tutorial.

Don't forget to set the speaker to 'main', or a wrong name will be displayed and her lip will not move.

6th Scene: Say something else, take another pose, move camera again, you can make it. Just don't forget to hit "Build Script".

Tip: When you setting the scene, you can use the minus button to shrink the script helper window. Or press Ctrl-F8 to hide/show the window.

7th Scene: Set the camera to the back of male student, input his words, set it like this then hit "Build Script", nothing special.



male says: Cuit! Let check her after lesson...

○ Include Camera ● Camera Anime ● Use Camera Timer

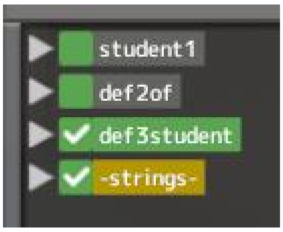
Camera Duration: 1.0

○ To file

○ VNSceneScript Save (for act)

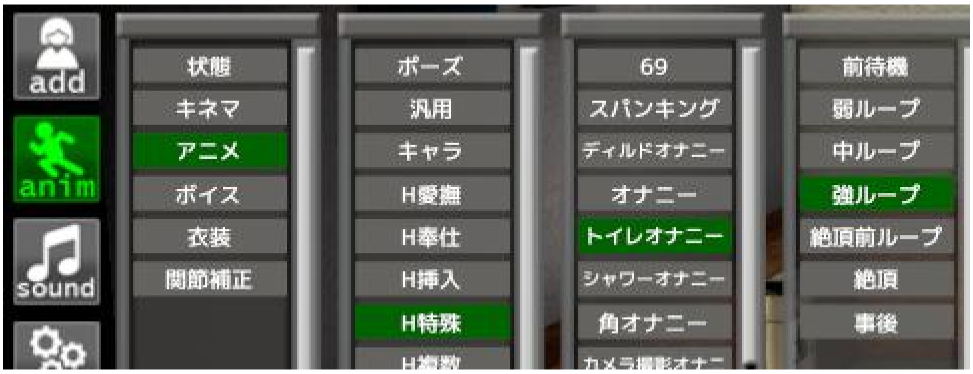
8th Scene: Now this game comes to the final scene. We need a lot of change in this scene.

- Uncheck the male student and teacher in workspace tree view to hide them.



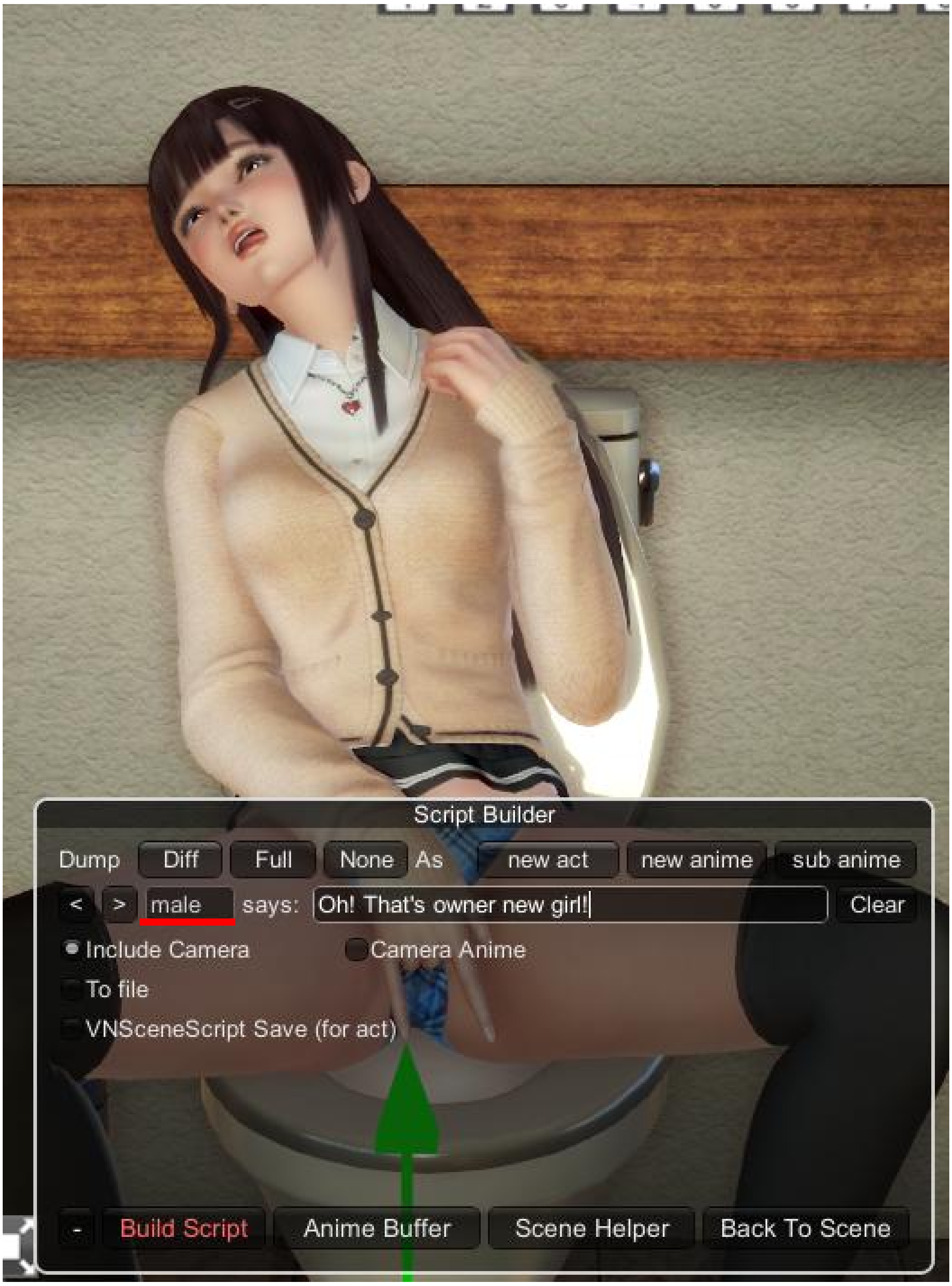
▶ □ student1
▶ □ def2of
▶ ✓ def3student
▶ ✓ -strings-

- Change the map to toilet.
- Change the female student's anime to the one shows below:



- Move and rotate her to sit on the toilet.
- Set her mouth and eyes to the one you like.
- Set the camera to the right position.

And then, in script helper, set like this and hit "Build Script" button:



male says: Oh! That's owner new girl!

● Include Camera ○ Camera Anime

□ To file

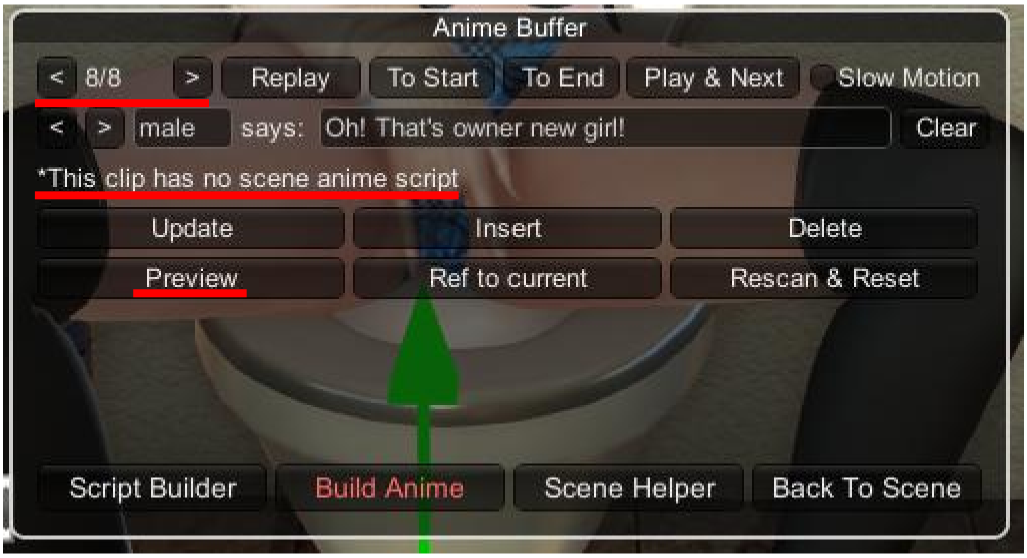
□ VNSceneScript Save (for act)

Tips: The sequence you edit your scene does not matter, only the state when you hit "Build Script" button will be dumped by the VNFrame Script Builder.

Though the male student is hidden, he can speaks as normal.

Check and Save our work:

We just dump 8 scenes, but where are they? They are in the "Anime Buffer" now. Switch to anime buffer and it should looks like this:



male says: Oh! That's owner new girl!

*This clip has no scene anime script

The dumped scenes are called clips here. The "8/8" in the top-left corner means we have 8 clips in buffer now, and now the 8th clip is selected. It is dumped as new act, so it has no scene anime script in it. And if you click the "Replay" button, no anime was played. "To Start" button will set the scene back to the start state of 8th clip, which is the end state of 7th clip. "To End" button will act as "Replay" button, they set scene to the end state of current selected clip.

To check all clips we built, you can select the first clip and click "Play & Next" button to check them one by one. Or another way, use the "Preview" button, then choose "Preview All". Preview is much more like a really game behavior, except the *quote*(for example #male#) in texts are not replaced yet.

From VNFrame 2.0, clips can be update/insert/delete after it was putted into the anime buffer. We will learn about it in another tutorial. For now, we finished our work and click the "Build Anime" button. System will tell us our anime is dump into a function, since we will not edit python file, just let it be.

Beware! our work are still in memory now, not saved on the hard disk! We must save the scene PNG and python script. "Scene Helper" can help us.

But wait, after we checked our scene, the current scene should be the toilet scene. Let's set the scene to the first scene then save the PNG file. To do this, switch to "Anime Buffer" page, select the 1st clip and then click the "Replay" button, now the first scene should be restored.

Then switch to "Scene Helper" page, use the "Save Scene" button and "Save Python" button to save our work. Now, you can reload the game and enjoy it!

Tips: If you found something goes wrong before you save the files, you can use the "Reload Scene" and "Revert Python" button to revert everything. After you save the files, you can go to the scene PNG folder (userdata\studioneo\scene\<your scene dir>\), there should has a .bak file of your PNG file, restore it manually. Python script has the same .bak file, in plugins\console\lib folder. But remember, VNFrame only provide one backup for last overwrite, you must backup the file yourself if more version is needed.

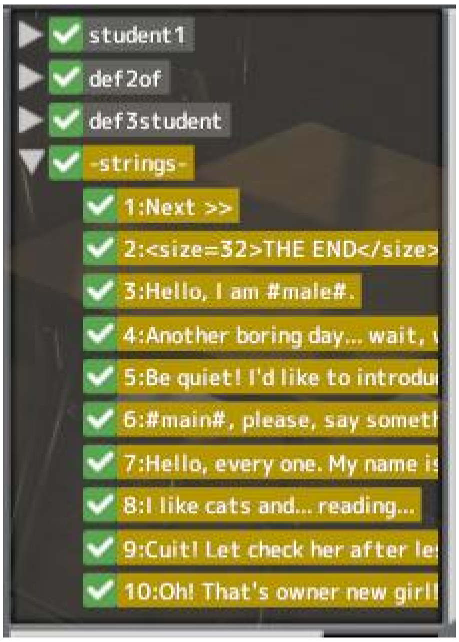
Summary and Tips:

To use VNFrame Script Helper build a game without any coding, you should follow these steps:

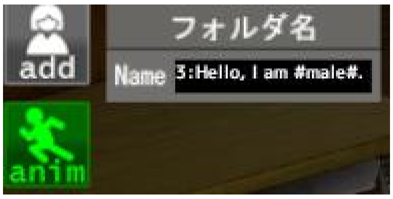
1. Make a scene, TAG which character you want to control as Actor, and those item/folder you want to control as Prop. You don't need to TAG those character/item/folder that will not change in you scene.
2. Create the game. Choose the setting carefully if you don't know how to edit python at all.
3. Reload the game. It is necessary. Because there are some important codes in the template file which helps script helper works properly.
4. Create you game scene by scene. Set the scene, check the setting and then click the "Build Script" button. Current scene status will be save into the "Anime Buffer". If you found something wrong, you can go to the buffer and delete it now. Dump "Full" for the first scene in anime and "Diff" for the rest scenes is recommended.
5. In this tutorial, we modified actor's anime, position, rotation, and changed the map. Besides, almost any setting on actor is supported by the VNFrame now. If you set a voice list for actor, she will say that! For a prop, visible, position, rotation, scale, color and anime speed is supported. For system, map, BGM and background PNG setting is supported, you can set built-in BGM or your own wav sound for BGM, change them between scenes, to make you game more great!
6. And by now, we build every script as act, the technique to build as anime and sub-anime will be introduced in another tutorial.
7. After built some scenes, go to the anime buffer to check them in sequence. Preview them again and again to check if everything in right position. If you are satisfied, click the "Build Anime" button to save them into python.
8. If you want to continue build you scene after used "Build Anime", you should set the scene to the last of previous anime, then click the "Rescan & Reset" button in anime buffer page. Then the buffer will be clean up and you can build new scenes from current scene. You new anime will be dump to another function when you click "Build Anime" again, and the new added function will be connect to the previous one, played continuously in game.
9. Add a new Actor/Prop in the middle of create scene is not recommended. If you must, click the "Build Anime" to save your work to python, create a new TAG to register Actor/Prop, then use the "Rescan & Reset" button in anime buffer. It will rescan TAG to make your new Actor/Prop works, but will clear all clips in buffer so you must save them into python before reset.
10. Don't forget to save your scene file and python file! Or everything will lost!! After load an auto generated game, you can always add new scene into it. Just run to the end scene of the game, then use the "Rescan & Reset" button once, and you can continue create scenes from current states.
11. Actor and prop can be replaced after game is built. For example, you can replace the female student in this the scene with someone else you like. Because replace character will not delete the TAG under her, she will still act in the same way. If you omitted the title when TAG the female student, VNFrame will use the character's own name. So if you change the character, her name changes too.

About UNICODE support:

For now, the python file in PlayHome and Koikatu CharaStudio could not support UTF-8 encode. VNFrame provide a workaround mechanism for using UNICODE texts: save the texts in the scene file instead of in the python file. You may noticed, in the above tutorial after we dumped our anime, there is a new folder in the workspace tree view called "-strings-". And if you expand it, it contents a lot of strings and looks like a list. Actually it's a dictionary. The number before the texts is the ID for it. In this tutorial, script helper build them for you automatically. If you want to modify it, you can just add/delete/modify them because they are just empty folders. like this:



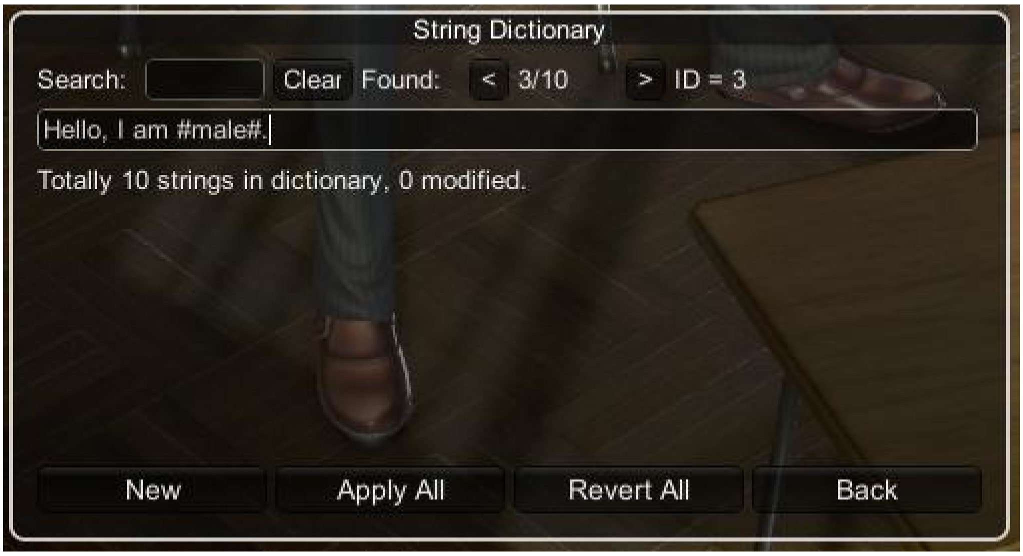
student1
def2of
def3student
-strings-
1:Next >>
2:<size=32>THE END</size>
3:Hello, I am #male#.
4:Another boring day... wait,
5:Be quiet! I'd like to introdu
6:#main#, please, say somet
7:Hello, every one. My name is
8:I like cats and... reading...
9:Cuit! Let check her after le
10:Oh! That's owner new girl!



フォルダ名

Name 3:Hello, I am #male#.

But, the input box for folder name is so small. So VNFrame provided an useful tool called "String Dictionary", you can access it from the "Scene Helper" page. It looks like this:



String Dictionary

Search: Clear Found: < 3/10 > ID = 3

Hello, I am #male#.

Totally 10 strings in dictionary, 0 modified.

This tool is an editor for string dictionary. You can new or modify strings here. It has a multiline input box so you can input a lot of words.

Input some keyword in Search box, it could found all the strings contains it. Or input a ID to jump to specified string. Clear the search can make it browse over all strings by click < or > button. After you new or modified a string, use the Apply or Apply All button, then the strings will be write to the -strings- folder. Click Revert All button will rebuild the string dictionary from -strings- folder. Remember, after you applied your change, you must save the scene file (not the python file) to save them on hard disk!

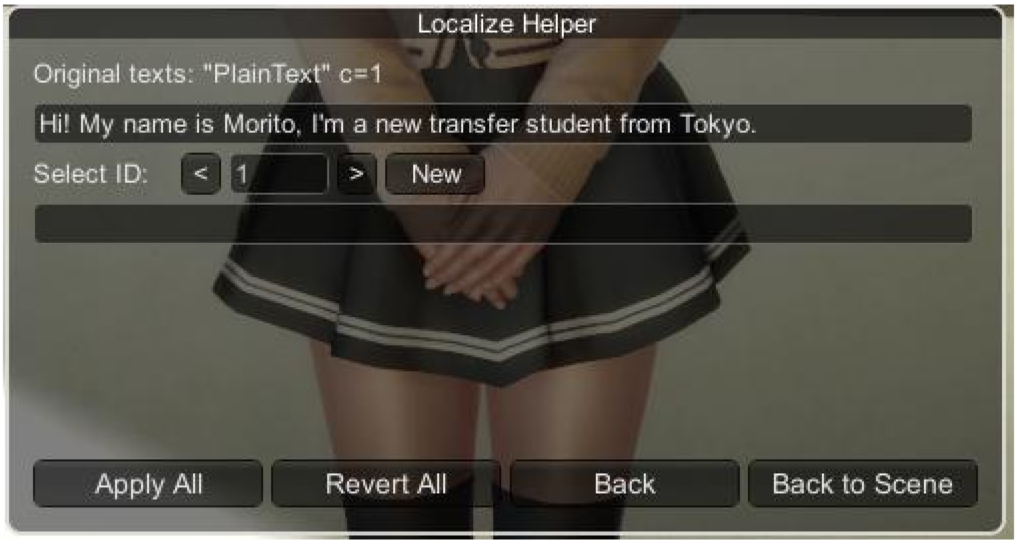
Even you are only use English in the game, UNICODE support is recommended if you don't want to edit python script. It allows you to edit the texts in scene file.

And, by using the UNICODE support, you can also use the *Quote* in texts. In this tutorial we input some texts like "#main#", it is a quote. It will be replaced by other texts in the game (no in the preview). Quote format is very simple, just #ID#, where:

- ID could be an actor's ID, and it will be replaced by the actor's title.
- ID could be a prop's ID, and it will be replaced by the prop's name.
- ID could be a string's ID, and it will be replaced by the string's text. But if the string contains another quote, it will not be replaced anymore.

Quote is especially useful for actor's title. Because you can replace the actor after build the game, and use quote can make the actor's name in the texts changes with the actor.

By the way, there is another tool "Localize Helper" in scene helper. It looks like this:



Original texts: "PlainText" c=1

Hi! My name is Morito, I'm a new transfer student from Tokyo.

Select ID: 1

This tool is mainly for translate a game into another language. If the original game is already used VNFrame and string dictionary, you can just edit the strings in dictionary. If it is not, this tool will help you replace the plaintexts in the python with a new string in dictionary. But some additional work must be done to make the original game work with VNFrame first, and that needs python programming skill, so we will not talk about it here.

Wish this tutorial helps!