How to create a girlpack

To help you in creating the many files and directories needed, there are two downloadable "pre-made" folders. One of them is for a girl without any interactions, the other for a girl with interactions.

You will need to rename the main folder to the name of the girl you want to add, fill the different files and put the videos you want to see in the game in the correct folders.

The same needs to be done for the interaction if you downloaded the pack containing interactions.

If you want to create the pack from scratch, you will first need to create a main directory named after the girl you want to add to the game. In this main directory, you must add four sub-directories, named "images", "JSON_data", "texts" and "videos". The names of those folders are not case sensitive but be sure to write them correctly.

The main directory must be put in the "Girlpacks" folder.

"images" folder

This folder should contain two files named "portrait.jpg" or "portrait.png" and "portrait_small.jpg" or "portrait_small.png"(the extensions must be present). The first file is the large portrait displayed during the recruitment and planning screens that shows the girl in her entirety. The second file is the small portrait used to drag & drop in the booth game and the planning. In both cases, if the file is not present, the game will display a default image.

Important: the "portrait.jpg" (or png) should be in a 2:3 format with a width of at least 600 pixels and a height of at least 900 pixels. The "portrait_small.jpg" (or png) should be in 1:1 format (a square) with a width and height of at least 120 pixels. It can be different in both cases, but the image could be cropped or stretched in a weird way.

It should also contain two folders named "Lessons backgrounds" and "Lessons portraits" if the lessons for this girl need specific backgrounds and portraits.

"JSON_data" folder

This is the only folder that MUST be present in the girlpack. If it is not, the game will NOT load the girl (and display an error message at launch).

This folder should contain only one file called "Description.json" (extension must be present). It contains the description of the different characteristics of the girl. The file must begin with "{" and end with "}". Each "value_name: value" pairs must be separated with a comma. The only time there must not be a comma is for the last field.

The format of each value is **always** " "value_name": value " with value_name always being between quotation marks and case sensitive and its value being either a string of characters (generally NOT case sensitive) between quotation marks or a number without quotation marks or true/false without quotation marks. There are a number of variables but they all have a default value except the "name" field. **It is the only field that MUST be in this file. Its value must ALWAYS be EXACTLY the same as the name of the main folder of the girlpack. It IS case sensitive.**

The other fields are less important because they have default values. Here is their details:

"height" → represents the height of the girl in centimeters.

          Expects a number.

"bustType" → represents the bust type (size) of the girl.

          Five possible values: "tiny", "small","medium", "big" or "very_big".

"eyeColor"→ represents the color of the eyes of the girl.

          Five possible values: "black", "brown", "blue", "green" or "hazel".

"hairColor"→ represents the color of the hair of the girl.

          Four possible values: "black", "brunette", "blond" or "red".

"bodyType"→ represents the body type of the girl.

          Five possible values: "petite", "athletic", "average", "voluptuous" or "thick".

"skinComplexion"→ represents the complexion of the skin of the girl.

          Three possible values: "light", "olive" or "dark".

"age"→ represents the age of the girl.

          Expects a number above or equal to 18.

          Any number below 18 will be treated as 18.

"dancing"→ represents the value of the dancing skill of the girl before hiring.

    Expects a number between 0 and 100.

    Any number below 0 will be treated as 0.

    Any number above 100 will be treated as 100.

"posing"→ represents the value of the posing skill of the girl before hiring.

    Expects a number between 0 and 100.

    Any number below 0 will be treated as 0.

    Any number above 100 will be treated as 100.

"foreplay"→ represents the value of the foreplay skill of the girl before hiring.

    Expects a number between 0 and 100.

    Any number below 0 will be treated as 0.

    Any number above 100 will be treated as 100.

"oral"→ represents the value of the oral skill of the girl before hiring.

    Expects a number between 0 and 100.

    Any number below 0 will be treated as 0.

    Any number above 100 will be treated as 100.

"sex"→ represents the value of the sex skill of the girl before hiring.

    Expects a number between 0 and 100.

    Any number below 0 will be treated as 0.

    Any number above 100 will be treated as 100.

"group"→ represents the value of the group sex skill of the girl before hiring.

    Expects a number between 0 and 100.

    Any number below 0 will be treated as 0.

    Any number above 100 will be treated as 100.

"popularity"→ represents the popularity of the girl before hiring.

    Expects a number between 0 and 100.

    Any number below 0 will be treated as 0.

    Any number above 100 will be treated as 100.

"openness"→ represents the openness of the girl before hiring.

    Expects a number between 0 and 100.

    Any number below 0 will be treated as 0.

    Any number above 100 will be treated as 100.

"moneyCost"→ represents the cost in money to hire this girl.

        Expects a number above or equal to 0.

        Any negative number will be treated as 0.

"reputationCost"→ represents the cost in reputation to hire this girl.

        Expects a number above or equal to 0.

        Any negative number will be treated as 0.

"influenceCost"→ represents the cost in influence to hire this girl.

        Expects a number above or equal to 0.

        Any negative number will be treated as 0.

        A girl with an influence cost above 0 will only be seen in the recruitment when the mafia services are available.

"inLottery"→ represents the fact that this girl is unlockable via the lottery service.
        Expects either **true** or **false** (without quotation marks).

All of these fields have default values, and if the conditions listed above are not respected, or if the field is not present in the "Description.json" file, the default value will be used.

I **strongly** advise to use the pre-made file and just modify the values accordingly. In this pre-made file, the fields are at their default values (except for the name).

Finally, each performance has a field associated with it that allows to specify each performance a girl won't do. These fields are all called "do" then the performance name (complete list detailed below). The default value for these fields is **true**, meaning that by default the girl will do the associated performance. Therefore, you only need to specify the performances **she won't do** by adding the correct field and giving it the value **false**.

List of "do" performances:

"doDance" → for the basic Dance performance

"doDanceCloser" → for the Dance closer performance

"doDanceTopless" → for the Dance topless performance

"doPoseNaked" → for the Pose naked performance

"doSoloFingering" → for the Solo fingering performance

"doToysMasturbation" → for the Toys masturbation performance

"doHandjob" → for the Handjob performance

"doFootjob" → for the Footjob performance

"doTitsjob" → for the Titsjob performance

"doBlowjob" → for the Blowjob performance

"doDeepthroat" → for the Deepthroat performance

"doFacefuck" → for the Facefuck performance

"doMissionary" → for the Missionary performance

"doDoggystyle" → for the Doggystyle performance

"doAnal" → for the Anal performance

"doThreesome" → for the Threesome performance

"doFoursome" → for the Foursome performance

"doOrgy" → for the Orgy performance

"doBodyCumshot" → for the Body cumshot finisher

"doTitsCumshot" → for the Tits cumshot finisher

"doFacial" → for the Facial finisher

"doSwallow" → for the Swallow finisher

"doCreampie" → for the Creampie finisher

"doAnalCreampie" → for the Anal creampie finisher

"texts" folder

This folder should contain only one file named "Bio.txt" (extension must be present). This file should contain the biography of the girl as displayed during recruitment. If the "Bio.txt" file or the "texts" folder is not present, the game will display an empty biography.

It should also contain a folder called "Lessons", in which are created the files describing this girl's possible lessons (see "Creating lessons" for more information) .

"videos" folder

This is where you will put the videos of the performances of the girl. It is a series of folders and sub-folders in which you must put the performance you want to see when it is happening in the game. For example, the videos you want to see for a basic dance act should be placed in "videos\Performances\Work\Dance\Dance\".

Each final sub-folder has a name that corresponds to a performance. The details of these sub-folders are given below. If one or more sub-folders do not contain any video or are not in the "videos\Performances\Work\" folder, the game will use default clips.

Important: the video files **must** be in MP4, WEBM, WMV or OGV format. The MP4 videos should use the h264 codec, the WEBM use the vp8 codec (vp9 will not work). Any other video format will be ignored.

I advise to use a (compressed) WEBM VP8 video format or (compressed) MP4 .h264. While both of these format work, WEBM VP8 is preferred since it is a more universal format, that will work with Linux system where MP4 will not.

The compression is not mandatory, it is just to try and keep the girlpack at a reasonable memory size.

It is strongly advised to use videos in a 16:9 ratio, to avoid distortions.

GIFs will NOT work.

Details of the sub-folders

(starting from "videos\Performances\Work\"):

"Dance\Dance\" → should contain the basic dance clips

"Dance\Closer\" → should contain the closer dance clips

"Dance\Topless\" → should contain the topless dance clips

"Finish\Body\Body\"→ should contain the finishing on body clips

"Finish\Body\Tits\"→ should contain the finishing on tits clips

"Finish\Facial\Face\"→ should contain the finishing on face clips

"Finish\Facial\Swallow\"→ should contain the swallowing clips

"Finish\Inside\AnalCreampie\"→ should contain the anal creampies clips

"Finish\Inside\Creampie\"→ should contain the creampies clips

"Foreplay\FJ\" → should contain the footjobs clips

"Foreplay\HJ\" → should contain the handjobs clips

"Foreplay\TJ\" → should contain the titsjobs clips

"Group\3Some\" → should contain the threesomes clips

"Group\4Some\" → should contain the foursomes clips

"Group\Orgy\" → should contain the orgies clips

"Oral\BJ\" → should contain the blowjobs clips

"Oral\Deepthroat\" → should contain the deepthroats clips

"Oral\Facefuck\" → should contain the facefucks clips

"Pose\HandMast\" → should contain the girl's masturbation with her hands clips

"Pose\Naked\" → should contain the girl's posing naked clips

"Pose\ToyMast\" → should contain the girl's masturbation with toys clips

"Sex\Anal\" → should contain the anal sex clips

"Sex\Doggy\" → should contain the doggystyle sex clips

"Sex\Missionary\" → should contain the missionary sex clips

Creating lessons

<u>Note:</u> if you want to create lessons, I strongly advise using a good text editor like Notepad++ (https://notepad-plus-plus.org/).

A lesson is a dialog between the girl and the main character. It starts with an introductory dialog between the two of them, then there is a question that the player must answer, then another dialog concluding the lesson. Answering the question can increase or decrease the girl's openness. It can also be a dialog without any question.

In order to create a lesson, you must create a JSON file (ending in .json) in the "/texts/Lessons/" folder. The file can have any name you want but the extension must be present. As always, it is strongly advised to use a pre-made lesson file and fill the fields as desired. The example girl pack contains several lessons that are examples of types of lessons. It is advised to pick one corresponding to what you want to make and fill it accordingly.

There are two more folders used for lessons: "/images/Lessons backgrounds/", "/images/Lessons portraits/".

The first folder is used for images displayed as backgrounds. Those images can have any name, but the extension must be present and must be either ".jpg" or ".png". Since they are displayed in full-screen, beware of images that contain transparency, as this will create a weird display. They are displayed in a 16:9 ratio, and the preferred resolution is 1920x1080, to avoid any distortion or loss of quality.

The second folder is used for images displayed as portraits of the girl. Those images can have any name, but the extension must be present and must be either ".jpg" or ".png". They are displayed in a 2:3 ratio, and should have a resolution of at least 600x900, in order to avoid any distortion or loss of quality.

If you need to test your lessons in-game, you can press Ctrl+D in the lounge screen to list every lessons for a girl and forcibly start one. Abusing it can break your savefile so I **strongly** advise NOT to save after using this.

You can also see examples of lessons in the ExampleGirl girlpack.

See below for a detailed list and explanation of the fields of the lesson file.

"beforeQuestionDialog"→ The dialog contained in the lesson, before the question.
It is described in the form of a list of five elements: the talker's name, the line itself, the background image ued for this line, the portrait used for this line and the portrait position for this line.
e.g.: [{"talker":"you", "line": "This is a line said by the main character","portrait":"Example portrait.jpg","background":"Example background.jpg","portraitPosition":"right"}].

"talker" is the talker's name. It can be either "you" or "mc" for the main character, "her" or "girl" for the girl, or "narrator" for the descriptive aspect of the dialog.

"line" is the content of the line itself.
If you want to start a new line, you must add \n inside the text at the place in which you want the new line to start.
If you want to add quotation marks (i.e. ") in the text, you must add a \ before (i.e. write \").
YOU MUST NOT USE THE NORMAL NEW LINE OR QUOTATION MARK BECAUSE IT WILL NOT WORK AND CAUSE AN ERROR.

"background" is the name of the image used as background for this line.
The game will go look for the image in the "/images/Lessons backgrounds/" folder.
If no background is specified, or the file cannot be found, the game will display a default background.
Expects a name of file, with the extension (either .jpg or .png).
Default value is "".

"portrait" is the name of the image used as a portrait for this line.
The game will go look for the image in the "/images/Lessons portraits/" folder.
If no portrait is specified, or the file cannot be found, the game won't display a portrait. You can specify displaying the default portrait by setting this field to the value "default".
Expects a name of file, with the extension (either .jpg or .png).
Default value is "".

"portraitPosition" is the position of the portrait for this line. It has three possible values : "left", "center" or "right".
Default value is "left".

There is no default value here, this field MUST be filled, at least with one line.

"answerOneDialog"→ The dialog displayed when the first answer to the question has been chosen. It also contains the answer itself.
It is described in the same way as the "beforeQuestionDialog".
<u>The first line of this dialog will be used as the first answer to the question of this lesson.</u>
If you don't want the lesson to have a question, **do not** specify this field.
If this field is not specified, the game will consider that the lesson is a simple dialog between the girl and the main character, without question. It will only use the "beforeQuestionDialog" lines.
Default value is an empty dialog ([]).

"answerTwoDialog"→ The dialog displayed when the second answer to the question has been chosen. It also contains the answer itself.
It is described in the same way as the "beforeQuestionDialog".
<u>The first line of this dialog will be used as the second answer to the question of this lesson.</u>
If you don't want the lesson to have a question, **do not** specify this field.
If this field is not specified, the game will consider that the lesson is a simple dialog between the girl and the main character, without question. It will only use the "beforeQuestionDialog" lines.
Default value is an empty dialog ([]).

"answerThreeDialog"→ The dialog displayed when the third answer to the question has been chosen. It also contains the answer itself.
It is described in the same way as the "beforeQuestionDialog".
<u>The first line of this dialog will be used as the third answer to the question of this lesson.</u>
If this field is not specified, the game will consider that the question only has two possible answers.
Default value is an empty dialog ([]).

"answerFourDialog"→ The dialog displayed when the fourth answer to the question has been chosen. It also contains the answer itself.
It is described in the same way as the "beforeQuestionDialog".
<u>The first line of this dialog will be used as the fourth answer to the question of this lesson.</u>
If this field is not specified, the game will consider that the question only has three possible answers.
Default value is an empty dialog ([]).

"minOpenness" → The minimum openness required by the girl to see this lesson.
Expects a decimal number between 0 and 100.
Default value is 0.

"maxOpenness"→ The maximum openness the girl can have and still see this lesson.

Expects a decimal number between 0 and 100.
Default value is 100.

"needLessonsSeen"→ The names of the lessons that must be seen and completed before this one can be seen.
Expects a list of the file names (with or without .json extension, not case sensitive) that describe the lessons that must be seen and completed before this one can be seen. The names must be between [] and separated with commas.
e.g.: "needLessonsSeen":["Lesson_1","Lesson_2"].
A wrong lesson name will not prevent the lesson from being reachable.
An empty list means that there are no lesson that must be seen and completed before this one.
Default value is an empty list ([]).

"rightAnswer"→ The number of the right answer to the question.
Expects an integer number, between 1 and 4 (both included).
The lesson can be reused until the right answer is picked.
When the right answer is picked, the lesson will be considered done, and won't be used again unless the "repeatable" field is at **true**.
Default value is 0, meaning there are no right answer.

"repeatable"→ If this lesson can be seen again, even after being seen and completed once.
Expects **true** or **false**.
Default value is **false**.

"opennessGains"→ The values of the openness gains (or losses if the value is negative) when answering the question.
Expects a list of decimal values between [], separated by commas.
e.g.: "opennessGains":[10,0.5,-4.2,0].
The first value is the openness gained from the first answer, the second value is for the second anwer, etc...
Default value is [0,0,0,0].

List of performances and their minimum openness required

Dance: 0 openness required.

Pose: 16.6 openness required

Foreplay: 33.2 openness required

Oral: 49.8 openness required

Sex: 66.4 openness required

Group: 83 openness required

Body cumshot: 33.2 openness required

Tits cumshot: 44.2 openness required

Facial: 49.8 openness required

Swallow: 60.8 openness required

Creampie: 71.9 openness required

Anal creampie: 83 openness required

Notes on removing a girlpack

A girlpack can be deleted from the "Girlpacks\" folder but the girl will be deleted from any save file she has been hired into when said save file is loaded.

How to add your music to the game

Simply put the musics you would like to hear in the game in the correct folder. The "Musics" folder is in the main game folder and contains two sub-folders.

The first one is called "Club" and should contain the musics that should be played during the "management" phase, meaning everything outside of the "work" phase.

The second one is called "Work" and should contain the musics that should be played during the "work" phase.

<u>Unfortunately, because of technical reasons, only musics in the OGGVORBIS (extension .ogg) format will work. Anything else will be ignored and won't be played in the game.</u>

If you put music in one of these folders, the default musics won't be played. The music that is played in-game is picked randomly from the ones in these folders.