Sylvain Seccia

seccia.dev

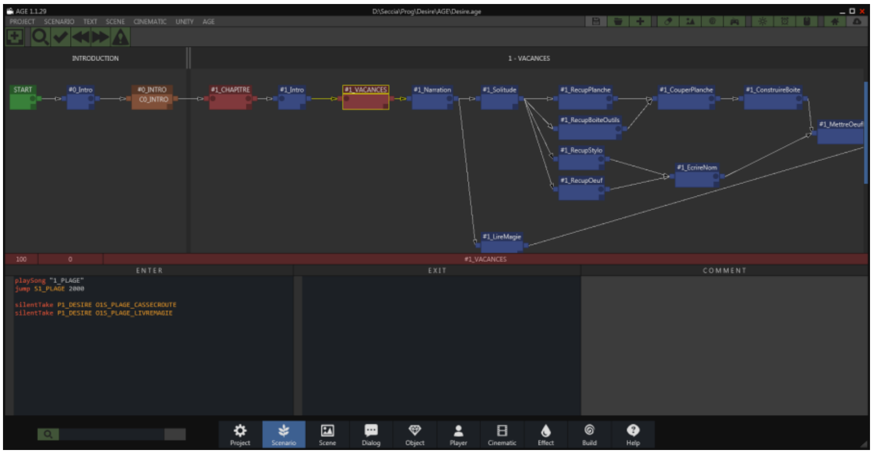
SECCIA EDITIONS

CONTENTS

INTRODUCTION

seccia.dev is a free software that allows you to create 2D adventure games without first having to have technical programming skills. A few notions of Scripting are enough to master all the features. The tools offered are very simple to use and are intended primarily for non-programmers, and more specifically for screenwriters, artists and game designers who want to create their own adventure games. Originally, I designed it as an internal tool to facilitate the development of my first adventure game Desired from 2012 to 2016. My goal was to adopt a screenplay approach to get as close as possible to the nature of the game I wanted to achieve.

Although seccia.dev is a proprietary IDE developed in C++ for Windows only, the generated games can run on many operating systems since the runtime has been developed with Unity. This means that you must first generate the building of the Unity project present in the software installation folder. In order to save valuable time on your projects, seccia.dev integrates a tool to start the compilation of the Unity project, as well as the automatic integration of the generated build. This constraint is due to the CGUs that require the developer to use his own Unity license. That said, at the installation, only the Windows version is pre-compiled and should only be used as a trial in the development phase of your projects.



```
playSong "1_PLAGE"
jump S1_PLAGE 2000

silentTake P1_DESIRE 015_PLAGE_CASSECROUTE
silentTake P1_DESIRE 015_PLAGE_LIVREMAGIE
```

Installation of the software

Before you can take advantage of the features of the software, it is necessary to follow a few instructions to ensure that all components are properly installed.

The software requires:

Windows 11 to current preferably Full HD resolution or more And ideally an SSD

And the installation of the Seccia Launcher app available at the following address:

- **https://www.seccia.com/download/launcher**

The updates will be available free of charge. You will be informed by a notification that will appear within the Seccia Launcher app interface whenever a new version is published.

This book refers to version 2.0.7.

Licence

I would like to point out that the software is completely free. No functional limitations are applied voluntarily for the purpose of restraining the product. On the other hand, the services offered can be paid for. You can develop and finalize your games for free for personal or commercial use. For more information, I invite you to visit the official website.

Compilation of the Unity project

The first step is to download and install, in the default folder, the current version of the Unity Hub program at the following address:

- **https://www.unity.com**

From the list of modules proposed during installation, select Android (not to mention submodules containing SDK), iOS, Web and Windows.

Once the installation is successful, the second step is to launch seccia.dev and open the UNITY/Unity project menu to select the desired platforms and launch the compilation by clicking the Build selected platforms button. Then wait...

Note that seccia.dev will always point to the most recent version of Unity installed on your machine, unless you set a specific path in the software preferences.

If you have to make changes to the Unity project to bring new features to your game, which is quite allowed by the terms of use, I strongly advise you not to change the properties of the Unity project at the risk of causing compilation errors.

Preferences

The preferences are available from the IDE/Preferences menu. If you encounter stability problems, you can disable multithreading support.

Documentation

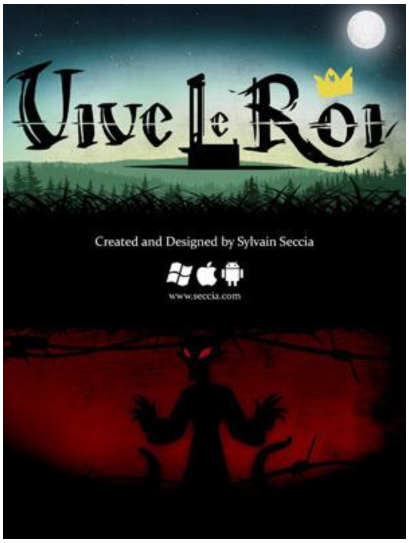
All functions of the language are documented and accessible from the software help page.

The F1 key is a shortcut and can even interact with scripts. Simply place the mouse cursor on the name of a function or keyword and press the F1 key to display the help page in question.

On this page you will also find the social networks, the websites of the seccia.dev community and the tutorials that will help you progress in your learning and I hope, will encourage you to change the community.

Don't hesitate to check the page regularly to become familiar with the language and be aware of the new content.

My games created with seccia.dev



28
03
5 7

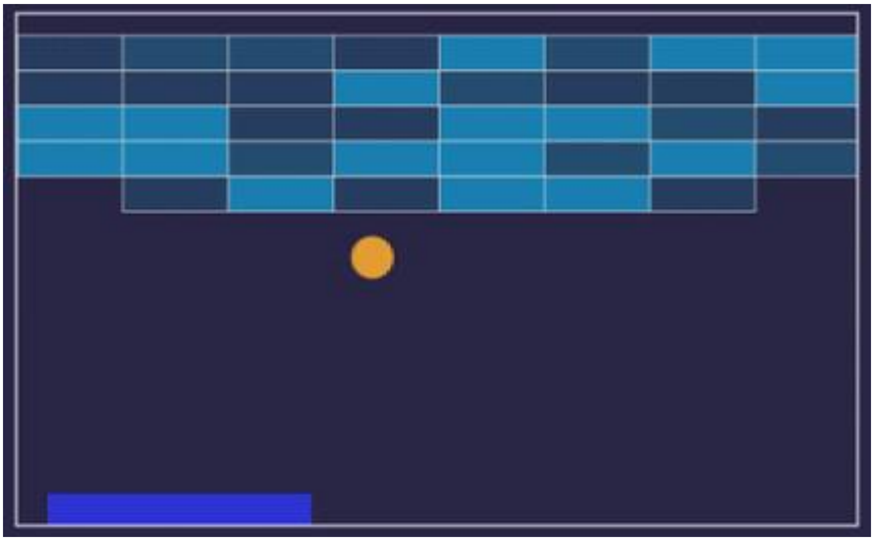
29
00
9 8

30
00
10 9

GAMEPLAY

According to Marc Goetzmann and Thibaud Zuppinger (2016): "the gameplay is the articulation between the game, the structures and rules of the game, and the play, the way the player appropriates the possibilities of the game by developing his own strategies, to meet the constraints that the rules of the game impose on him"

Before you rush into the realization of your first seccia.dev project, it is important to clearly discern the characteristics of a narrative adventure video game to better understand the interest of the gameplay and how its role will strongly influence the player's experience.



What is a video game?

A video game must above all be built around a gameplay. It is the main component that characterizes a video game and that should take your attention from the beginning of the project. Moreover, the mechanics and playability of a game are built by iteration by testing and retesting loops until you find the right balance. Thus, a video game does not need to be artistically cared for to be fun. This phase is often underestimated and disappointments unfortunately arise in progress or at the end of development.

In addition, video game is an artistic work because it uses several arts and disciplines to exist. It is in this case drawing, architecture, photography, writing, staging, music, noise, special effects and many other qualities. Therefore, these elements are important but are in fact only a dress to enrich your game, make it unique and manage to transmit emotions to the player while giving meaning to your work.

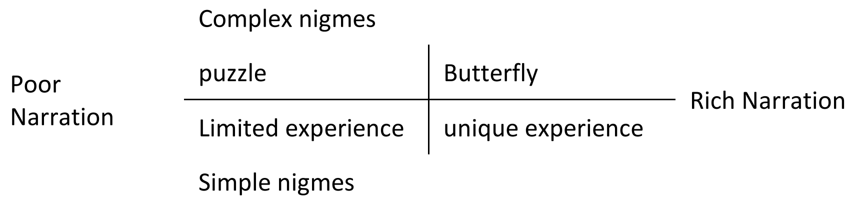
What is a narrative adventure game?

The narrative adventure game holds a special place in the world of video game. Would the narrative of an adventure game be more important than the gameplay? It is often the feeling that one might feel when playing a narrative game. Besides, some players even think that the adventure game essentially based on a narrative scheme would be closer to an interactive movie because of a gameplay too in retreat. I think they are wrong.

I'm talking about a genre apart because the storytelling of an adventure game is part of and cannot be separated from the gameplay. If the story frame is uninteresting or awkwardly written, the mechanics won't be enough to make the gameplay fun enough for the player. So the feeling we feel is right, because the gameplay of a narrative adventure game is based on both staging and puzzles.

The four gameplay profiles

If the gameplay of an adventure game has two components that we come to evoke, i.e. narration and puzzles, we can then propose an illustrated model of four profiles through the following diagram:



Depending on the level of complexity of the gameplay in terms of storytelling and puzzles, you will get one of these four profiles:

Limited experience	The storytelling is poor and puzzles are too simple to stimulate the player. This configuration will likely lead to a fad game without real challenge.
Headphones	Your game is unnarrative and contains a lot of complex puzzles. It would rather be a puzzle and not an adventure game. For an Escape Game, you could be in the right category. That said, more orientated phases within your adventure would contrast the player's experience.
Blowjob	This is probably the most difficult profile to master because the narrative is rich and complex puzzles. I advise against it unless the puzzles are justified and actually serve the staging. As with the previous profile, you can use it at key moments in your story. Find the right balance.
One-time experience	A rich narrative with simple puzzles is undoubtedly the ideal profile for a successful adventure game. Writing simple puzzles does not mean that you have to fall into ease and rely on banalities. Nor does it mean that puzzles must be easy to solve. In fact, the tasks to be performed must be short, simple and clear for the player, and it is the combination of these small puzzles that will prompt him to dig up the meninges to find the solution without cheating. Frustration occurs when it feels like he is no longer moving forward over a long period of time.

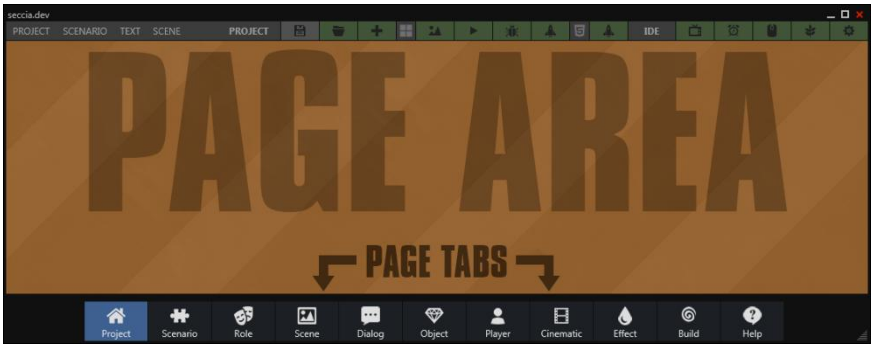
INTERFACE

seccia.dev offers an interface approach based on a concept of pages, a method uncommon in applications development software. On the other hand, the model based on asset windows is widely used in IDEs, as it is generally practical and efficient in many situations. But is it suitable for all development environments?

For example, in a programming tool, it is inconceivable not to be able to open multiple source files simultaneously, while maintaining the history of the changes. Conversely, in a video editing software, working simultaneously on multiple timelines is often useless or even counterproductive.

Although seccia.dev is more like programming software than a video editing tool, I chose to inspire the latter to design the ergonomics of its interface. Why? Because this approach reflects, in my opinion, more faithfully the stages of production of a seccia.dev project. In the end, it simplifies and undoubtedly accelerates the creation of your game.

The interface is divided into two parts: the common part and the part reserved for pages. The common part consists of fixed and ubiquitous elements: the main menu (top left), the toolbar (top right), and the navigation of the pages (bottom of the window). These shared areas remain accessible at any time and offer common features for several publishers. The page part, however, is dynamic and dedicated to the different stages of the project. It includes tools to manage the script, generate executables, consult educational resources, and access to publishers. Among them, the stage editor and the role editor will probably be the ones you will use most frequently.



Publishers are dedicated to the production of assemblies, i.e. the creation and modification of essential elements of the game. This includes: objects (including characters), scenes (levels), roles (playing logic), dialogues (conversations between characters), players (including inventories), cinematics (video in MP4 format) and effects (shaders).

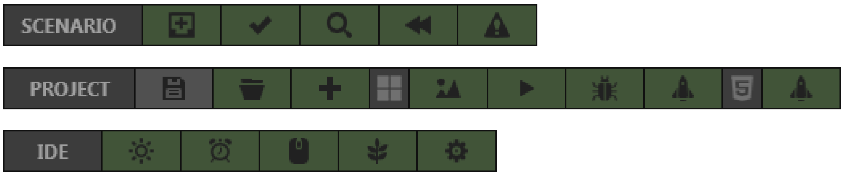
Main menu

The main menu, located at the top left of the window, allows access to the more advanced features of the software or less frequently requested, so they are removed and grouped by production step.

PROJECT	Creation and management of the project.
SCENARIO	Graph nodal, puzzles, gameplay...
TEXT	Dialogues, fonts, color palette, location...
SCENE	Batch processing, export of previews...

Toolbar

The toolbar, located at the top right of the window, provides quick access to the common features of the software and editors. It is presented in the form of three groups:



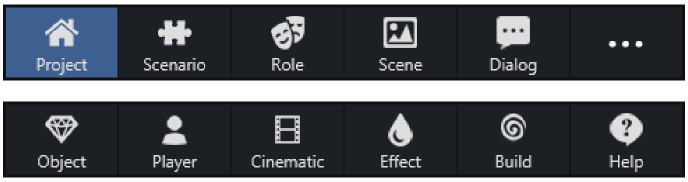
With the exception of the first group, these icons remain accessible regardless of the selected page, and for some they are assigned a keyboard shortcut.

⊞	Create a new box (shortened with P, S, D keys)
✓	Consider a box as resolved
🔍	Preview the entire scenario to move around
⏪	Select the START box of the scenario
⚠	Select the box containing an error
💾	Save current project

📂	Show project folder in Windows Explorer
✚	Create a new asset (shortened with the ALT key)
🖼	Launch the open scene or a specific scene in Live mode
▶	Launch the game in Live mode
🐞	Launch the game in Debug mode
🚀	Launch the game in Release mode for play_windows or play_web
☼	Enable or disable the rendering of the game in editors
⏰	Enable or disable real-time rendering in editors
🖱	Enable or disable Assisted Scripting mode
⚘	Access to graphic assets
⚙	Access software settings

Navigation bar

The navigation bar, located at the bottom of the window, allows you to access the pages and search the project.



The pages are also navigable using a keyboard shortcut.

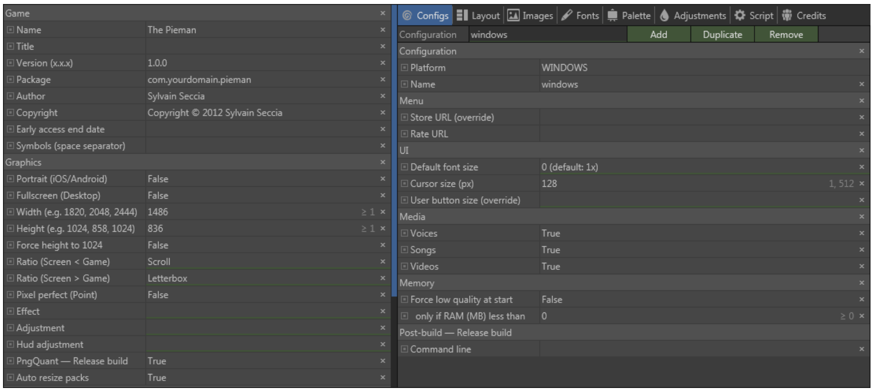
CTRL+Tab	To access the previously opened page.
ALT+Left	To access the previous page of the list.
ALT+Right	To access the next page of the list.

PROJECT

A seccia.dev project consists of a JSON file centralizing the data of the game, and externally a tree of folders and files. Files are mainly text and multimedia data. The main binary files produced by the software are in some way the compilation data of the game.

The main advantage of this approach is to bring greater flexibility to the project via the editing and versioning tools you will eventually use. However, it is important to avoid manual editing of the file bearing the seccia extension in order to preserve the links of unique identifiers. In other words, nothing prevents you from changing it as you please, provided you master what you do.

seccia.dev uses the PNG format to store all images in the game in external files. However, the JPEG format is accepted to import images that will be automatically converted.



Game	
Name	The Pieman
Title	
Version (x.x.x)	1.0.0
Package	com.yourdomain.pieman
Author	Sylvain Seccia
Copyright	Copyright © 2012 Sylvain Seccia
Early access end date	
Symbols (space separator)	
Graphics	
Portrait (iOS/Android)	False
Fullscreen (Desktop)	False
Width (e.g. 1820, 2048, 2444)	1486 ≥ 1
Height (e.g. 1024, 858, 1024)	836 ≥ 1
Force height to 1024	False
Ratio (Screen < Game)	Scroll
Ratio (Screen > Game)	Letterbox
Pixel perfect (Point)	False
Effect	
Adjustment	
Hud adjustment	
PngQuant — Release build	True
Auto resize packs	True

Configuration	windows
Configuration	
Platform	WINDOWS
Name	windows
Menu	
Store URL (override)	
Rate URL	
UI	
Default font size	0 (default: 1x)
Cursor size (px)	128 1, 512
User button size (override)	
Media	
Voices	True
Songs	True
Videos	True
Memory	
Force low quality at start	False
only if RAM (MB) less than	0 ≥ 0
Post-build — Release build	
Command line	

Structure

The project contains a file defining the project's identity and a list of files:

ASSETS	An asset subfolder containing all the dependent files: texts, images, sounds...
BINARIES	The location of the build generated and stored by configuration.

IMAGE	Images of the game interface. This folder is also accessible from the Project page.
LIBRARY	The folder of interfaces to export libraries and to contain a subfolder per interface.
PLATFORM	Platform files (mainly icons).
SCENARIO	The export scenario in PNG acting as GDD and chapter-related backups.
SNAPSHOT	Scene previews.
SONG	Music in OGG format for Windows and MP3 format for other platforms.
SOUND	The sounds in WAV format.
TEMP	Location of temporary files. In case of deletion, seccia.dev will have to regenerate these files.
TEXT	Texts to be translated in CSV and HTML format, customizable font and other texts.

Each asset has a unique identifier assigned to its creation and the folders return this identifier. Again, it is strongly advised not to rename the folders by the Windows Explorer. When the name of an asset is changed within the software, all the dependent files as well as the links are automatically updated. It goes the same way for deletion.

There are seven types of assembly: Role, Scene, Object, Dialog, Player, Cinematic and Effect. The first letter of each type is included in the names of the Asset. That is, an Asset of type Scene necessarily starts with the letter S, for example S_HOME and by deduction: R for Role, O for Object, D for Dialog, C for Cinematic, P for Player and E for Effect.

Regarding images, you can open the Project page at any time to view the complete list of images from the game interface. All images are grouped in the IMAGE folder.

Parameters

All project parameters are grouped and centralized on the Project page.

Game

Name	The name of the game, also used to name the files.
Title	The title of the game. If the field is empty, the title will be the name of the game. To write in UTF-8, prefix it by @.
Version	Version of the game in the form X.X.X (1.0.0).
Package	Name of the package of your game (com.domain.name)
Author	Name of developer.
Copyright	Information on rights reserved (Copyright © Year Name)
Early access end date	It is possible to unlock actions by code after a predefined date via the early_access function.
Symbols	Allows you to set a list of symbols (words separated by spaces) to the generation of the game and to test the presence of symbols by code via the symbol function.
Graphics	
Portrait	Only available for mobiles, this option allows to force portrait orientation to start.
Fullscreen	Exclusively for Desktop platforms, this option allows you to force full screen mode to start the game. In this case, the player will no longer be able to change the resolution in the options.
With & Height	The size of the game in pixels. The maximum size is 4096. Scenes may have a different size. If your game is in 16/9 format, the ideal resolution is 1820x1024 (2 textures). If your game is in 21/9 format, the ideal resolution is 2048x858 (2 textures). In the latter case, it might be better to use 2444x1024 (3 textures) resolution if you intend to zoom frequently in order to keep optimal quality while optimizing memory since the sets use textures of 1024x1024. If the ratio is not a priority criterion and you want to optimize the entire textures surface, you can choose the resolution 2048x1024 (2 textures) corresponding to 18/9 or 2/1.
Ratio (Screen < Game)	When the resolution ratio of the screen is smaller than the resolution ratio of your game, several choices are available to you to compensate for the space lost

	vertically: Scroll: The height of the scene is equal to the height of the screen and a horizontal scrolling is enabled Letterbox: Black bands are added at the top and/or bottom Crop: The scene is trimmed at the top and/or bottom
Ratio (Screen > Game)	When the resolution ratio of the screen is greater than the resolution ratio of your game, several choices are available to you to compensate for the space lost horizontally: Letterbox: Black bands are added to the left and/or right Crop: The scene is trimmed to the left and/or right
Pixel perfect	If enabled, Unity will use FilterMode.Point.
PngQuant	This option is only valid for the build release and allows to encode transparent PNGs in 8 bits instead of 32 bits thanks to the external PngQuant library.
Auto resize packs	If the option is validated, seccia.dev runs through all the scenes to know the largest size of the object. If it is less than 100% of the object, the images are resized to optimize the number of textures to generate. In case of change, it is necessary to regenerate the textures. If you intend to zoom frequently in your scenes and thus keep an optimal quality of your textures, it is better to disable the option and ensure the correct size of the PNG files.
Filter	
Effect	An effect can be applied to the entire scene. This option is ignored if it is overdefined in the stage editor.
Adjustment	The adjustment is applied at the time of generation of the game. This option does not take into account HUDs.
Hud adjustment	This option only changes images related to HUDs.
Light	
Baked	This option makes it possible to optimize the rendering by performing a pretreatment in case of a change of lights. This pre-rendering is then stored in another texture. The advantage of this method is to increase the framerate of your game, the disadvantage is to use more graphic resources. If the

	scene requires constantly changing the state of the lights, it is not necessary to activate the option. I advise you to use it for static lights or that change occasionally.
Ambient	This value sets the background light of the scene between 0 and 255. If the value is 0, the screen of the game will be completely black. If the value is 255, the pixels will have the original color. It is not possible to further illuminate the scene by this property.
Blur scale	This option allows to apply a blur effect on the light to soften the radius. The higher the value and the more the performance will be impacted.
Low quality	For performance gains, enable this option to reduce the size of the textures reserved for rendering lights. The result will be less detailed and more pixelized.
Bokeh	
Shape width/height	This option allows you to define the maximum size of the shape of a bokeh in order to reproduce the effect of an optical. A spherical optic produces round shapes while anamorphic lenses used in cinema produce oval shapes. The default values are 128 for width and 256 for height.
Menu	
Custom menu	This option allows you to customize the interface by indicating the scene that will become the menu home screen. The jump_menu function will allow you to navigate from one screen to another within the menu.
Effect	An effect can be applied to the native menu. This option does not apply to the custom menu.
Splash durations	Wait time in seconds. If the value is 0, the player must click or touch the screen to continue.
Menu Options	View the Options menu.
Quality menu	View menu to change texture quality.
Merge Audio/Text menu	Allows you to differentiate between the language of the subtitles and the audio language.
Font size menu	View menu to change font size.

Subtitle menu	View menu Subtitles to change the mode and speed of scrolling of the text.
Privacy policy URL	Link that returns to your web page containing your privacy policy. This link is required if you are using the online backups.
URL Store	Link that returns to the page of your games.
User button position	Position of customizable buttons.
User button size	Size of the customizable buttons of the menu. On scale 1, the size is 96 pixels.
User button URL	Link that returns to a web page.
User button show	By default all buttons are visible. To only display the first three buttons, just write "123".
Text	
Font	The name of the font proposed by seccia.dev. It is possible to provide a custom font.
CJK do	The name of the Asian font used for texture generation. The font must be installed on the computer at the time of generation. If the field is empty or the font could not be loaded, the software will attempt to use a default font.
Font color	The default text color.
Font color (Highlight)	The color of the text when the player points to a replica choice during a conversation.
Font color (Door)	The color of the text when the player points on a door.
Font color (Menu)	The text color of the menus.
Font color (Menu/Rate)	The text color of the EVALUATE the game menu.
Font color (Menu/Highlight)	The color of the text when the player points to a menu.
Font color (Menu/Values)	The color of the text indicating the values of the options.

Font color (Credits)	The color of the text in the generic.
Font color (Credits/Title)	The color of the generic text when the line starts with an arobase.
Back color	Displaying a rectangle behind the text for easy reading. The color is in hexadecimal RGBA format.
Default subtitles speed	Sets the default mode and speed.
Subtitles margin	Sets the margin between text and the bottom of the screen.
Typewriter	If this option is enabled, the dialogs will write on the letter-by-letter screen according to the speed of the game options.
Vibration	Allows to animate the text with a undulation effect.
Randomize dialogs	Allows you to mix the order of the choices of the replicas of all conversations at the start of a new game.
Balloon: image	The inside margins in pixels separated by a comma to define the variable area of the image when adjusting the bubble to the text in the following order: left, top, right and bottom. The values must be positive. It is necessary to include an ITEM_BALLOON image of 128x128 pixels to use COMICS mode.
Balloon: text	The inside margins in pixels separated by a comma to define the text box in the following order: left, top, right and bottom. The values must be positive.
Choice: size	Size in pixels of the bubble where the text of the choice proposed to the player will be displayed. It is necessary to include an ITEM_CHOICE image of 256x256 pixels to use the NOVEL VISUAL mode.
Choice: icon size	Size in pixels of icon.
Choice: horz space	Pixel space between two bubbles.
Choice: green space	Pixel space between the bubbles and the edge of the screen.

Choice: margin	The inside margins in pixels separated by a comma to define the text box in the following order: left, top, right and bottom. The values must be positive.
Audio	
Level menu	Allows you to merge the volume of audio sources into a single menu or disable the menu.
Preload sounds	Allows you to load in memory all the sounds (WAV files) from your game to the launch of the application.
Savegame	
Menu	Allows to enable or disable manual backup management.
Server URL	The folder link containing seccia.dev scripts to manage online backups via your own site. If the field is empty, the online backup menu will be inaccessible.
Server game	Name of the game and therefore the subfolder to be created on your server.
Version	The version of the game backup. If two versions of the game are too different, it is better to create this value. The old backups will then no longer be compatible but this will avoid crashing the application or getting risky results. If the value is 0, the backups will never be compatible between two versions.
Autosave	Allows you to run an automatic backup every time a puzzle box is solved.
Languages	
Native	The default language used in the editor. It is usually the native language of the developer or game.
Default	The default language at the first launch of the game.
OS if possible	If the option is enabled, the application will try to recover the language of the device. In case of failure, the default language will be taken into account.
Language	Before you can use a language, it is important to enable to access the fields. When you disable a language, the texts are not deleted. Use the

	TEXT/Purge unused languages menu to perform a complete (irreversible) cleanup.
Post-build command lines	
Zip	It is possible to compress all the buildings at the end of the generation by activating this option.

Configuration Settings

You can create build configurations with different settings depending on the platform and version to be distributed. By default, there are two configurations required for testing within the software: play_windows and play_web. Platforms have their own settings.

Parameters common to all platforms	
URL Store	Link that returns to the page of your games. If the field is used, the equivalent link of the Menu group is ignored.
Rate URL	Link that returns to the page of the store allowing to evaluate the game.
Default font size	The font size at the first launch of the application. If the value is 0, a default value is used.
Cursor size	The size of the cursor in pixels.
User button size	The size of the customizable menu buttons. If the field is used, the equivalent menu group link is ignored.
Media	Allows to include voices, music and videos.
Low quality force	Enables low-quality mode when the application is first launched if the Quality menu is accessible. By default, the application is based on the device's performance to activate the mode.
Command line	It is possible to run a command line program at the end of the generation. This feature only applies to build release.
Android-specific settings	

Package	Name of the configuration-specific package. If the field is empty, the Game group package is used.
Code version	The version of the APK or AAB file as an integer starting with 1.
Bundle	By default, an APK file is generated. To generate an AAB file to publish on the Google Play store, it is necessary to activate the option.
File	The Keystore file generated by the Android SDK to sign the APK or AAB file.
Store password	The password that was used to generate the file.
Alias	The name that was used to generate the file.
Password	The password that was used to generate the file. If both passwords are identical, you must complete both fields.
Web-specific settings	
Distribution	Allows you to specify the type of distribution. Choose Local to test your game within the software.
URL	Content folder link if the assets are accessible from another location.
Check domain	Allows to check the validity of the domain name of the site where the application is hosted. If the option is enabled, it is necessary to provide a list of authorized domains.
Allowed domains	The list of accepted domain names. The application will not launch if the site hosting the game is not allowed.
With & Height	The resolution of the game inside the web page.

Appropriations

Credits scroll at the end of the game and are also visible from the main menu or when calling the show_credits function. Each language has its own text.

The text color is defined by the Font color (Credits) property. To change the color of an entire line, just add the @ character at the beginning of the line. This color is customizable via the Font color (Credits/Title) property.

You can also include animated objects on a line by writing the name of the object preceded by the arobase character. Only STOP animation will be played.

You can add predefined keywords to recover certain properties of your project. To do this seccia.dev offers a list of constants accessible from the dialogs, text fields and variables of your scripts.

{OS}	The current build platform: windows, web, android, ios
{LANG}	The language selected by the player: English, French, German, Spanish, Italian, SimplifiedChinese...
{NAME}	The name of the game without space or special characters
{TITLE}	The title of the game
{VERSION}	The game version
{AUTHOR}	The developer's name
{COPYRIGHT}	The Copyright field indicated in the properties

Terminal

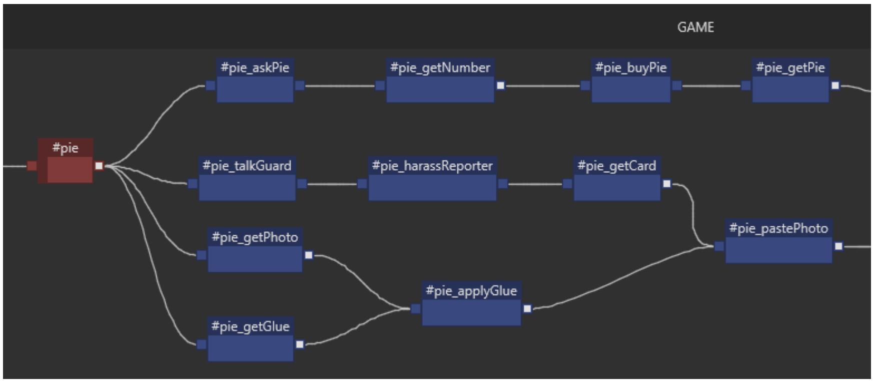
The terminal, accessible from the main FILE menu, allows you to apply a set of changes to your project via a command line interface. You can easily perform batch processing by inserting a list of commands in the text field. Any changes will be final. Don't hesitate to duplicate your project before accessing the terminal if you are groping.

```
object_copyFrame PNG ASSET ANIM DIR index object_setTitle
ASSET "title" LANGUAGE {...} object_update ASSET {...}
project_deleteAsset ASSET {...} project_newAsset ASSET
{...} project_renameAsset OLDASSET NEWASSET {...}
```

SCENARIO

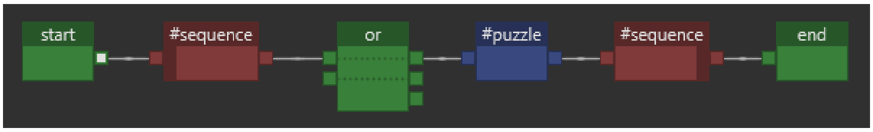
The scenario editor is at the heart of the environment, allowing to define the narrative unfolding of the game. It is therefore not necessary to program a state-of-the-art machine to control the puzzles since seccia.dev already offers a ready-to-use tool. Just add nodal boxes to the script and connect them to each other. At the runtime, an interpreter is in charge of updating the states when the game creator decides to validate a puzzle following a player's action. In other words, the creator only needs to notify the validation of a box when the player solves the puzzle in question. That said, the scenario editor is only an abstract representation of your puzzles which allows above all to structure the narrative and facilitate the upstream work.

The scripts integrate into the nodal boxes of the script, roles and timelines. You will find an exhaustive list of the language functions by clicking on the button on the help page at the bottom of the window.



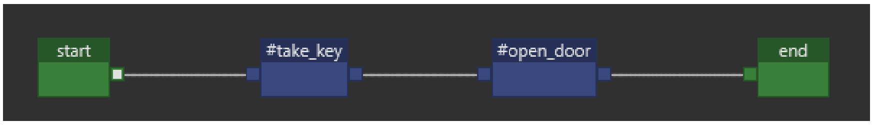
Nodal box

There are only five types of nodal boxes: puzzle, sequence, gold, start and end. Each box has specific properties to remember. To add a box to the scenario, simply click on the [+] button of the main toolbar to make appear the list of types, or use the keyboard keys: P, S, O or E. You will certainly notice, the key represents the first letter of the type. Only the start box is unique. The names you will give to the puzzle boxes and sequence will always have to start with the # character.

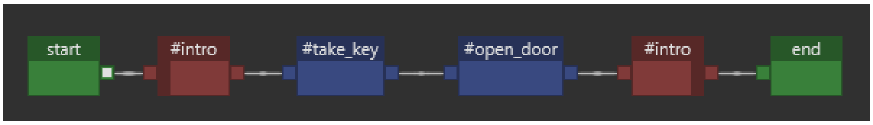


The puzzle and sequence boxes have two connectors: one in and one in and one out. The start box has only one output and the end box only one input. On the other hand, the gold box is slightly more complex with its two inputs and three outputs but don't worry, we will come back to it. An input connector can receive several connections and an output connector can also transmit several signals. In addition, the boxes can have an incoming script and a script coming out like a classic state machine with the exception of the script executed in a loop that doesn't exist. Before studying the connections, it is important to describe the particularities of the five types mentioned and to understand that the progression can only take place from left to right, in other words from the past to the future without the possibility of going back.

Start and end boxes are mandatory. They cannot be removed (there must remain at least one end box). When a new part is launched from the game menu, the first box executed is start as its name indicates. By definition, there can be no boxes located before start and after end. All the boxes end will lead to the credits and will definitely finish the game. Distinguish the persistence of data from the games and the game. Values can coexist between all the games, giving you the possibility to extend the replayability of your game by encouraging the player to achieve, for example, all possible ends.



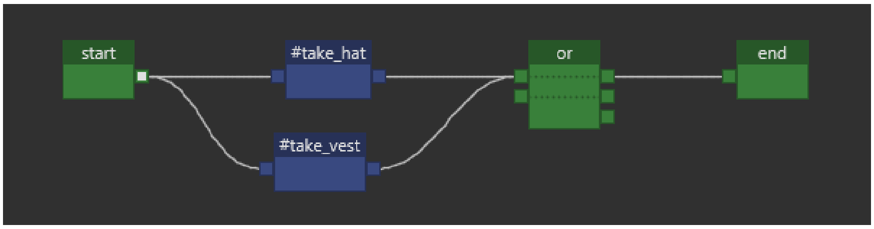
The puzzle box is the most common because it is used to define the player's actions. You have to imagine the blue box as an infinite mini loop where the only way out is to validate the puzzle. There are several ways to do it: by code via the success function, by role editor via the success action or by the properties of some editors.



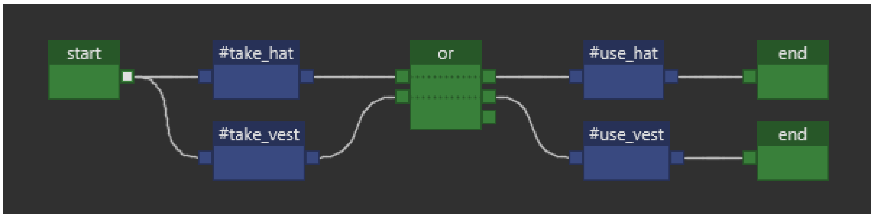
Unlike puzzles, the sequences are not blocking and the outgoing script is executed immediately without the player's interaction. The interest of the sequence box is to improve the structure of the puzzles to gain flexibility and readability, and thus to better sequence the script. This is part of the best practice. Even more interesting, a sequence automatically defines a puzzle range and then just call the started, unstarted, ended and playing functions to find out if the player is inside or if the entire range has already been solved.

```
if playing #intro // code here end
```

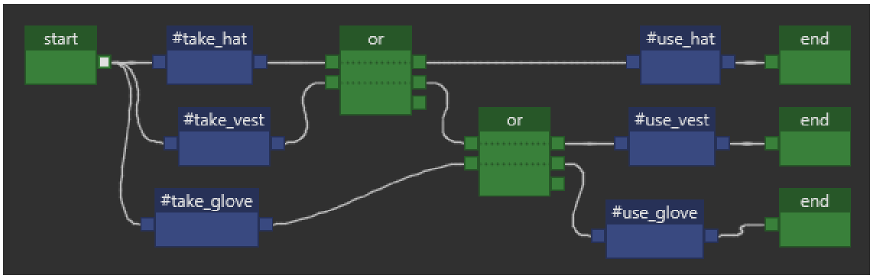
It is of course possible to interlock beaches provided that the concept of PEPS (first come out first) is respected. This particularity is used above all to force the evolution of narration. A priori the player should not be able to solve puzzles located outside of a beach as long as he is not out. If this constraint does not satisfy you, it means that the concept of the beaches is not finally adapted to your needs. To illustrate this case, let's take a very simple concrete example: the player finds himself in a watch-pens that would temporarily limit him in his travels until he manages to escape.



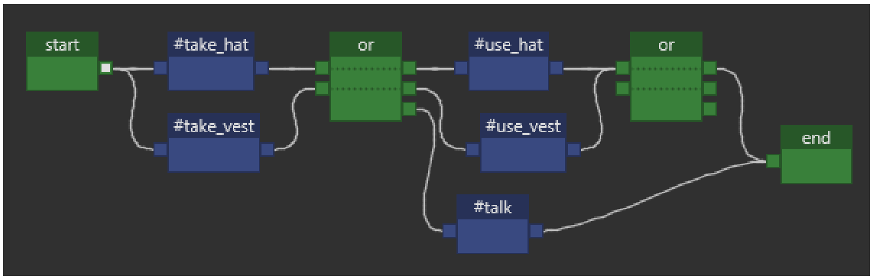
Finally, among the five types, the gold box allows the player to solve a puzzle in a predefined list by eventually producing an impact on the story. In this configuration, the player is no longer supposed to solve all the puzzles offered to him to move on to the next step. Thus, the first riddle that will lead to the gold box will become the selected riddle and will invalidate the others. From the editor, you can invalidate puzzles by marking them as lost in order to simulate a game without having to replay everything from the beginning. To use the box in this way, simply connect your puzzles to the first entry and get the signal out via the first output as shown in the diagram above. Keep in mind that the condition always performs at the entrance and never at the exit.



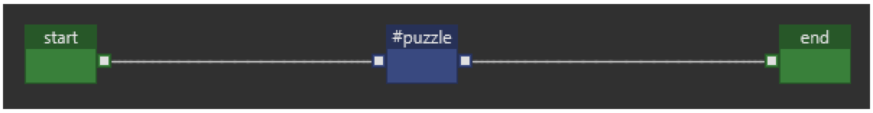
The gold box also makes it possible to change the narrative according to the player's choices thanks to the branch concept. To use it in this way, you have to use the first two circuits. The first circuit connects the first entry with the first exit, and the second circuit connects the second entry with the second exit. These are separate circuits: if the signal enters through a circuit, it will necessarily exit through the same circuit. The first signal entered will therefore define the output circuit. It is very convenient for two branches but what is it with three branches since the box offers only two circuits? Take a look at the picture below.



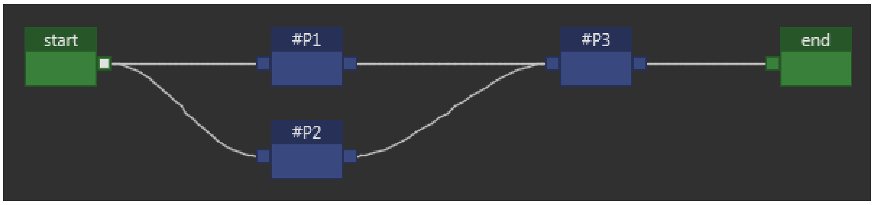
Thanks to this trick, you can define as many branches as you want. It's about tying up the gold boxes to add additional branches.



That's not all, you've probably noticed the presence of a third exit that actually represents a third circuit without input. Whatever the input chosen, the signal will come out through this third circuit. It's not a catch-all but a mandatory output if a cable is connected.



To summarize the operation of the nodal boxes, according to the diagram above at the start of a new game, it is first the starting script that is executed, then in the same frame of the application, it is the turn of the incoming puzzle script to be executed. As the peculiarity of the puzzle boxes is to block the progression of the scenario, the interpreter waits for the notification of the creator to come out of it as we have already mentioned. It is therefore only at the time of the validation of the puzzle by the triggering of an action caused by the player, that the outgoing puzzle script is executed, followed by the incoming script end.



Apart from that, when a box receives two incoming connections, the behavior is slightly different. The incoming script of #P3 is executed only after the validation of the boxes #P1 and #P2, i.e. after the outgoing scripts are executed. To better understand it, imagine that the outgoing signal of #P1 is on hold before the entry of #P3 until all the other signals have arrived.

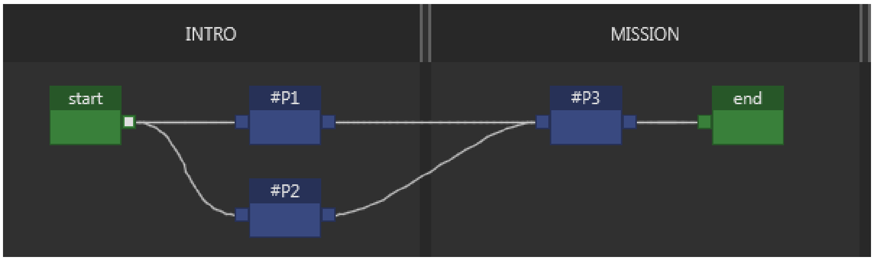
Since the scenario represents a global vision of your project, these scripts are supposed to change exclusively the general state of the game. It is not advisable to code actions that would not advance the narrative. This approach will allow you to simulate a game by checking the solved boxes in order to test your game more easily at a specific moment in history.

To do this, first create a #take_key puzzle and add the following instruction in the outgoing script:

```
take P_HERO O_KEY
```

Just validate the puzzle elsewhere in your application, either by code or by editors, to place the object in the inventory. Thus, in case of game simulation, the object will already be present in the inventory because the scripts of the scenario will be executed at the launch without interaction of the player.

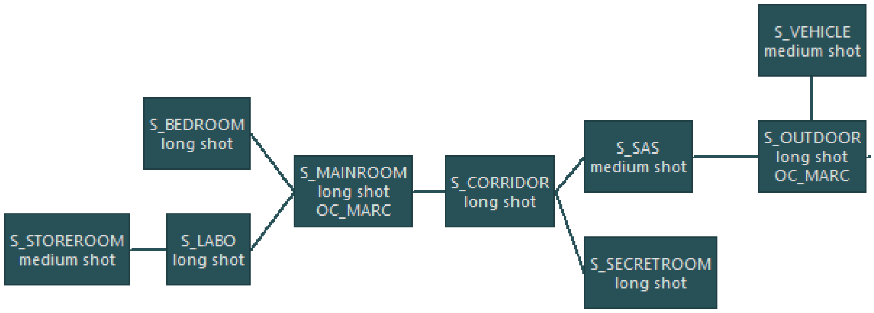
Section



The sections are optional and allow to better structure the scenario chronologically. They are represented as columns that can contain different types of data. To add a new section, just right click on the background or section title and select Add Section. Double click on the title allows to edit the section.

Keep the Shift key down to avoid automatically moving the boxes when you resize a section.

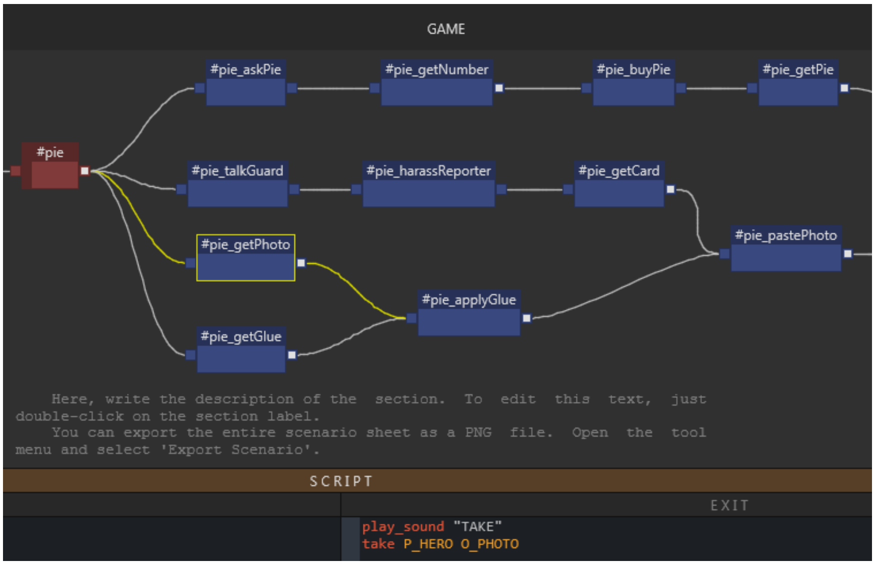
A descriptive text can be displayed at the bottom of the section. This text may contain several paragraphs and is not limited in number of characters. The dasset names are recognized in the description and automatically updated in case of a name change.



To document the location plan to be explored by the player, a mini diagram tool allows you to easily add boxes and connect them to each other to designate the accesses to the scenes. Click on the corner at the bottom right of the box to add a connection. This tool can of course be used to present other forms of data if needed. Keep in mind that it is only a way to expose and share notes, under no circumstances will these relationships be applied by the software.

Script

The nodal boxes in the script have at least one incoming script or an outgoing script with the exception of the puzzles that have both. You can edit these scripts by selecting a box to make appear the code editor just below the graph.



```
    Here, write the description of the  section.  To  edit  this  text,  just
double-click on the section label.
    You can export the entire scenario sheet as a PNG  file.  Open  the  tool
menu and select 'Export Scenario'.
```

```
play_sound "TAKE"
take P_HERO O_PHOTO
```

You don't have to worry about calling scripts, the interpreter is responsible for you. You only need to validate the puzzle boxes with for example the successful function by writing this instruction:

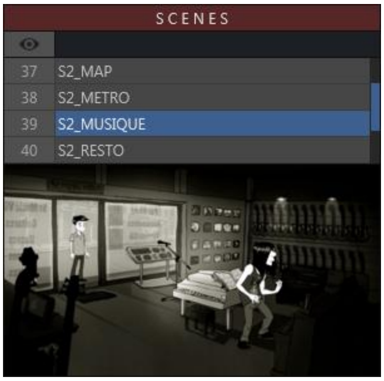
```
success #1_intro_takeKey
```

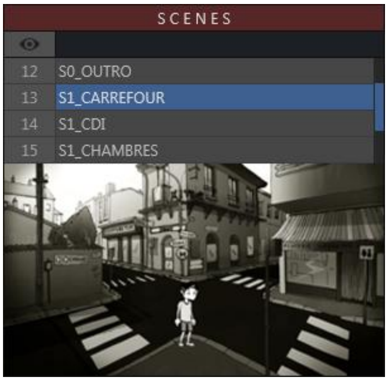
Export

If several people are working on the same project, it may be convenient to share scenario information within the team. The goal of seccia.dev is to encourage script writing directly from the software without going through an intermediate tool, such as word processing, as this allows to limit the production steps. It is therefore possible to export the full scenario into a single PNG file via the SCENARIO/Export scenario menu. When nodal boxes and sections have been defined, the capture should give all the scenario guidance. Box-specific comments are exported to an HTML file.

Good practice

Given the large number of puzzles to be expected in a game and therefore the number of boxes to be set up in the project scenario, it is preferable to prefix their name by referring to the chapter number. If the game has only one chapter, then it would be wise to split the narrative into deeds for development. It is not a mandatory rule, it is up to you to decide according to your history in order to adopt the most appropriate method.



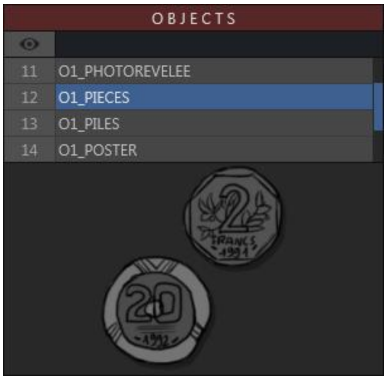


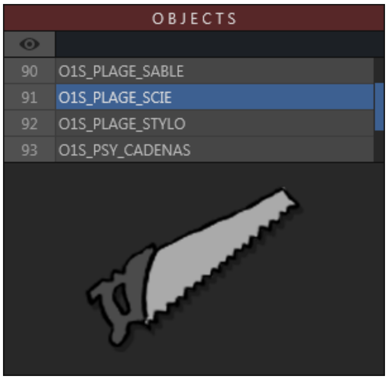
With regard to assets, the nomenclature is very simple with one exception for objects. The chapter number can also be included in the names of the assets.

S_HOME S_LIVINGROOM P_HERO C_INTRO E_NIGHT

The first letter, as we have seen before, is imposed and allows to define its type. A short and precise word is enough to describe the assembly. If you opt for the numbering of the chapters:

S1_HOME S1_LIVINGROOM P1_HERO C1_INTRO E1_NIGHT





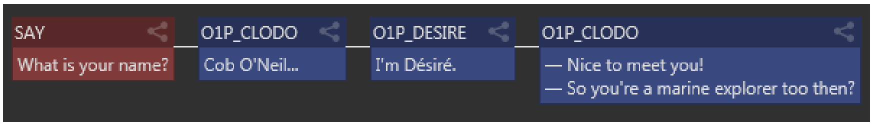
With regard to the exception, an additional letter makes it possible to distinguish the different types of objects.

OC_HERO OI_KEY OS_DOOR

Or else:

O1C_HERO O1I_KEY O1S_DOOR

The letter C indicates that it is a character, whereas the letter I indicates that it is an object to be placed in the inventory and will therefore need one or more icons. The letter S is reserved for objects present in scenes that will have to interact with the player. Static objects of a scene not having any interaction or having only a decorative function to furnish the place or manage the depth (perspective) will be to be implemented among the stills. Again, there is only good practice.



SAY
What is your name?

O1P_CLODO
Cob O'Neil...

O1P_DESIRE
I'm Désiré.

O1P_CLODO
— Nice to meet you!
— So you're a marine explorer too then?

There are usually two types of dialogues: conversations and contextual replicas. To better differentiate them it is better to take the name of the character or the place.

D_OLDMAN D_HOUSE D_HERO

For roles, the software needs to associate a scene with a role to edit the links and synchronize them during the game. On the same principle, choose a name identical to the stage or the player.

R_HOUSE R_GARDEN R_HERO

As for riddles, the title of the boxes is crucial to find it quickly and then optimize the debugging work. I strongly advise you to use the following syntax for the blue boxes:

chapter_sequence_label

The chapter allows the scenario to be cut into several parts.

The sequence is a single short word representing a situation, for example: intro, evasion, hunt, fight, negotiation...

The label specifies the context.

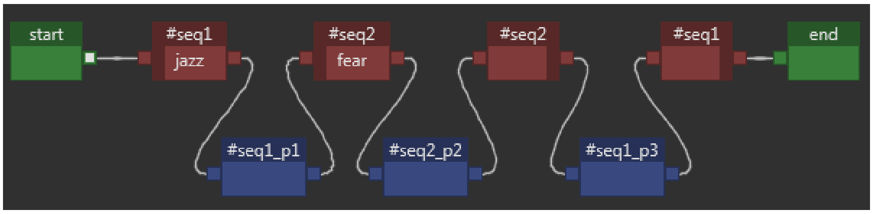
If necessary, add a note to your box to enrich the context. For use of objects, it would be the name of the objects concerned:

O_KEY + O_DOOR or O_WATER + O_BOTTLE = O_FULLBOTTLE

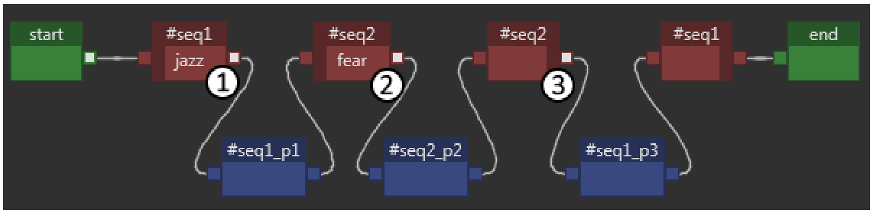
It is important to apply ranges to know easily and at any time the progression of the player via the functions related to sequences like started or ended. Think about taking the name of the sequence in each puzzle. A tool via the popup menu allows you to quickly rename several boxes after selecting them.

The text of the body of the incoming sequence box could indicate the musical atmosphere of the sequence. It is up to you to define in advance a list of atmospheres specific to the game (love, joy, sadness, fear, anger, shame, intrigue...). A very useful information for the artists and especially for the composer who will be led to explore the universe of the game.

The sequences can merge and it is moreover one of the main interests of the beaches. Coupled with the indication of the musical atmosphere, it becomes convenient to chain music with melted from one sequence to another. In the example below, the music of the sequence 2 sequence on the music of the sequence 1.



To do this, it is sufficient to interact with a role at the following locations:

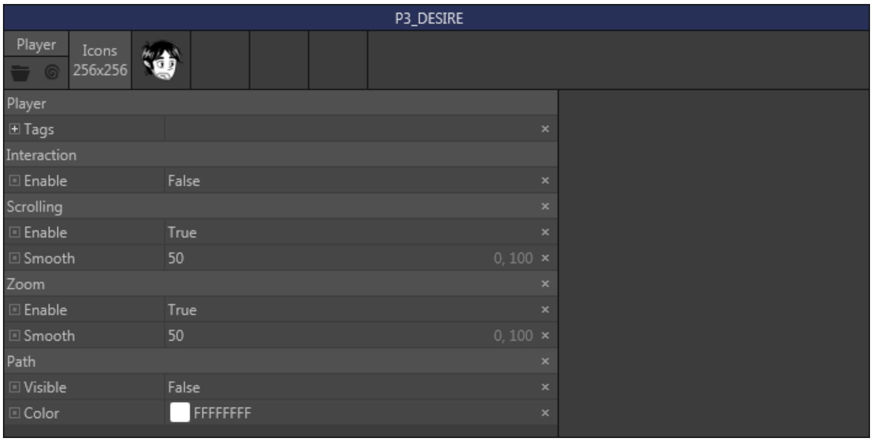


```
1) Play the song "SONG1" 2) Play the song "SONG2" in the
fondue chained 3) Play the previous song in the fondue
chained
```

PLAYER

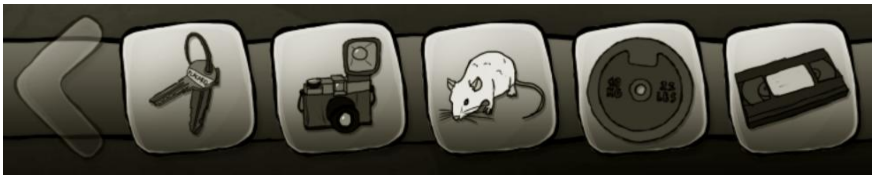
The player editor allows both to control the character incarnated by the player and to manage his inventory. It is quite possible to have multiple inventories by changing characters via the code or interface of the game.

The camera is associated with the current player and can be set directly from the editor.



P3_DESIRE

Player		
⊞ Tags		
Interaction		
⊟ Enable	False	
Scrolling		
⊟ Enable	True	
⊟ Smooth	50	0, 100
Zoom		
⊟ Enable	True	
⊟ Smooth	50	0, 100
Path		
⊟ Visible	False	
⊟ Color	⬜ FFFFFFFF	



Icon

A player has up to four icons named and numbered from 0.png to 3.png. The default icon is the first of the list and it is required to be able to display the character's head in the inventory bar. The set_player_icon function allows you to specify the index of the current icon. That said, if the game does not allow to incarnate several characters, then no icon is finally necessary.

The icons are PNG files and must be placed in the ICON folder of the assembly. You can also click on the preview of the icon to replace the file with another.

The optimal image size is 256x256 pixels in 32 bits. The texture will be loaded in memory on the graphic card only if the icon is present in the inventory bar.

Scroll & Zoom

It is possible to enable or disable the automatic scrolling of the scene relative to the horizontal position of the character by clicking on the Scrolling check box or by calling the enable_scrolling and disable_scrolling functions.

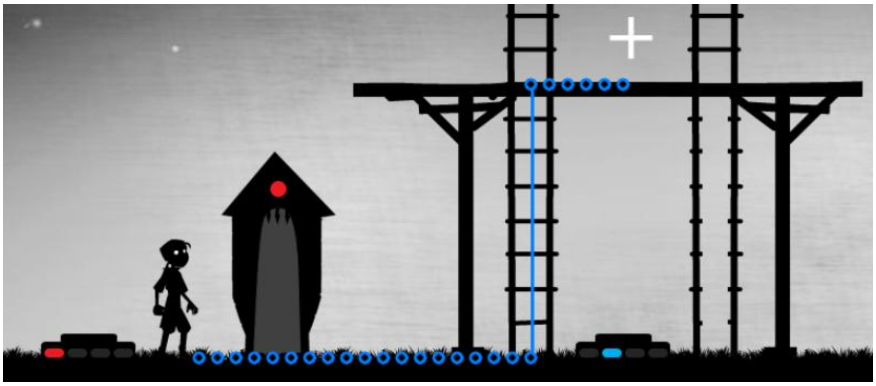
The scrolling can be refined by a smooth effect, allowing a gradual deceleration to be applied instead of a sharp and sharp stop.

The same goes for zooming the camera via the Zoom check box and the enable_zoom and disable_zoom functions.

There are other functions to control the camera such as set_shake and set_wave. These functions are documented in the software help page.

Trajectory

By default, the path taken by the character during a move never appears on the screen. This option allows you to make it visible and choose its color.



Note that the image of the tablet used to draw the trajectory is located at the IMAGE\WAY.png location. Remember that you can go to the Project page to view the full list of images from the game interface.

Code integration

The first step is to associate a player with a character (object) via the control function. A player can control only one object at a time. Whatever the object is linked, the inventory is unique to the player.

```
control P_HERO OC_HERO
```

The second step is to select the current player, i.e. the active player, thanks to the switch function.

```
switch P_HERO
```

That's it.

In case of multiple inventory, you must first define the list of players to be included in the inventory bar. The current player is never displayed.

```
set_player_list P_HERO P_FRIEND
```

We must not forget to define the initial scene of each player.

```
set_player_scene P_HERO S_HOUSE
```

To change the icon of a player:

```
set_player_icon P_HERO 1
```

OBJECT

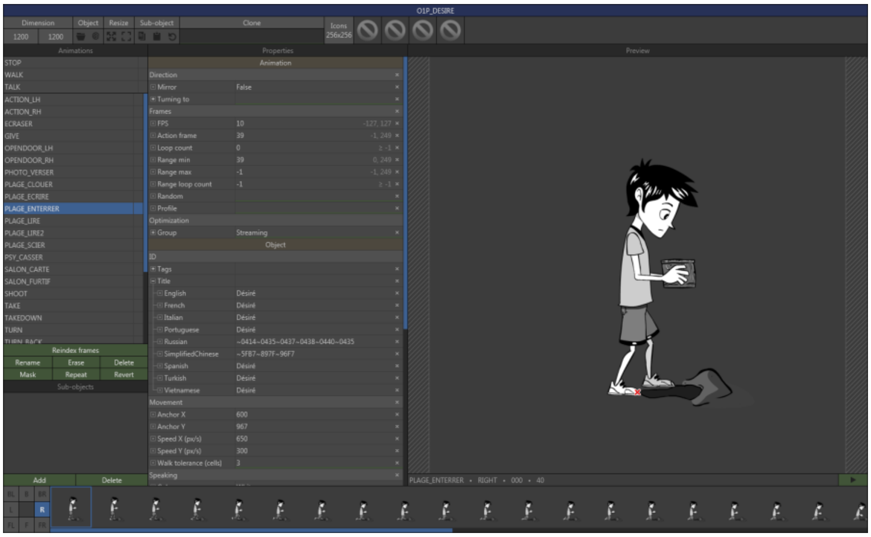
The object editor makes it possible to create objects in the broad sense of the term and to set them up and then handle them in the scenes. The generic object term is therefore used to mention both active objects and static objects of the scene, items of the inventory and characters of the game.

The first column of the interface lists the three default animations STOP, WALK and TALK even if they are empty, and just below the name of your animations. Each animation has its own settings.

The second column displays the properties of the object, animations and sub-objects by category.

The third column is reserved for viewing the selected frame.

Selected animation frames are displayed at the bottom of the editor.



STOP
WALK
TALK
ACTION_LH
ACTION_RH
ECRASER
GIVE
OPENDOOR_LH
OPENDOOR_RH
PHOTO_VERSER
PLAGE_CLOUER
PLAGE_ECRIRE
PLAGE_ENTERRER
PLAGE_LIRE
PLAGE_LIRE2
PLAGE_SCIER
PSY_CASSER
SALON_CARTE
SALON_FURTIF
SHOOT
TAKE
TAKEDOWN
TURN
TURN_BACK

Animation		
Direction		
Mirror	False	
Turning to		
Frames		
FPS	10	-127, 127
Action frame	39	-1, 249
Loop count	0	≥ -1
Range min	39	0, 249
Range max	-1	-1, 249
Range loop count	-1	≥ -1
Random		
Profile		
Optimization		
Group	Streaming	
Object		
ID		
Tags		
Title		
English	Désiré	
French	Désiré	
Italian	Désiré	
Portuguese	Désiré	
Russian	~0414~0435~0437~0438~0440~0435	
SimplifiedChinese	~5FB7~897F~96F7	
Spanish	Désiré	
Turkish	Désiré	
Vietnamese	Désiré	
Movement		
Anchor X	600	
Anchor Y	967	
Speed X (px/s)	650	
Speed Y (px/s)	300	
Walk tolerance (cells)	3	
Speaking		

Subject matter

In the second property column, you will find an exhaustive list of detailed object parameters in the following table:

ID	
Title	The title is displayed on the screen when the player interacts with the object. For example, when the mouse cursor overflews the object's interaction area. If the field is empty, the interaction will be limited to the selections. The pipe separator lets you set multiple titles for the same object. By default, only the first title of the list will be active but thanks to the set_title function, you can choose another title by specifying its index starting with 0.
Movement	
Anchor X/Y	This point, marked by a red cross, is used to manage the depth of the scene, i.e. the order of display of the objects. Because given the environment of the scene in two dimensions only, the co-ordinate Y is interpreted as a co-ordinate Z: the smaller this value and the more the object will be removed from the camera. This point represents the base of the object in contact with the ground or a support. For example, for a character it will be the feet while for a statue it will be the base. If the object is placed on a table, another property is provided in the scene to define the height of the object in relation to the ground. This case of figure is relevant only if the character is to move both behind and in front of the statue. The values are in pixels and relative to the PNG image. You can either enter the values or double-click on the preview. Finally, if your object is supposed to change direction, the position X must be at the center of the image to avoid a transition shift. I therefore advise you to center your characters on the basis of the code also change the value of the spin.
Speed X/Y	This is the speed of walking in pixels per second. Note that these values can be overdefined in the stage editor if your perspectives differ between two scenes.
Walk tolerance	When the player clicks on a part of your set, the character automatically moves to this point by travelling the shortest possible path with the least possible change of direction. If the destination is too close to the current position of the character, the move will be seen as a quick jump where the animation of the walk will be dryly interrupted. To

	avoid this unpleasant graphic effect, just set the minimum distance allowed in number of cells to trigger a move. This constraint is ignored when the player interacts with an element.
Speaking	
Color	If your object is able to speak, you can customize the color of the replicas via a list of 16 predefined colors. The RGB value of each color is editable from the project page if it does not suit you.
Movie style	If the option is enabled, the replicas will appear at the bottom of the screen as movie subtitles. If it is disabled, the replicas will appear above the character.
Avatar	To display the photograph or the avatar of the character who speaks, you must select an object from the list. TALK animation is played when the character keeps the speech and STOP animation when the player is prompted to choose an answer. STOP animation is used instead of TALK animation if the latter is absent. If STOP animation does not contain images, no avatar is displayed at the time of choice. The parameters of the Avatar group will be displayed in the object in question.
Avatar	
Layout image	The location of the image relative to the resolution of the game in pixels in the form "x, y, width, height". X and Y identify the top left corner of the frame.
Text layout	The position of the text relating to the resolution of the game in pixels in the form "x, y, width, height". X and Y identify the top left corner of the frame.
Text	The alignment of the text in the frame. By default the text is centered vertically and horizontally.
Darkness	Allows darkening of the entire scene to bring out theavatar and its text. The value is between 0 and 100. This option is ignored if the STOP animation does not contain any images at the time of selecting a replica.
Interaction	
Enabled	This option allows you to disable possible interactions with the object. You can change this value from scripts

	with the enable and disable functions. If your object does not have a title, it is preferable to disable interactions to avoid getting incomplete sentence constructs.
Bounding box	By default the active area of the object is defined by a rectangle, aligned to the X/Y axes and encompassing all pixels of the current frame. This rectangle can therefore be different from the size of the image. By deactivating the option, you will get more precision based on a pixel detection of your images. Of course, this precision has a significant cost for high resolutions and animated objects. I advise you to activate it only in case of justified necessity.
Subjects	This option disables possible interactions with sub-objects. You can change this value from scripts with the enable_subs and disable_subs functions. If your sub-object does not have a title, it is preferable to disable interactions.
Graphics	
Adjustment	The adjustment is applied at the time of the game generation.
Animated FX	You can associate a Flame internal shader to simulate a flame effect. The internal shader will use the alpha component of your PNG images as the opacity value, and the RGB color will affect the shade of the flames. Two other parameters will allow you to adjust the animation speed and the rendering scale.
Monochrome tint	In monochrome mode, the texture format is converted by applying a palette of two colors: your custom color from the property to RGBA format and transparency. The images are then coded in two bits only, so a 32-bit texture allows you to store up to 32 textures in this ideal mode to compress silhouettes. If your color is 0.0,0 then the monochrome mode is disabled.
Optimization	
Pack size	seccia.dev generates texture atlases to limit the number of textures to be loaded to runtime by the graphics card. A texture atlas is a PNG file containing several images separated by two transparent pixels in order to optimize the loading time. Moreover, the

	sizes expressed in pixels are variable to save graphic memory. It is better to have three textures of 512x512 than a single texture of 1024x1024 to save 262144 pixels. The maximum size of texture atlases is modifiable. By default the value is 2048 (understood 2048x2048). If your images are larger than the maximum size set, they will then be resized to the detriment of their quality.
Low quality	This option allows you to divide the size of all the textures of the object by two in order to gain in performance: four times less greedy! But artifacts could degrade the rendering.
Trim frames	By default the images are cut to avoid leaving empty portions without pixels. It is sometimes useful to disable this option, especially to display objects in the generic.

Animation

The editor uses the folder tree to list the animations. A folder represents an animation containing frames from all directions in PNG format.

The four primary directions are RIGHT, LEFT, FRONT and BACK. The four secondary directions are FL (front left), FR (front right), BL (back left) and BR (back right). Secondary directions are useful if you want to make a journey in eight directions. When the object has only one direction, it is preferable to use RIGHT. No direction is required.

The directions of an animation can contain a different number of frames. The index of a frame must be between 0 and 249 (maximum 250 images per direction). The index is formatted on three characters.

```
O_MYOBJ\ANIM\STOP\BACK_000.png
O_MYOBJ\ANIM\STOP\FRONT_000.png
O_MYOBJ\ANIM\STOP\RIGHT_000.png
O_MYOBJ\ANIM\WALK\BACK_000.png
O_MYOBJ\ANIM\WALK\BACK_001.png
O_MYOBJ\ANIM\WALK\FRONT_000.png
O_MYOBJ\ANIM\WALK\FRONT_001.png
O_MYOBJ\ANIM\WALK\RIGHT_000.png
O_MYOBJ\ANIM\WALK\RIGHT_001.png
```

Directorate	
Mirror	The LEFT, BL and FL directions can be determined by the RIGHT, BR and FR directions by applying a mirror effect if the option is activated. The first advantage is to limit the size and number of textures.
Turning to	By default, there is no transition when the character changes direction. To add it, you must specify the target animation in this field for each transition. For example, if you want to create a transition to move from LEFT direction to RIGHT direction. You must first create a new TURN_LEFT animation (name to choose) with RIGHT_*.png frames where the character will turn right to left, and then choose the LEFT option. In the other direction, it is the same principle by adding LEFT_*.png frames to TURN_RIGHT animation.
Frames	
FPS	The number of frames per second.
Action frame	By default, an event is called at the end of an animation, i.e. either at the index of the last frame or at the -1 index. In some situations, for example for TAKE animation, the event would be better triggered at the time of taking the object.
Loop count	The animation can be played once or several times in a loop. Specify -1 for an infinite loop. For the three main animations, the loop is necessarily infinite.
Min/max range	You can create a subloop by defining the initial frame and the final frame. Specify -1 to use the last animation frame.
Range loop count	The number of loops of the previously defined range.
Profile	The profile allows you to change the characteristics of the speech. For more information, please refer to the Word section of this chapter.
Optimization	
Group	It is very important to group atlases of textures according to the use of animations. If you have an animation played only once in the game, it is not necessarily necessary to keep it permanently in

	memory. There are different ways to proceed according to the mode of use. Loaded with object: By default the animation is loaded with the object when it is present in a scene. It is the mode without optimization. Loaded only for scene: The animation is loaded in memory only at the opening of the specified scene. No interest if the object is found in a single scene. Loaded on the fly: The animation is loaded entirely before its appearance on the screen, then unloaded at the end of the playback. A slight expectation can be felt if the animation is voluminous and even more so if the graphics card is not performing well. Streaming: The animation is loaded image by image as for a video. This mode is to be tested on small configurations to be sure that the FPS of the animation can be guaranteed to divide the versions by the two versions.

Sub-object

Sub-objects are rectangular and interactive sub-parts of the image. In the same object, if you want to have several areas clickable with different names, you can add sub-objects. You should not confuse sub-object with rectangular portion. Sub-objects only allow to declare the elements inheriting your object, for example the hat or the cane of a character. Then you have to define the clickable rectangle on all necessary frames. Collision detection on the pixel of the object is also taken into account for sub-objects.

After creating and selecting a sub-object, new properties appear in two new sections: Frame and Sub-object.

Concerning first the identity of the selected sub-object:

ID	
Name	The name of the sub-object. A color will automatically be assigned to it in the editor to differentiate rectangles.
Title	This is the title of the sub-object. If the field is empty, the interaction will be limited to the selections.

Concerning the properties of the selected frame:

Interaction	
Rectangle	The coordinates of the rectangle. Another way to define the rectangle is to select the portion of the image by remaining pressed on the right mouse button. You can use the toolbar to copy the current coordinates in memory to quickly apply them to other frames.

Inventory

L的asset Object offers a list of four customizable icons. Only one icon can be used at a time and display in the inventory bar or in place of the mouse cursor.

The icons are PNG files of 256x256 pixels preferably. They can be lower or higher but must be square. To change the icons, you can click on the preview to browse your folders and choose a new file. The image will be resized if needed. You can also go through the Windows Explorer as for animations.

During the game, to change the current icon, specify the new index with the set_icon function.

```
set_icon O_BOTTLE 1 set_title O_BOTTLE 1
```

One of the interests of this feature is to be able to change the appearance of the object after a riddle resolution without creating new objects. For example: filling a bottle, opening a can or more generally changing the state of an object. Changing the icon often involves changing the title.

Clone

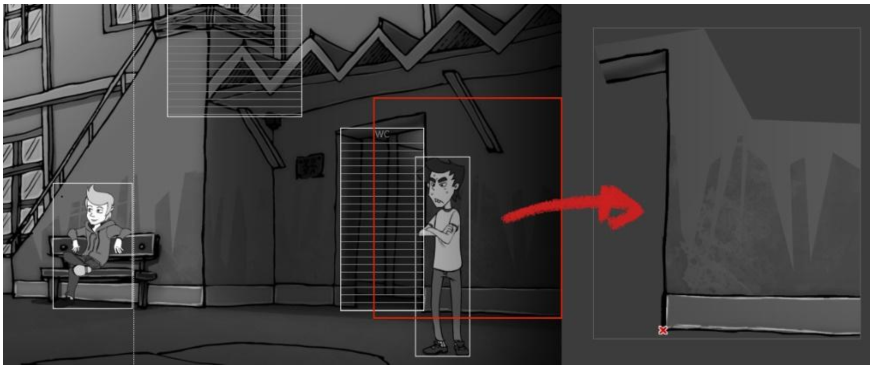
Sometimes scenes need to duplicate objects on the screen, but there is no notion of instance in seccia.dev. It is an arbitrary choice because I wanted to simplify the development of adventure games to facilitate the task to creators. And it is in any case quite rare to instil objects in a Point & Click or Visual Novel, even more widely in a narrative adventure game unlike a Shoot & Em Up. However, having to duplicate the resources of an object in seccia.dev is not however a reasonable option, hence this notion of cloning.

A clone is an object in its own right that shares the graphic resources of another object. It is not simply a reference to another object. For this to work, the original object must be in the same scene as its clones. To be more precise, it is the graphic resources that must be loaded in memory so that the clones can access it.

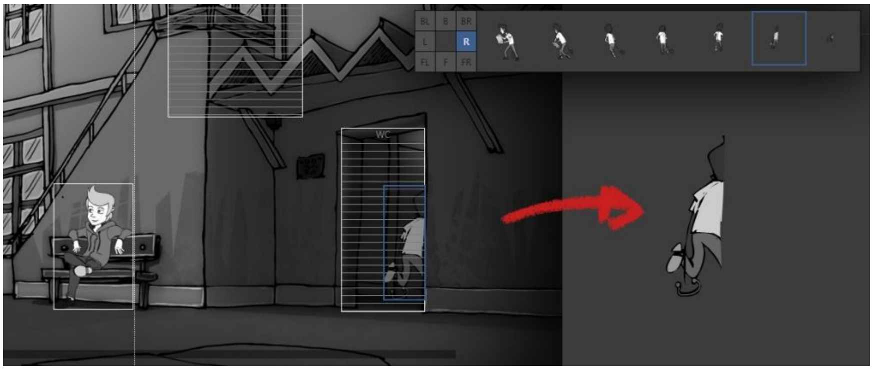
After selecting an object to be cloned, many properties will be overshadowed to give way only to clone-specific properties.

Mask

You can easily remove an image portion on a frame by applying an exclusion mask. I've used this technique in the Desired game as David leaves the toilet at all speed.



The right part of the wall is an object positioned above the decor. It is used to manage the entrance and exit of Desire. When it gets closer to the door, the wall displays above and allows to give the illusion that the character enters the room. But David's displacement is already included in the animation, i.e. the object does not change XY position and does not allow to reverse the order of display of the elements. In order to obtain the same illusion, the character should be cut off on the last three frames of animation manually with precision using a graphical software. Fastidious?

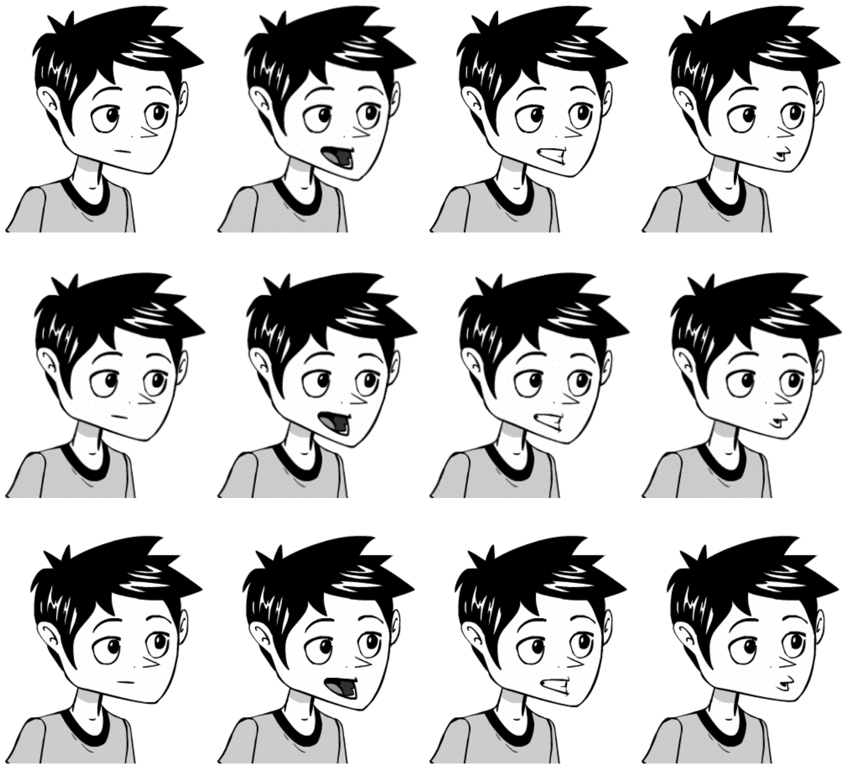


You can do this more easily by selecting a frame of the sequence and then clicking on the Mask button and editing the PNG. The editor will ask you to choose the scene in question and the object correctly placed as a mask: here it is the wall. Repeat the manip if necessary on the other frames by clicking on the Repeat button. In case of error, don't worry, the original files are kept and can be restored by clicking the Revert button.

Word

You have to provide three head inclinations so seccia.dev can pick the frames randomly, applying constraints according to the selected profile. Good to know: the algorithm will never draw by lot twice the same image.

The inclinations must contain identical facial poses. If your animation has four poses, then you will need twelve frames in total (i.e. 4 poses x 3 inclinations) as shown in the example below.



The images are numbered from 0 to 11 from left to right and from top to bottom. The first line corresponds to the normal inclination of the head. The second to the downward inclination and the third to the upward inclination. Note that the inclinations must be barely perceptible to obtain a natural rendering.

Code integration

There is an important subtlety to know right now about objects. It is necessary to distinguish between the three modes of use existing when handling objects. It is crucial to understand the meaning of functions and why they are divided into three categories in the documentation.

We have seen in this chapter how to create and configure an object from the editor but this is not the only method, some properties are also accessible through scripts. All the functions to read or modify these properties are presented under the Object category of the help page. You will even find additional features such as kill and revive, not present in the editor.

When an object is placed in a scene, it takes another form of use and becomes an object scene (a stage object) because its characteristics are dependent on the scene. It is therefore necessary to differentiate the notion of object and the notion of object of scene because there may be as many objects of scene from the same object as from scenes. Changing the title of an object, will have an impact on all the objects of scene since this property is dependent on the object. On the contrary, changing the TALK animation by default with the function set_default_talk will impact only the object of scene. The functions relating to the objects of scene are documented in Scene Object of the help page.

The third difference is still a little more subtle because the objects of stage finally have two modes of use. The second mode concerns the attributes of the object that do not persist beyond the scene. If you start a movement with walk or start_path to name only two functions, and you change scene during the move, the position will not be recorded. Upon return, the object will return to its original position. These functions are arranged in the Play section of the help page.

Good practice

It is important to properly name the animations by classifying them by place or by action to prepare the optimization of the game memory and limit to the maximum the refactoring that will have a negative impact on your deadlines.

If an animation is specific to a scene and the object appears in several places, you can opt for one of these three modes of optimization.

Only for scene	Advantage: No latency when playing animation. Disadvantage: Loading textures when opening the scene even if animation is not played.
On the fly	Advantage: Less textures to load if animation is not played. Disadvantage: Significant latency before playing animation due to loading of all frames.
Streaming	Advantage: Less textures to load (frame by frame) and very little latency before playing animation. Disadvantage: On some low performance hardware configurations, animation may slow down and not be played at the right speed, especially if the speed of animation is high.

As a general rule, it is better to stream large and non-recurring animations. The additional Low option will allow you to significantly reduce the weight of animations that do not require a high definition, for example a cloud of smoke and debris occupying the entire space of the scene.

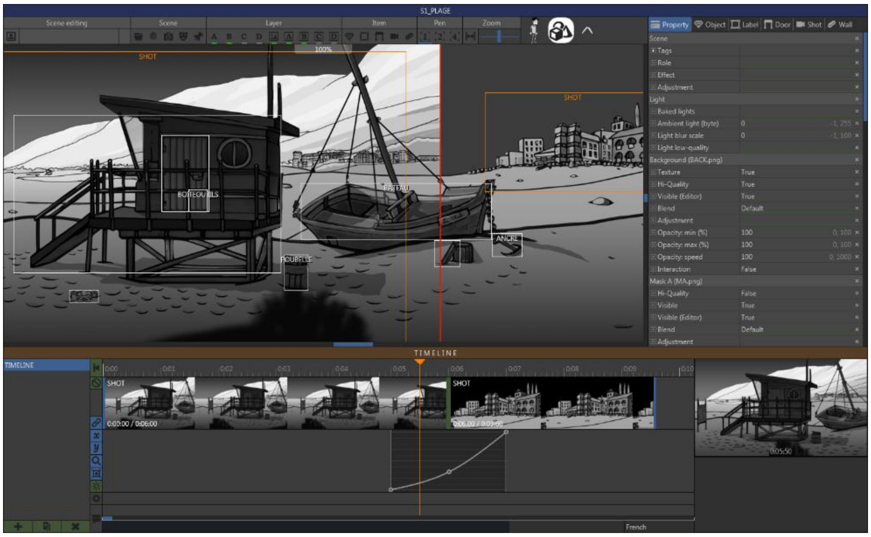
seccia.dev is not so stupid, it is able not to unload an object if it is also present in the new scene to load. If your main character is therefore in all scenes, its resources will never be unloaded during the game.

SCENE

The stage editor allows you to compose the levels of the game by placing the objects and adding the necessary interactions. Most of your time will be devoted to this editor.

One of the forces of seccia.dev lies in its simplicity to define the walking area of the character and to associate an action with a behavior. In the majority of cases when the player decides to recover an object present in the setting by clicking on it, the character must first move to this object and possibly play the appropriate animation to pick up the object. This sequence of steps is possible without writing a single line of code thanks to the grids and cells that we will detail later.

You can also create in-game kinematics thanks to the timelines by placing shots in your scene.

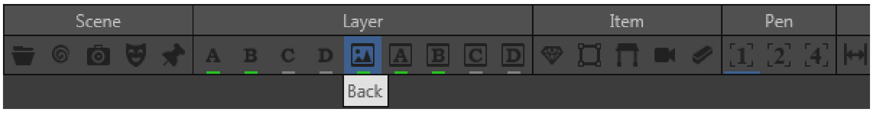


Scene		
Tags		
Role		
Effect		
Adjustment		
Light		
Baked lights		
Ambient light (byte)	0	-1, 255
Light blur scale	0	-1, 100
Light low-quality		
Background (BACK.png)		
Texture	True	
Hi-Quality	True	
Visible (Editor)	True	
Blend	Default	
Adjustment		
Opacity: min (%)	100	0, 100
Opacity: max (%)	100	0, 100
Opacity: speed	100	0, 1000
Interaction	False	
Mask A (MA.png)		
Hi-Quality	False	
Visible	True	
Visible (Editor)	True	
Blend	Default	
Adjustment		

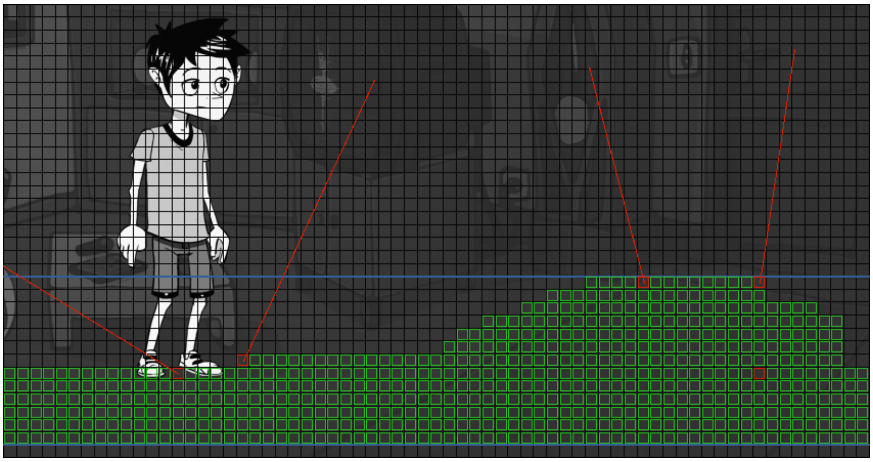
Scene

A scene consists of a main decoration, layers in the front or background, stills, objects and interactive areas.

After creating a new scene asset, the first step is to import an image in PNG format to replace the default main set. From the Editor's Scene toolbar, right-click the Back icon and import a PNG file. Otherwise, open the Assembly folder to transfer the PNG file directly via Windows Explorer.



The main decoration must bear the name BACK.png and be located in the LAYER folder. The dimensions of the image define the size of the scene and therefore the number of cells of the grid. If the size of the scene differs from the resolution of the game, black bands will be inlaid or a horizontal scroll will be applied according to the parameters of the project. The size of a cell is set to 16x16 pixels regardless of the resolution of the game.



There are two types of layers: the distant decorations (FAR) and the masks. You can add up to four distant decorations and four masks per scene. The order of display of the nine images, including the main decoration, thus presents itself, from the foreground to the background.

MA.png	The A mask displayed in the foreground (the closest to the camera).
MB.png	The mask B hidden by mask A.
MC.png	The C mask hidden by the B mask.
MD.png	The D mask hidden by the C mask.
BACK.png	The main decoration hidden by the masks that defines the size of the scene.
FA.png	The first distant decor.

FB.png	The second distant decor.
FC.png	The third distant decor.
FD.png	The farthest decor from the camera.

If the main decoration is completely opaque, the distant decorations will not be visible. Layers can be activated or deactivated by code by adding molten effects chained.

THE PROPERTIES OF THE SCENE

When no item is selected, you can access the scene properties from the list at the top right of the editor.

Scene	
Tags	You can set up up to four tags per scene. They allow you to identify scenes otherwise than by their name in order to group them by category. A fifth tag is editable by the set_asset_tag function.
Role	It is important to associate a role with a scene to facilitate the interaction between the two publishers. Moreover, it allows you to automatically launch a role at the opening of a scene and stop it when it closes. It is thereforadvisable to always create a role for stage to implement the specific actions, conditions and events of the scene.
Effect	An effect can be applied to the entire scene. If the property is empty, the scene will use the Effect field of the project.
Adjustment	The adjustment is applied at the time of the game generation.
Light	
Baked lights	Allows to overdefine the property defined in the properties of the project.
Ambient light	Allows to overdefine the property defined in the properties of the project. -1 uses the value of the project.

Light blur scale	Allows to overdefine the property defined in the properties of the project. -1 uses the value of the project.
Light low-quality	Allows to overdefine the property defined in the properties of the project.
Bokeh	
Shape width/height	Allows to overdefine the property defined in the properties of the project. 0 uses the value of the project.
Size	Change the size of the shape of the bokeh by specifying a value between 0 and 100. This value is taken into account with the camera zoom to adjust the size in real time.
Max zoom	Change the maximum zoom value by specifying a value between 100 and 400. If the parameter is 100, the zoom of the camera will have no impact on the size of the bokeh.
Background (BACK)	
Texture	Allows to exclude the recording of texture in the build.
Hi-quality	If the option is enabled, the decor is divided into several textures of 1024x1024. If the option is disabled, the decor is resized into a single texture of 1024x1024.
Visible (editor)	Change the visibility of the decor in the publisher.
Blend	Change the layer fusion mode. Default: The transparency simulation uses RGB values pre-multipled by the alpha value. Addition: The colors are added to produce the final rendering. Black pixels will become transparent.
Adjustment	Apply an adjustment only to the decor.
Opacity: min/max	Defines as a percentage of space used to produce a transition effect between two values.
Opacity : speed	Sets the speed multiply of the transition. At 100%, the factor is equal to 1.
Interaction	Intercept the player's click if the pixel is opaque.

Mask (MA, MB, MC, MD)	
Hi-quality	If the option is enabled, the mask is divided into several textures of 1024x1024. If the option is disabled, the mask is resized into a single texture of 1024x1024.
Visible	Change the visibility of the scene in the game.
Visible (editor)	Change the visibility of the decor in the publisher.
Blend	Change the layer fusion mode. Default: The transparency simulation uses RGB values pre-multipled by the alpha value. Addition: The colors are added to produce the final rendering. Black pixels will become transparent.
Adjustment	Apply an adjustment only to the decor.
Tile X/Y	Apply horizontally and/or vertically an image repeat to fill the empty space.
Offset X/Y	Horizontally and/or vertically the image in pixels.
Scroll speed X/Y	Change the horizontal scrolling speed of the scene. At 100% the mask scrolls as fast as the main decor, at 50% it scrolls twist as fast, at 200% it scrolls times fater and at 0% the speed is adjusted according to the size of the main decor and the mask.
Parallax	This is the shift parallelax. To create a depth illusion when the character moves away from the camera, it is necessary to emphasize the enlargement of the layers in the background compared to the background. By default, all layers have the same width factor.
Opacity: min/max	Defines as a percentage of space used to produce a transition effect between two values.
Opacity : speed	Sets the speed multiply of the transition. At 100%, the factor is equal to 1.
Interaction	Intercept the player's click if the pixel is opaque.
Far (FA, FB, FC, FD)	
Hi-quality	If the option is enabled, the decor is divided into several textures of 1024x1024. If the option is

	disabled, the decor is resized into a single texture of 1024x1024.
Visible	Change the visibility of the scene in the game.
Visible (editor)	Change the visibility of the decor in the publisher.
Blend	Change the layer fusion mode. Default: The transparency simulation uses RGB values pre-multipled by the alpha value. Addition: The colors are added to produce the final rendering. Black pixels will become transparent.
Adjustment	Apply an adjustment only to the decor.
Tile X/Y	Apply horizontally and/or vertically an image repeat to fill the empty space.
Offset X/Y	Horizontally and/or vertically the image in pixels.
Scroll speed X/Y	Change the horizontal scrolling speed of the scene. At 100% the remote decor scrolls as fast as the main decor, at 50% it scrolls twist as fast, at 200% it scrolls times fater and at 0% the speed is adjusted according to the size of the main decor and the remote decor.
Parallax	If the option is enabled, the decor will contain its size lookless of the zoom applied to the scene. This option is useful for very remote scenery (mountains, sky, horizon...) that must not be alternated by the distance of the camera.
Opacity: min/max	Defines as a percentage of space used to produce a transition effect between two values.
Opacity : speed	Sets the speed multiply of the transition. At 100%, the factor is equal to 1.
Interaction	Intercept the player's click if the pixel is opaque.

THE ACTIONS OF THE SCENE

In the role editor, some synchronised actions are specific to the scenes we list briefly below:

camera	Allows you to make a camera movement.
examine	Allows large display of an object.

jump	Allows you to change scene.
popup	Allows to display a popup window.
timeline	Allows you to launch a timeline.
Wait.	Allows you to pause before going to the next box.

THE CONDITIONS OF THE SCENE

In the role editor, certain conditions are specific to the scenes we list briefly below:

stage is	Lets you know the current or previous scene.
scene has	Allows you to know the visibility of an element of the scene.

THE EVENTS OF THE SCENE

In the role editor, some events are specific to the scenes we list briefly below:

on click	When the player clicks at a place of the stage.
We're going in.	When the player changes place and enters a scene.
we're going out there.	When the player changes place and exits the current scene.
on input	When the player performs an action with the mouse or touch screen.
on select layout	When the player clicks on a customizable button of the game interface.

Still

Stills can significantly reduce the number of objects to create in your project. These are still objects that will not interact with the player but will be able to manage the depth of the scene. Stills are not shared with other scenes and do not create any dependency. Thus, a stage library will also contain the stills. All images are stored in a texture atlas as for objects.

To add a still, right-click on the scene and then add stills to select one or more PNG files. Without going through the popup menu, you can drag and drop your files from the Windows Explorer provided you have opened the still mode. If the image is the same size as the main decor, it will be trimmed and placed in the right place of the scene. With this trick, the artist will be able to export the items in separate layers and the PNG will be easily imported without placing them by hand to the nearest pixel. It will only be necessary to set the still.

PROPERTIES OF THE STILL

Properties	
Tags	You can set up up to four tags per still. They allow you to group the stills by category. A fifth tag is editable by the set_still_tag function.
ID	Unique read-only identifier generated automatically at its creation.
Width/Height	The image size recovered from the file.
Name	The name of the still.
Visible	The visibility of the default still. Show_still and hide_still functions allow you to change the visibility dynamically by code. This can allow you to change the status of the items by overlaying them (lamps turned off or turned on).
X/Y	The position in pixels of the still in the scene. The position corresponds to the point at the top left of the rectangle.
Elevator	Allows to change the height of the still as for objects. The base is always located at the bottom of the rectangle and centred horizontally. This height can be negative. Stills are placed on the same layer as objects drawn after the BACK decoration.

Stage object

We have already developed the concept of stage object in the previous chapter. To sum up, it is a version specific to the scene that has their specificities.

PROPERTIES OF THE OBJECT OF SCENE

Properties	
UID	The object reference.
Tags	You can set up up to four tags per object. They allow you to identify objects other than by name in order to group them by category. A fifth tag is editable by the set_tag function.
Visible in this scene	Force the object's change of visibility at the stage launch. By default, the object keeps its visibility through the scenes knowing that it cannot be visible in two places. If you want to force its appearance at each stage launch, you need to change the value to True. Go through the code if you need to apply conditions.
Parent	Allows you to manage relative positions to other objects, labels or mouse cursor. You can use them for example to get a flashlight effect in a cellar.
X/Y	The position of the object in pixels. The position takes into account the pivot point defined in the object editor.
Elevator	Allows you to change the height of the object (position in the space and not the size of the object). If you have a table and a vase placed on it for perspective (and you have defined the anchor point at the base of the vase and at the foot of the table closed to the camera) you will have a concern for depth. Remember that the Y coordinate of your objects is also used as coordinated Z to manage the depth of the scene. Since the table and vase are not the same position Y, and the table is closed to the camera (Y table > Y vase) you will see the table drawn above the vase. To remedy this problem, you could go down the wrong point of the vase but unless you have a valid reason I advise you not to do so. The anchore point must be relative to the object and not to the solution already.
Elevator scattering	By default, the size change doesn't take into account the Elevator parameter.
Elevator rotation	By default, the rotation does not take into account the Elevator parameter.

Parallax	To create a depth illusion when the character moves away from the camera, it is necessary to emphasize the enlargement of objects in the background compared to the background. By default with the value 100, all objects have the same scope factor. If the value is equal to 0, the object will be considered as a very distant object and will never change size.
Speed X/Y	It is possible to overdefine the moving speed of the object. If the field is empty, seccia.dev recovers the specified speed in the object editor.
Placement	Allows you to change the order of display of an object. By default the objects are drawn just after the hand decoration (BACK). The step of the scene on the Z axis when the objects move is dependent on the placement. This feature is very useful for adding visual effects with greater flexibility.
Blend	Change the fusion mode of the object. Default: The transparency simulation uses RGB values pre-multipled by the alpha value. Addition: The colors are added to produce the final rendering. Black pixels will become transparent. Soft Light: Similar to Photoshop's soft light. Colors have no effect on black.
HUD coords	If the option is enabled, the object coordinates will be relative to the camera.
Drag	Source: The object can be slide to an object or label. Target: The object can receive a drag-and-drop object. Both: The object can be either the object slide or the recipient object.
Cheat	When the player remains pressed on the middle mouse button, an icon defined by the image CHEAT_OBJECT or CHEAT_DOOR is displayed at the location of the object.
Door	The doors allow to indicate to the player a change of location or a passage leading to another room. The text color and cursor may be different.
Skippable walk	Allows the player to shorten the move to the object by selecting it a second time.
Light Properties	

Enabled	Allows to turn on or off the light.
Ambient	Value between 0 and 255 to change the ambient light of an area. If the scene still has atmosphere lighting at 255, the light will have no effect. The value adds to the ambient lighting value of the scene. If the scene is darkened to 50, that a first light of 20 and that a second light of 10 illuminated the same area, the letter will receive an ambient value of 70.
Diffuse	The colour is applied to the illuminated area with or without transparency.
Angle	Value between 0 and 360. The 360 degree value allows you to have omnidirectional light.
Directorate	Value between 0 and 359 to define the direction angle. If the light is omnidirectional, the direction is ignored.
Distance	The lighting distance in pixels, -1 for infinity.
Detention	Value between 0 and 10. Light intensity is altered by holding as a function of distance. The distance between each each pixel and the source point of light is calculated and converted to a value between 0 and 1. This value is then raised to the holding power: pow(dist01, attn). If the distance is infinite or the holding is 0, no effect is applied.
Editor Properties	
Locked	Change the holding of the object in the editor. When an object is locked, it can no longer be selected via the mouse or shortcuts without going through the list on the Object tab.
Visible	Change the visibility of the object in the editor.

THE ACTIONS OF THE OBJECT OF SCENE

In the role editor, some synchronised actions are specific to the stage objects that we list briefly below:

animate	Allows you to play animation of an object.
examine	Allows large display of an object.

light	Allows to enable or disable the light of an object.
path	Allows you to start moving an object by following a predefined path.
select	Allows to reproduce a click on an object without interaction of the player.
show	Allows you to change the visibility of an object.
stop	Allows you to stop the movement of an object.
walk	Allows you to start moving an object via pathfinding.

THE CONDITIONS OF THE OBJECT OF SCENE

In the role editor, certain conditions are specific to the stage objects that we briefly list below:

object has	Allows you to know the state of the elements of an object.
scene has	Allows you to know the visibility of an object.

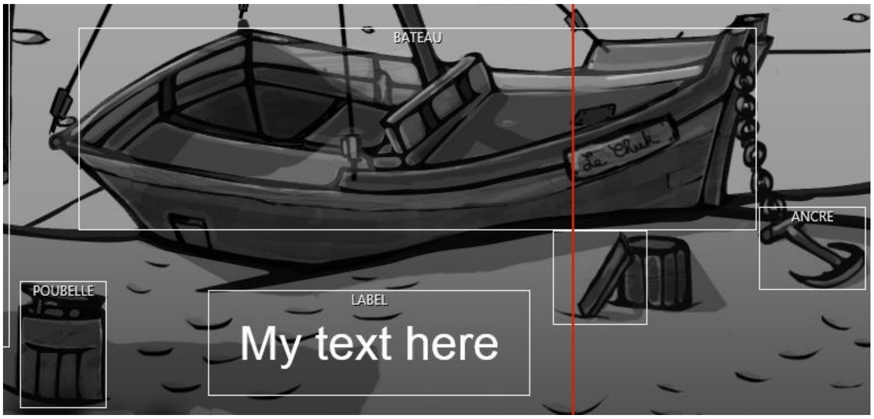
THE EVENTS OF THE OBJECT OF SCENE

In the role editor, some events are specific to the stage objects that we briefly list below:

on drag object	Allows you to receive a request to drag and drop and to be able to refuse if necessary.
on drop object	Allows you to receive a notification when an object has been deposited on no target.
on select object	Allows you to receive a notification when the player clicks on an object present in the scene.
on use label	Allows you to receive a notification when an object and a label are combined.
we use object	Allows you to receive a notification when two objects are combined.
on walk	Allows you to receive a notification when the pathfinding of an object is launched or stopped.

Label

A label allows either to define an interactive rectangular area without adding additional graphic assemblies to the scene, or to incorporate text on the screen.



LABEL PROPERTIES

Properties	
Tags	You can set up up to four tags per label. They allow you to identify labels other than by name in order to group them by category. A fifth tag is editable by the set_label_tag function.
Name	The name of the label without special characters.
Parent	Allows you to manage relative positions relative to other objects, labels or mouse cursor.
X/Y	The position of the label in pixels.
Width/Height	The size of the label in pixels.
Visible	Change the visibility of the label.
Enabled	Allows you to disable interaction with the player. By code you have to use the enabled_label and disable_label functions. If your label does not have a title, it is better to disable interactions.

Title	The title used to interact with the label. If the title is empty, the interactions will be limited to the selections. The pipe separator allows you to have several titles and thanks to the set_label_title function to be able to select it by its index.
Text	The text displayed on the screen inside the frame.
HUD	If the option is enabled, the label will be displayed over the objects and its position will be relative to the screen.
Forward	Allows to change the priority of interaction. By default the objects are priority, i.e. if an object and a label are superimposed, the player will not be able to reach the label by clicking on the object. If the option is enabled, the label becomes priority.
Size	Text size in pixels, 0 for default size.
Color	The color of the text.
Align	The alignment of the text in the frame. By default the text is centered vertically and horizontally.
Drag	Authorizes the label to receive a drag and drop.
Cheat	When the player remains pressed on the middle mouse button, an icon defined by the image CHEAT_LABEL or CHEAT_DOOR is displayed at the label location.
Door	The doors allow to indicate to the player a change of location or a passage leading to another room. The text color and cursor may be different.
Skippable walk	Allows the player to shorten the move to the label by selecting it a second time.
Editor Properties	
Locked	Change the label lock in the editor. When a label is locked, it can no longer be selected via the mouse or shortcuts without going through the list on the Label tab.
Visible	Change the visibility of the label in the publisher.

LABEL ACTIONS

In the role editor, some synchronised actions are specific to the labels we list briefly below:

show	Allows you to change the visibility of a label.

THE CONDITIONS OF LABEL

In the role editor, certain conditions are specific to the labels we list briefly below:

scene has	Lets you know the visibility of a label.

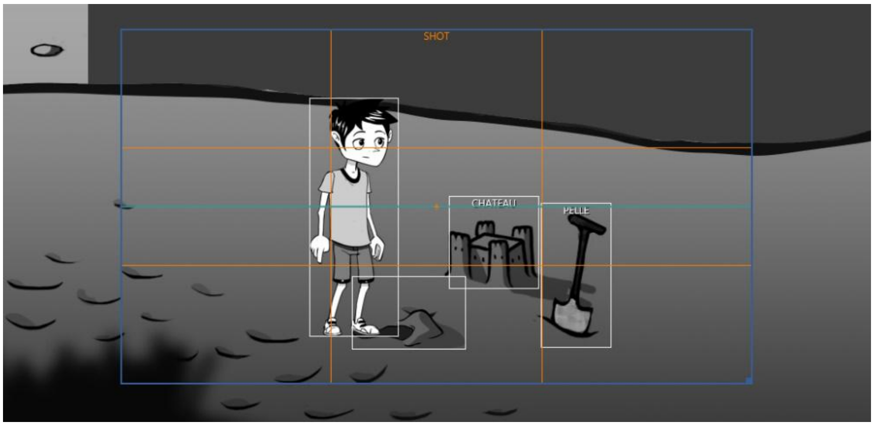
THE EVENTS OF LABEL

In the role editor, some events are specific to the labels we list briefly below:

on select label	Allows you to receive a notification when the player clicks on a label.
on use label	Allows you to receive a notification when an object and a label are combined.

Shot

Shots or camera shots allow you to set the location of the camera to add transitions by code or timelines. When a shot is selected, a grid will be displayed to help you compose your image according to the third-party rule. By selecting the shot and then holding the ALT key down, all guides will be hidden except the selected grid. It is possible to change its position and width. The height is automatically adjusted according to the ratio of your game defined in the project properties.



By code, you can activate a shot or make a transition between two shots by calling the shot function with settings to customize the effect. However, a more intuitive, convenient and elegant way is to use the timelines documented a little further in this chapter.

SHOT PROPERTIES

Properties	
Name	The name of the plan without special characters.
X/Y	The position of the plane in pixels.
Width	The width of the plane in pixels.
Editor Properties	
Locked	Change the plan lock in the editor. When a plan is locked, it can no longer be selected via the mouse or shortcuts without going through the list on the Shot tab.
Visible	Change the visibility of the plan in the editor.

SHAT ACTIONS

In the role editor, some synchronised actions are specific to the shots we list briefly below:

shot	Allows to launch a camera transition between two shots or between the current position of the camera and a shot.

Wall

Walls allow you to define shadow zones using dynamic lights. They do not exist visually and must match the lines of a decor. It is not possible to change their position by code.

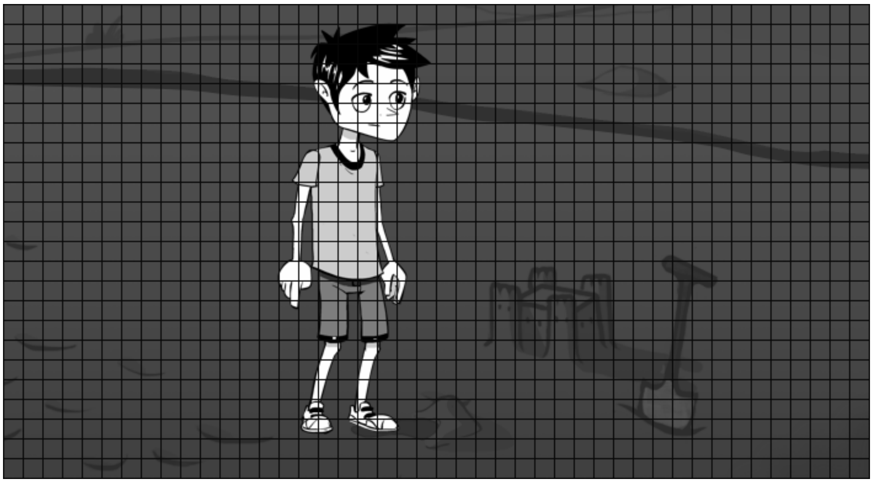
WALL PROPERTIES

Properties	
Name	The name of the wall without special characters.
X/Y	The pixel position of the first point of the segment.
X2/Y2	The pixel position of the second point of the segment.
Editor Properties	
Locked	Change the wall lock in the editor. When a wall is locked, it can no longer be selected via the mouse or shortcuts without going through the Wall tab list.
Visible	Change the visibility of the wall in the editor.

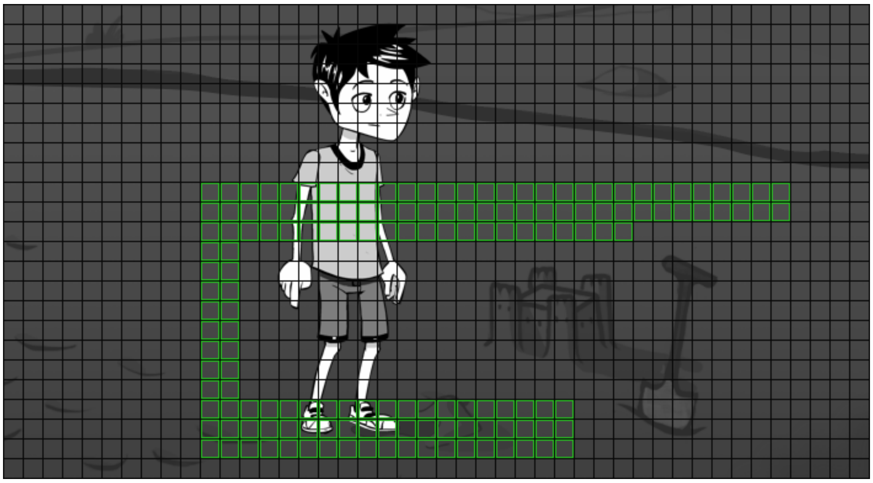
Grid & Cell

The grid allows you to define the moving areas and interactions with the elements of the scene (objects, labels, etc.). Only objects can contain grids up to eight. A grid is composed of adjustable cells. To edit a grid, just double-click an object. By pressing the keys on the numeric pad (0 to 7) you can change the grid selection to edit mode. To change the grid by code, you need to use the set_grid function.

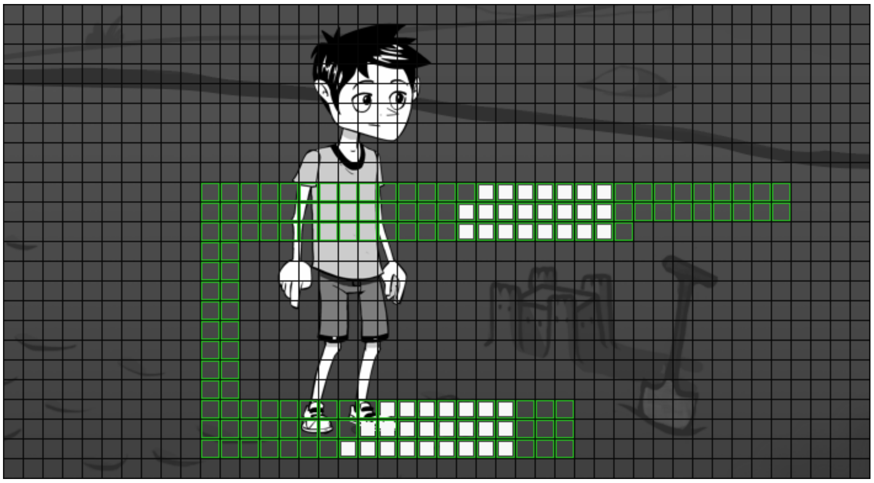
In grid editing mode, the entire scene is displayed in black and white with a less pronounced contrast (except the selected object) and lines appear to delimit the cells.



To add an active cell on the grid, press the SHIFT key and click on the desired cell with the left mouse button. As with a brush in a drawing software, you can hold the left button down to fill several cells quickly and even choose a larger brush size from the toolbar.



To select the defined cells, choose the brush that suits you and simply click on the cells that will become white like this.



You can delete them with the LED key or copy/cut them into the clipboard with the CTRL+C/CTRL+X shortcuts. The removal of cells can also be done with the CTRL and SHIFT keys by clicking on the left button. Finally, the arrow keys allow you to move the selection. When one or more cells are selected, a list of properties relating to cells will appear at the top right of the editor that we will now detail.

CELL PROPERTIES

Cell	
Index	The cell position on the grid in number of rows and columns.
Name	The cell name without special characters.
Event	If the option is enabled, the cell can call the event on reach cell.
Magnet	Allows you to place the object at the center of the cell at the end of a step. When the cell is linked to an action, this option is automatically enabled.
Walkable	Allows you to make a cell not accessible to walking. This option can be useful when using bridges and you want to customize cells on the character's trajectory.
Flag	Attributes a value between 0 and 9 to the cell. With the cur_flag function, you will know at a precise moment if an object is on a cell bearing one of these numbers.

Speed factor X/Y	Speed multiplier when the object is on this cell.
Scale factor	Multiplier of scale when the object is on this cell.
Walk animation	Force an animation when the object moves on the cell.
Walk directions	List of valid directions separated by spaces, when the object is on the cell. Useful for example to limit the object to one or two directions on a portion of the trajectory.
ENTER	
Position	You have to define a FROM cell and a TO cell so that the character can move from the FROM cell to the TO cell.
Previous scenes	Allows you to set a list of scenes. If the previous scene (the character's origin) matches a scene in the list, the move will then be made. If your scene only comes in, you can ignore this parameter.
Animation	Allows you to play an animation at the end of the trip. Only works with TO cells.
Directorate	Allows you to force the direction of the character at the end of the movement. Only works with TO cells.
SELECT Object	
Linked object	Allows to link the cell to an object. So when the player clicks on the object, the character will automatically join the cell before the event is called.
IF resolved	By default the following parameters are taken into account regardless of the progression of the scenario. It is possible to add a condition if you want to check the state of a riddle. In this case if the specified riddle has not been solved, the parameters will be ignored. Tolerance: This is the minimum number of cell to consider that the destination is reached. If the value is 0, there is no tolerance. If the value is 1, the contiguous cell will be allowed. Animation: Allows you to play an animation on arrival. Direction: Allows you to force the direction of the character on arrival. The direction is applied to the specified animation.
SELECT Label	

Linked label	Allows you to link the cell to a label.
IF resolved	cf. SELECT Object
USE free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (5606) free movement of goods (56)	
Linked object A	Allows you to bind the cell when two objects are combined. The object A is always from the inventory.
Linked object B	Allows you to link the cell when two objects are combined. The object B is always the destination object (player or scene).
IF resolved	cf. SELECT Object
USE trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606) trade mark (5606)	
Linked object	Allows to bind the cell when an object is combined with a label. The object is always from the inventory.
Linked label	Allows to bind the cell when an object is combined with a label.

IF resolved	cf. SELECT Object
DETACH Object	
Linked object	Allows you to link the cell when an object containing two elements is separated. To use this feature, it is necessary to add an ITEM_DETACH image to the project.
IF resolved	cf. SELECT Object

CELL ACTIONS

In the role editor, some synchronised actions are specific to the cells we list briefly below:

walk	Allows you to start moving an object via pathfinding.

THE CONDITIONS OF THE CELL

In the role editor, certain conditions are specific to the cells we list briefly below:

object has	Lets you know if the object is on a cell.

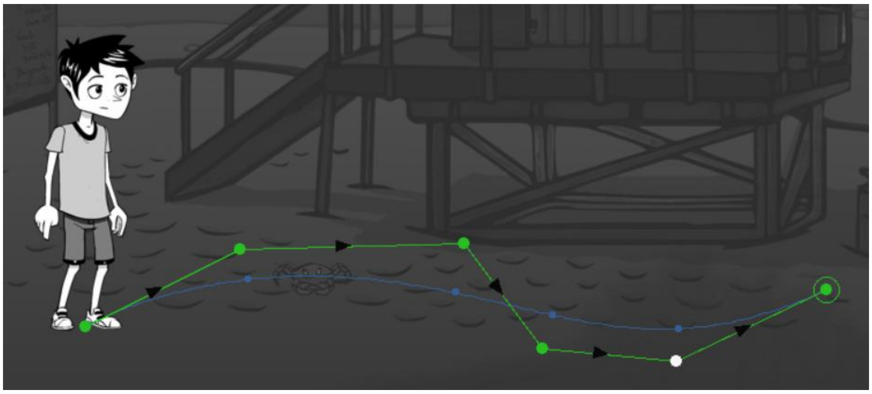
THE EVENTS OF THE CELL

In the role editor, some events are specific to the cells we list briefly below:

we get cell	Allows you to receive a notification when an object reaches a cell.

Path & Spot

There is another way to move objects by following a predefined path in a curve or straight line. As with grids, seccia.dev offers you eight pathes per object that can be activated with the set_path function. To edit the path, select the object and then Editpaths via the popup menu.



By holding the SHIFT key down, you can click on the left mouse button to add spots to your scene and thus trace a linear path. Paths have properties and as for cell, each spot has its own properties.

To select a spot, just click on its point with the left mouse button. If you click on a spot by holding the CTRL key down, you will select or deselect the spot according to its state.

For faster and more convenient selection of multiple elements at a time, just hold the left mouse button down to make a white circle appear on the screen. All spots coming into contact with this circle will be automatically selected. To not lose the current selection, press CTRL before the circle display and then release the key. To reverse the state of a spot, hold the CTRL key down during scanning.

Use the arrow keys on the keyboard to move the selected spots to a pixel in the desired direction.

Use the PageUp and PageDown keys to switch to the previous or next spot. The Home key will place the selection at the beginning and the End key at the end of the path.

It is possible to apply a curve to your path in order to break the linearity of the displacement and make it more natural thanks to the Spline property of the path. In this case, the position of the spots in the game will be defined by the position of the points of the curve.

PATH PROPERTIES

Path	
Spline	This is the shape of the curve. The larger the value and the more the curve will move away from the straight lines. The calculation time will also be longer. By default, the value 0 disables the use of the curve.

Speed	The speed of moving the object in pixels per second. This is the initial speed that can be modified by the spots.
Zoom	Allows you to take into account the Zoom values indicated in the spots.
Loop count	Sets the number of loops to perform. If the value is -1, the path will loop to infinity. The default value 0 allows you to make the journey only once.

SPOT PROPERTIES

Spot	
Index	The position of the spot in the list.
Name	The name of the spot without special characters.
X/Y	The position of the spot in pixels in the editor.
Speed	Allows you to change the speed of the object. The speed is interpolated between two spots. If the value is 0, the speed of the previous spot is then kept.
Zoom	Allows you to change the zoom of the camera. The value is interpolated between two spots. If the zoom is 0, then the zoom value of the previous spot is retained. This property is ignored if the Zoom of the path is disabled.
Break	If the value is greater than 0, the object will stop at the spot and wait for the specified time in seconds. If the value is -1, the object will stop without continuing its path. The continuous_path function will allow to resume the path.

SPOT ACTIONS

In the role editor, some synchronised actions are specific to the spots we list briefly below:

path	Allows you to start moving an object by following a predefined path in the editor.

THE CONDITIONS OF THE SPOT

In the role editor, certain conditions are specific to the spots we list briefly below:

object has	Lets you know the object is on a spot.

THE EVENTS OF THE SPOT

In the role editor, some events are specific to the spots we list briefly below:

on reach spot	Allows you to receive a notification when the object reaches a spot.

Drag & Drop

The drag & drop or drag & drop in French has a real interest in creating mini puzzles to accompany the narrative frame of your games. By clicking on an object and holding down the left mouse button, you can drag the object present on the screen to another object or a stage label. The logic is exactly the same on touch screen by sliding the object with the finger.

The first step is to determine the behavior of the object by choosing Source, Target or Both via the Drag property of the stage editor.

Source	The source is the object that can be controlled by a drag-and-drop. At the moment the player decides to move the object, the event on drag object is called to allow or reject the request by returning a different value of 0. It is always implicitly allowed.
Target	The target is an object or label that can receive source objects. When a source object is deposited on a target object, the event on use object is called to notify the validation of the drag and drop. The first parameter identifies the source object. For a label, it is the event on use label that is called.
Both	In this case, the object may be the source or target.

When a drag and drop does not reach any target, the event on drop object is called.

Keep in mind that by activating drag and drop on your objects and labels, interactions with inventory will no longer be available.

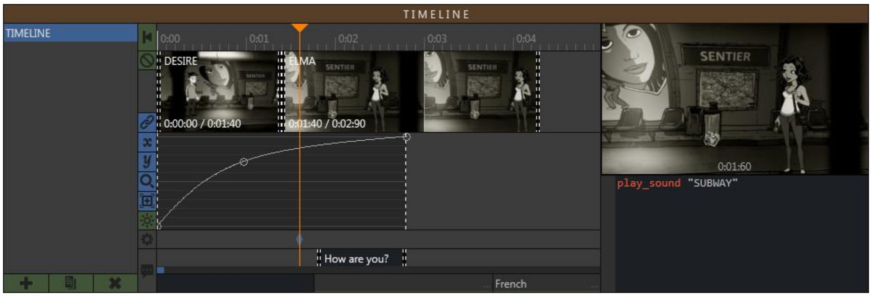
THE EVENTS OF THE GLISSER-DEPOSER

In the role editor, some events are specific to drag and drop that we list briefly below:

on drag object	Allows you to receive a request to drag and drop and to be able to refuse if necessary.
on drop object	Allows you to receive a notification when an object has been deposited on no declared target.
on use label	Allows you to receive a notification when an object is deposited on a label declared as a target.
we use object	Allows you to receive a notification when an object is deposited on another object declared as a target.

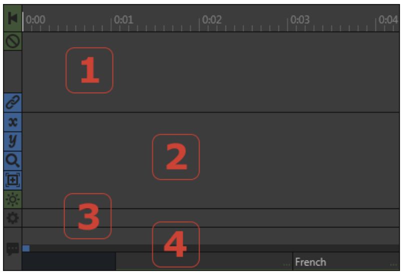
Timeline

Timelines have two main uses: creating in-game cutcenes and transitions by taking advantage of shots. One of the practical advantages is to be able to attach music and add subtitles while respecting the synchronization of all elements present on the timeline, including script execution.



```
play_sound "SUBWAY"
```

The first column on the left allows you to add and manage the timelines. A scene can contain as many timelines as you want. The right part allows you to preview the rendering in real time during editing and also to write your scripts. Finally, the central part allows you to manage the timeline elements.

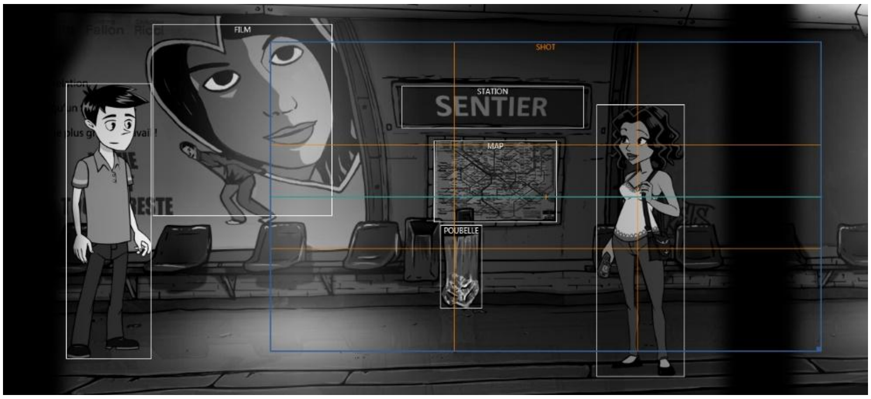


1 Shots 2 Transitions 3 Scripts 4 Subtitles

To play a timeline during the game, use the timeline box and to stop prematurely use the stop box.

SHOTS

You must first add and correctly place your shots in the scene as for a label. However, you will not be able to directly change the height that is determined by two values: the width of the shot and the ratio of the resolution of the game.

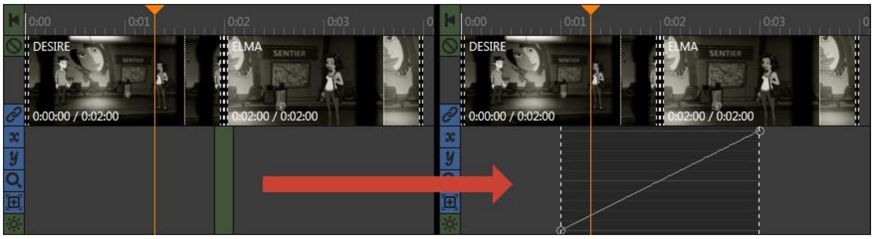


When a shot is selected, a geometric rule (called a third-party rule) appears inside the frame to help you compose your image. Without entering into details, the rule is to place the subject or a key element either at an intersection or on a line. Insert as many shots as necessary but it is important to name them with an explicit short name if possible. By analogy with a video editing software, it is for the moment to import your rushes into your project.

You must then click on an empty space in the first row of the timeline to open a dialog box with the list of available shots and add them to your sequence.

TRANSITIONS

When two shots are side by side, a green bar appears below the connector to allow you to apply a linear transition by clicking on it. Of course, you can adjust the duration of the transition by stretching the frame through its ends.

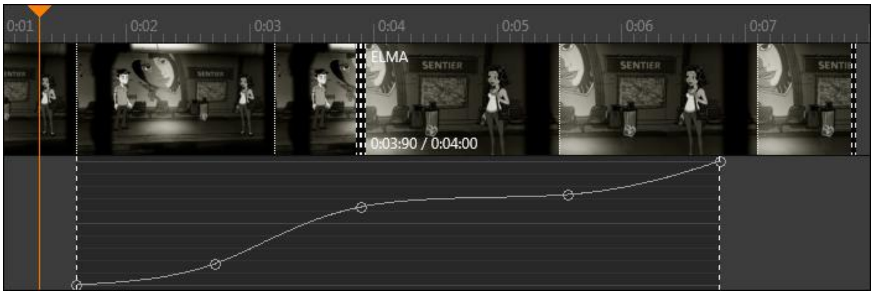


Once the transition is applied, it is no longer possible to detach the two shots without explicitly removing it from the popup menu.

SPLINE INTERPOLATION

Note that it is very easy to act independently on the X and Y position of the camera, as well as on zoom, blur, depth of field, focus and black melt thanks to the buttons on the left edge. The first button symbolized by a chain allows you to control a group of parameters.

By default the interpolation is linear using two opposite points that can be repositioned on the axis of the ordered. To benefit from Spline interpolation in order to create smoother accelerations and decelerations, you need to add intermediate points. To do so, just click with the left mouse button on the curve at the desired location. A simple right click to remove them.



The number of intermediate points is limited to eight, i.e. a total of ten points per transition, including the ends.

MUSIC AND SUBTITERS

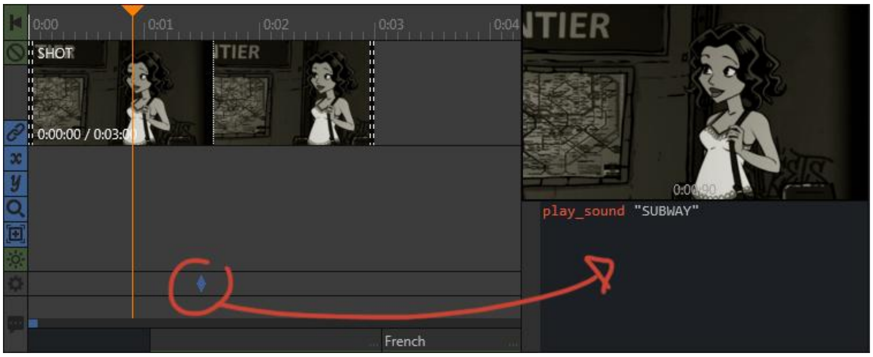
To synchronize a music, instead of going through scripts, add an OGG file to the SONG folder of the project and then select it for each language directly from the drop-down list at the bottom right of the timeline.

The subtitles of the timeline work the same way as to subtitle a movie. Move the cursor to the place where you want to start the subtitle and enter the text in the field below. As with shots and transitions, you can stretch their frame to adjust the duration. Subtitles are not previewable in the editor.

Unlike kinematics, subtitles are more easily translated as they are integrated into CSV files and enter the production loop.

SCRIPTS

Another advantage of the timelines is to be able to insert a script at any time of the sequence, in particular to launch actions, play animations or sounds in a scripted way.



```
play_sound "SUBWAY"
```

Click where you want to place a new script on the timeline and add code in the text field below the preview.

Code integration

First you have to open a scene via the jump or jump_back function. To jump the first parameter is the name of the scene and the second parameter is the transition time in milliseconds. The jump_back function allows you to reopen the previous scene without specifying its name.

```
jump S_GARDEN 500
```

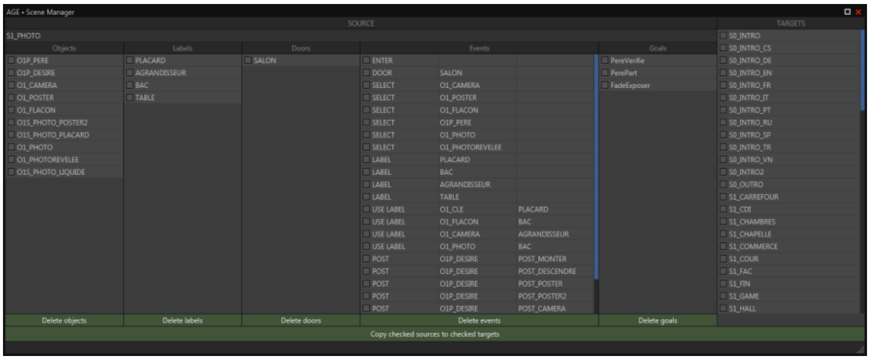
To identify the current or previous scene, you can use the stage and old_scene functions.

```
if scene S_GARDEN S_HOUSE end
```

In the above code, the condition is true if the current scene is either S_GARDEN or S_HOUSE.

Scene Manager

The scene manager is accessible from the main menu of the software. The purpose of this manager is to be able to duplicate the elements of a scene to other scenes in just a few clicks. You can also delete elements from the selected scene. The elements are stills, objects, labels, shots, walls and timelines.



Objects	Labels	Doors	Events			Goals	TARGETS
O1P_PERE	PLACARD	SALON	ENTER			PereVerifie	S0_INTRO
O1P_DESIRE	AGRANDISSEUR		DOOR	SALON		PerePart	S0_INTRO_CS
O1_CAMERA	BAC		SELECT	O1_CAMERA		FadeExposer	S0_INTRO_DE
O1_POSTER	TABLE		SELECT	O1_POSTER			S0_INTRO_EN
O1_FLACON			SELECT	O1_FLACON			S0_INTRO_FR
O1S_PHOTO_POSTER2			SELECT	O1P_PERE			S0_INTRO_IT
O1S_PHOTO_PLACARD			SELECT	O1_PHOTO			S0_INTRO_PT
O1_PHOTO			SELECT	O1_PHOTOREVELEE			S0_INTRO_RU
O1_PHOTOREVELEE			LABEL	PLACARD			S0_INTRO_SP
O1S_PHOTO_LIQUIDE			LABEL	BAC			S0_INTRO_TR
			LABEL	AGRANDISSEUR			S0_INTRO_VN
			LABEL	TABLE			S0_INTRO2
			USE LABEL	O1_CLE	PLACARD		S0_OUTRO
			USE LABEL	O1_FLACON	BAC		S1_CARREFOUR
			USE LABEL	O1_CAMERA	AGRANDISSEUR		S1_CDI
			USE LABEL	O1_PHOTO	BAC		S1_CHAMBRES
			POST	O1P_DESIRE	POST_MONTER		S1_CHAPELLE
			POST	O1P_DESIRE	POST_DESCENDRE		S1_COMMERCE
			POST	O1P_DESIRE	POST_POSTER		S1_COUR
			POST	O1P_DESIRE	POST_POSTER2		S1_FAC
			POST	O1P_DESIRE	POST_CAMERA		S1_FIN
							S1_GAME
							S1_HALL

To duplicate information, first select the scene and check the desired items. Then in the right column, check the scenes to edit and click the button to launch the copy. If an item already exists in a scene, it will be replaced except for the stills. For the latter, new items will be created.

Good practice

Here are some practical tips to optimize the performance of the game:

- Disable Hi-quality from your layers if you do not need to keep the high fidelity of the original file (e.g. for lighting layers and silhouette primers).
- Merge the layers upstream with your favorite graphic editing software.
- Limit the number of effects to be applied to the scene, especially for mobile devices. You can also use texture adjustments applied to the

application generation as an alternative method to find the best compromise between quality and performance.

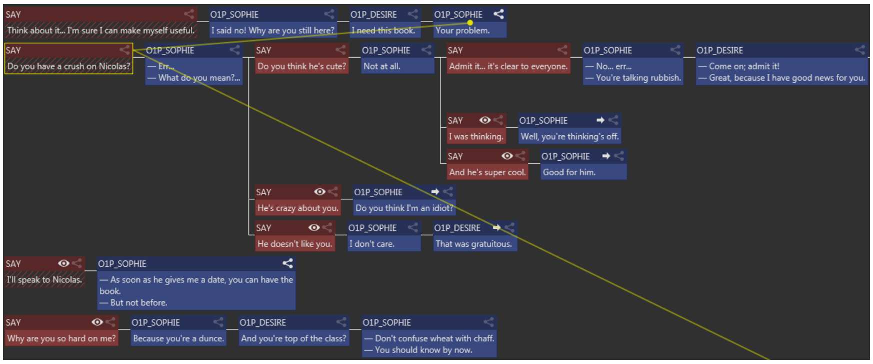
- Limit static objects of large size by merging them with the decor and layers if possible, or use the stills to benefit from the depth. The more objects to display in the scene and the slower the game will slow. I remind you that the stills are grouped in a common texture.
- Limit the number of images of animations or use Streaming mode in the object editor.
- Limit the number of lights and activate the Baked lights option if applicable.
- Enable 8-bit PNG compression with PngQuant from project parameters. Unless you use photographs, the difference will hardly be noticeable.

DIALOGUE

The dialog editor allows you to write and structure your interactive conversations. For a narrative adventure game, you will often have to work in parallel on the stage editor, role editor and dialogue to trigger state changes. You can stand down on this editor if your gameplay is essentially based on puzzles of objects without narrative text, as was the case for my game Vive le Roi.

For practical reasons, during a conversation, the player does not have the possibility to save the game. He must first leave the conversation when a list of choices is offered to him.

Quick access to the current language allows you to translate or check sentences without exporting CSV files.



SAY — Think about it... I'm sure I can make myself useful.
O1P_SOPHIE — I said no! Why are you still here?
O1P_DESIRE — I need this book.
O1P_SOPHIE — Your problem.

SAY — Do you have a crush on Nicolas?
O1P_SOPHIE — — Err... — What do you mean?...
SAY — Do you think he's cute?
O1P_SOPHIE — Not at all.
SAY — Admit it... it's clear to everyone.
O1P_SOPHIE — — No... err... — You're talking rubbish.
O1P_DESIRE — — Come on; admit it! — Great, because I have good news for you.

SAY — I was thinking.
O1P_SOPHIE — Well, you're thinking's off.

SAY — And he's super cool.
O1P_SOPHIE — Good for him.

SAY — He's crazy about you.
O1P_SOPHIE — Do you think I'm an idiot?

SAY — He doesn't like you.
O1P_SOPHIE — I don't care.
O1P_DESIRE — That was gratuitous.

SAY — I'll speak to Nicolas.
O1P_SOPHIE — — As soon as he gives me a date, you can have the book. — But not before.

SAY — Why are you so hard on me?
O1P_SOPHIE — Because you're a dunce.
O1P_DESIRE — And you're top of the class?
O1P_SOPHIE — — Don't confuse wheat with chaff. — You should know by now.

Dialogue

A dialogue consists of a choice tree in blue and red replicas.

DIALOGUE PROPERTIES

Dialog	
Tags	You can set up up to four tags per dialog. They allow you to identify dialogs other than by name in order to group them by category. A fifth tag is editable by the set_asset_tag function.

DIALOGUE ACTIONS

In the role editor, some synchronised actions are specific to the dialogues we list briefly below:

stop	Allows you to finish the current conversation.
talk	Allows you to start a conversation by having the ability to define the beginning and end phrases.

THE CONDITIONS OF THE DIALOGUE

In the role editor, certain conditions are specific to the dialogues we list briefly below:

dialog is	Lets you know the dialogue in progress.

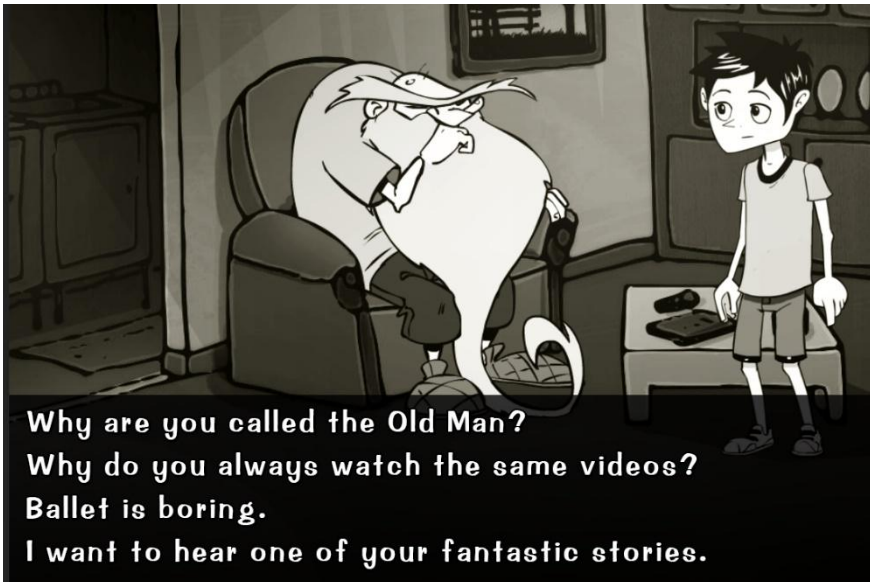
THE EVENTS OF THE DIALOGUE

In the role editor, some events are specific to the dialogues we list briefly below:

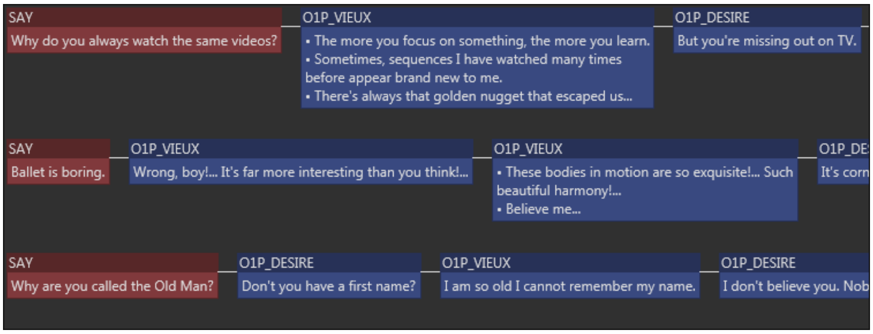
we enter dialog	Allows you to receive a notification when the dialog starts.
we exit dialog	Allows you to receive a notification when the dialogue ends.

"Choice" red box

The red boxes allow you to offer the player a choice of replicas at different times in the conversation. These replicas are attributed to the current player.



Why are you called the Old Man?
Why do you always watch the same videos?
Ballet is boring.
I want to hear one of your fantastic stories.



SAY
Why do you always watch the same videos?
O1P_VIEUX
• The more you focus on something, the more you learn.
• Sometimes, sequences I have watched many times before appear brand new to me.
• There's always that golden nugget that escaped us...
O1P_DESIRE
But you're missing out on TV.

SAY
Ballet is boring.
O1P_VIEUX
Wrong, boy!... It's far more interesting than you think!...
O1P_VIEUX
• These bodies in motion are so exquisite!... Such beautiful harmony!...
• Believe me...
O1P_DE
It's corn

SAY
Why are you called the Old Man?
O1P_DESIRE
Don't you have a first name?
O1P_VIEUX
I am so old I cannot remember my name.
O1P_DESIRE
I don't believe you. Nob

PROPERTIES OF THE CHOICE BOX

Choice	
ID	Unique identifier of the box used by the functions.
Tags	You can set up up to four tags per choice. They allow you to group them by category. A fifth tag is editable by the set_sentence_tag function.
Visible	Allows you to change the visibility of the choice when launching a new game.
Do not say	By default the choices are pronounced by the talking object.

Never hides	By default, the choices are withdrawn after selection.
Locked	By default the player can leave a conversation at the time of a choice. By activating this option, the player will be forced to choose a reply before being able to interrupt the conversation at the next choice.
Randomly	When the box contains several separate choices of an empty line, the option allows you to choose one randomly.
Mutual show	Makes the boxes indicated by their identifier visible after the replica. The link is then represented in the editor by a line when one of the targeted boxes is selected. The identifiers are separated by a space.
Hide branch	Allows you to hide the entire branch after the replica.
Go to	Allows to reach another box after the replica.
Exit	Allows you to leave the conversation after the reply.
Content	
Icon	Allows to display the current icon of an object instead of text.
Object	
Directorate	Allows you to change the direction of the talking object during the replica.
Look at	Allows you to direct the talking object to another object during the replica.
Keep new direction	Allows you to keep the new direction after the replica.
Animation	Allows to change the animation of the talking object during the replica.
Keep new animation	Allows you to keep the new animation after the replica.

THE ACTIONS OF THE CHOICE BOX

In the role editor, some synchronised actions are specific to the choices we list briefly below:

show	Allows you to change the visibility of a choice instead of going through the functions show_choice and hide_choice.

THE CONDITIONS OF THE CHOICE BOX

In the role editor, certain conditions are specific to the choices we list briefly below:

dialog has	Lets you know if the choice is visible.

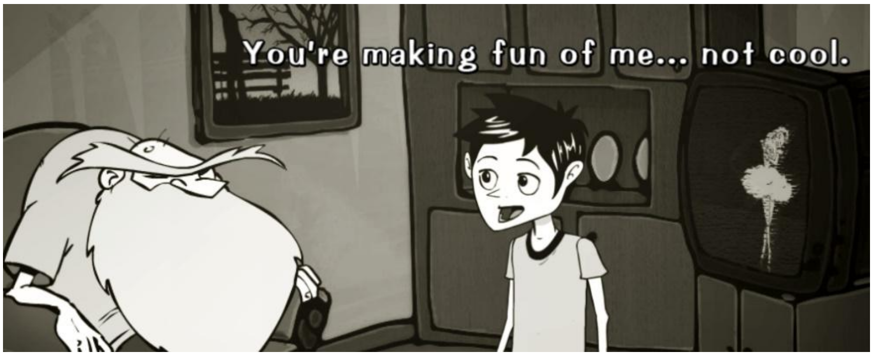
EVENTS OF THE CHOICE BOX

In the role editor, some events are specific to the choices we list briefly below:

We're gonna say it.	Allows you to receive a notification when the choice is displayed. The Said property allows you to distinguish the first display. The Sub-choice property is ignored for the choices.
on unlock choice	Allows you to receive a notification when the choice is released for the first time.

Blue box "Sentence"

The blue boxes allow to give the replica to a character present in the scene.



BOX PROPERTIES SENTENCE

Sentence	
ID	Unique identifier of the box used by the functions.
Tags	You can set up up to four tags per replica. They allow you to group them by category. A fifth tag is editable by the set_sentence_tag function.
Entry point	Allows you to launch the replica if all the choices at the root are exhausted at the start of the conversation. If there are several entry points then only one replica will be drawn by lot.
Duration	Allows to force the duration of the replica for reasons of synchronization with audio or for an action for example. The property is ignored if the value is 0.
Randomly	When the box contains several replicas separated from an empty line, the option allows you to choose one randomly.
Mutual show	Allows the boxes indicated by their identifier to be visible after the replica. The link is then represented in the editor by a line when one of the targeted boxes is selected. The identifiers are separated by a space.
Hide branch	Allows you to hide the entire branch after the replica.
Go to	Allows to reach another box after the replica.
Exit	Allows you to leave the conversation after the reply.
Object	
Speaker	The talking object. VOICE-OVER allows to display a dialog in the form of a subtitle without the need to add an object to the scene. The set_voice_over_color function will be used to customize the text color before or during the conversation.
Directorate	Allows you to change the direction of the talking object during the replica.
Look at	Allows you to direct the talking object to another object during the replica.
Keep new direction	Allows you to keep the new direction after the replica.

Animation	Allows to change the animation of the talking object during the replica.
Keep new animation	Allows you to keep the new animation after the replica.
Puzzle	
Success on Say	Allows to solve the specified puzzle after launching the replica for the first time.

THE CONDITIONS OF THE SENTENCE BOX

In the role editor, certain conditions are specific to the choices we list briefly below:

dialog has	Lets you know if the replica has already been displayed.

THE EVENTS OF THE SENTENCE BOX

In the role editor, some events are specific to the choices we list briefly below:

We're gonna say it.	Allows you to receive a notification when the reply is displayed. The Said property allows you to distinguish the first display from the Sub-choice property to compare the index to the paragraph chosen by the player.

Interface

You can move a box with or without its children by selecting it and sliding it to another box. Control the location with CTRL and SHIFT keys.

branch [before]	branch [inside]	box [before]	box [inside]
	CTRL	SHIFT	CTRL + SHIFT

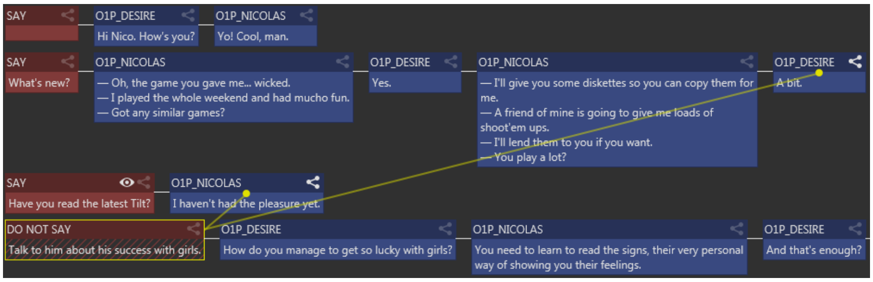
The default behavior positions the selected branch before another branch. Other behaviors are accessible by using the keyboard. Before releasing the left mouse button when moving a branch or box, press:

- the CTRL key to move the selected branch within another branch.
- the SHIFT key to position the selected box before another box.
- CTRL and SHIFT keys to move the selected box inside another branch.

The same goes for copying to the clipboard via the popup menu. The data in JSON format will be accessible by all Dialogue Assets, including by the other instances of the software. Note that the identifiers will be modified.

BRANCH	BOX	BRANCH	BOX
CTRL + C	CTRL + CHIFT + C	CTRL + X	CTRL + CHIFT + X

The Mutual show property has a very convenient shortcut to quickly unlock choices after the display of one or more replicas. First select the choice to make appear and click on the icon at the top right of the box that will unlock the choice. At the time of the click, the Visible of choice property (selected box) will automatically pass to False.


SAY

O1P_DESIRE
Hi Nico. How's you?

O1P_NICOLAS
Yo! Cool, man.

SAY
What's new?

O1P_NICOLAS
— Oh, the game you gave me... wicked.
— I played the whole weekend and had mucho fun.
— Got any similar games?

O1P_DESIRE
Yes.

O1P_NICOLAS
— I'll give you some diskettes so you can copy them for me.
— A friend of mine is going to give me loads of shoot'em ups.
— I'll lend them to you if you want.
— You play a lot?

O1P_DESIRE
A bit.

SAY
Have you read the latest Tilt?

O1P_NICOLAS
I haven't had the pleasure yet.

DO NOT SAY
Talk to him about his success with girls.

O1P_DESIRE
How do you manage to get so lucky with girls?

O1P_NICOLAS
You need to learn to read the signs, their very personal way of showing you their feelings.

O1P_DESIRE
And that's enough?

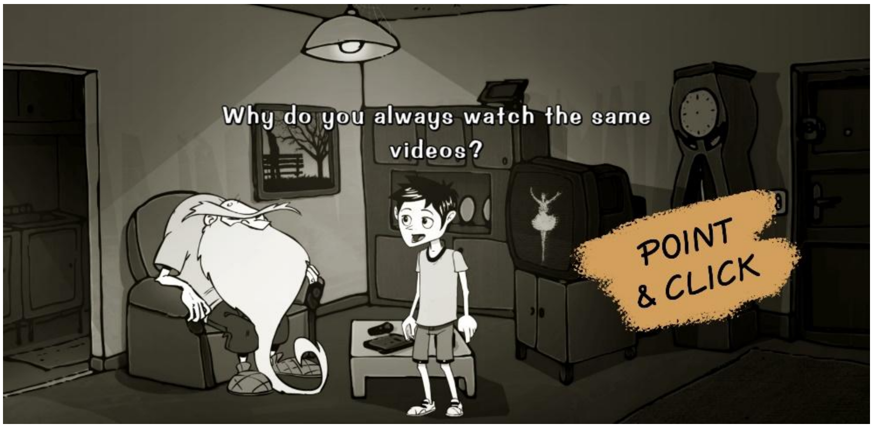
On the runtime side, all boxes must be read so that the choice will appear on the screen. So you don't have any line of code to write to offer new choices to the player as the conversation progresses if you do so.

Style

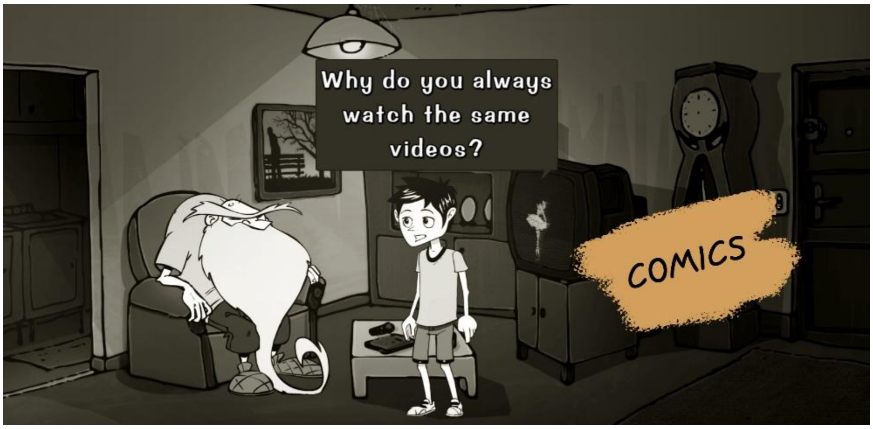
seccia.dev introduces four formatting models that will help you to better define the style of your dialogues. Again, you have to distinguish between the replicas of the characters and the choices offered to the player. Replies are useful to advance the story, the choices to strengthen the gameplay.

THE RESPONSE

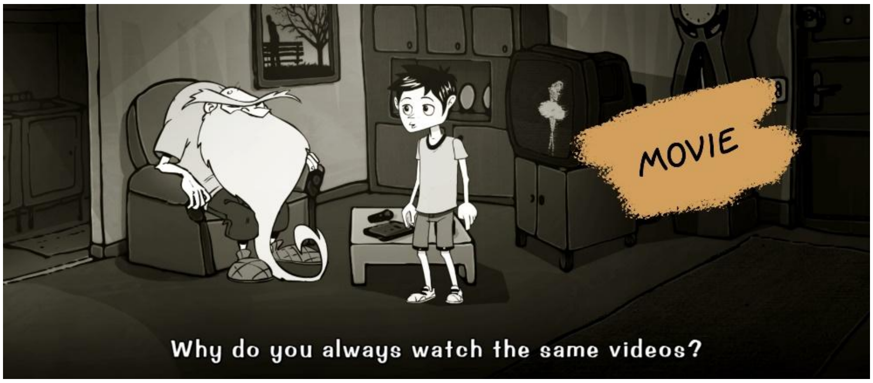
With respect to replicas, four styles are available: POINT & CLICK, COMICS, MOVIE and RPG. Whatever the style used, it is possible to apply Typewriter and Vibration effects to text from the project properties. These two effects affect the whole game.



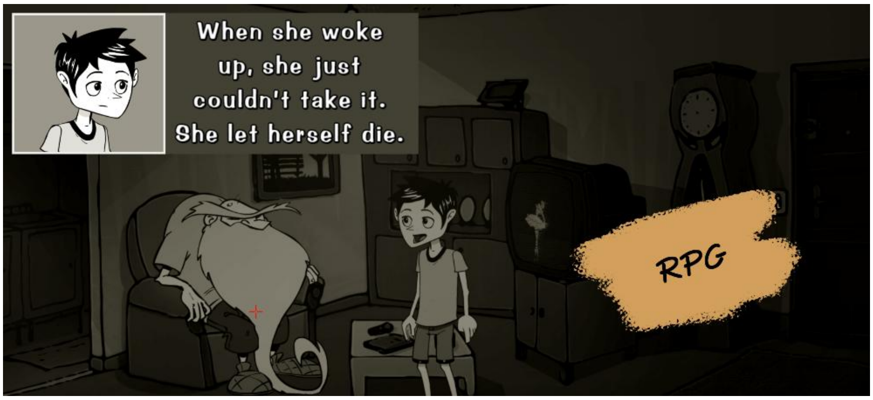
The POINT & CLICK style proposed by seccia.dev is inspired by LucasArts games where the replicas are always placed above the character who is speaking. The position and alignment of the text are automatically managed by the software. This is the default style. It is important to assign an animation TALK to the character to attract the player's attention. To better distinguish the spoken words, you can also assign one color per character or maybe two or three colors at the most to avoid the Christmas tree effect. A color for the hero, a second color for the other protagonists and possibly a third color for the secondary roles.



The COMICS style designs a customizable bubble for each replica. You need to provide a BALLOON image of 128x128 pixels with or without transparency to be placed in the IMAGE folder. The bubble size will adjust relative to the text. To do this you need to specify additional coordinates in the project properties. seccia.dev offers a free template with its predefined margins to be recovered in the Resources folder of the software.



The MOVIE style positions the text at the bottom of the screen as movie subtitles. Just activate the Movie style property of the character in the object editor. So you can combine the two styles in a single discussion depending on the active character.

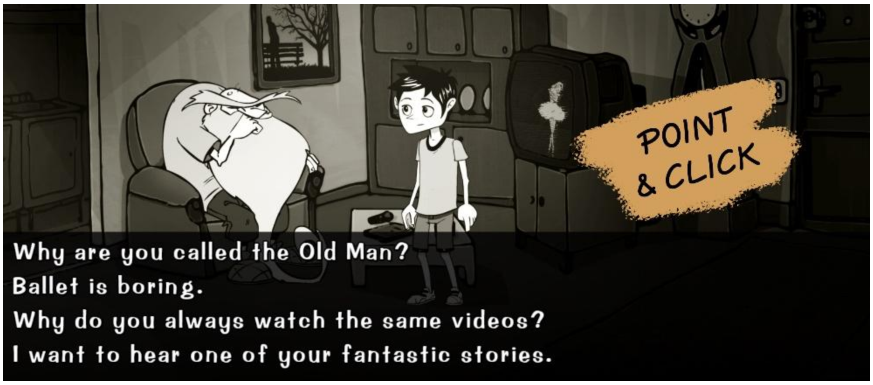

When she woke up, she just couldn't take it. She let herself die.
RPG

Finally, the RPG style adds an avatar during the conversation. The position of the text will depend on the text layout property of the avatar. To set up this style, the necessary prerequisite is to create a new object containing STOP animation and possibly a TALK animation, then to be able to select this object in the Avatar property of the character via the object editor.

TALK animation is played when the character retains the speech and STOP animation when the player is prompted to choose an answer. STOP animation is used instead of TALK animation if the latter is absent. If STOP animation does not contain images, no avatar will be displayed.

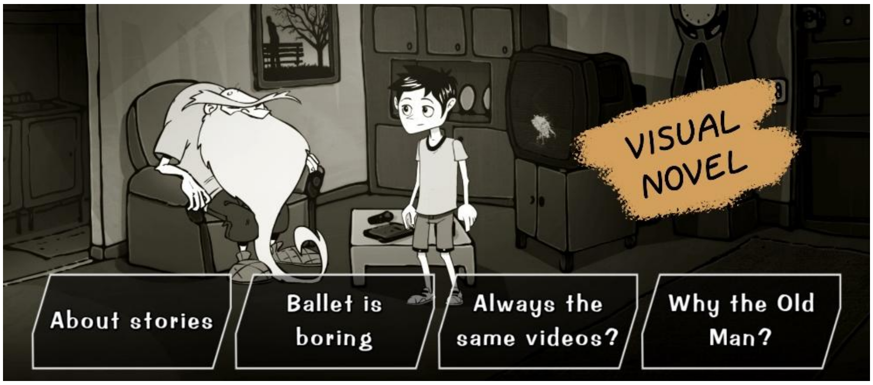
CHOICE

With regard to the choice of replicas, two styles are available: the POINT & CLICK style and the VISUAL NOVEL style. Whatever the style used, it is possible to apply the Vibration effect to the text from the properties of the project.



Why are you called the Old Man?
Ballet is boring.
Why do you always watch the same videos?
I want to hear one of your fantastic stories.

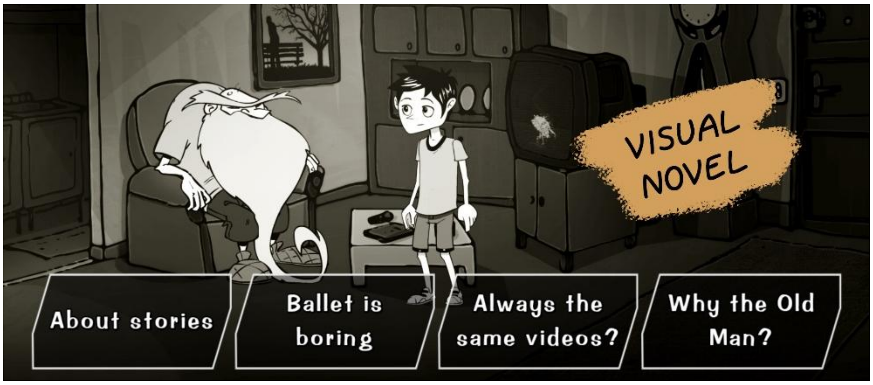
The POINT & CLICK style displays the choices per line at the bottom of the screen. It is the default style and takes over the traditional Point & Click game codes. If the sentence is too long, it will scroll horizontally to the mouse cursor. This mode is ideal to develop its purpose.



The VISUAL NOVEL style presents the choices in the form of customizable boxes. This more graphic mode is better suited to offer short choices to the player as topics for discussion in a few words.

The first step is to draw the graphic box in a single PNG file. The size of the image must be 256x256 pixels with or without transparency. If your frame is rectangular, just leave an empty space above it without resizing the file. The images are centred horizontally and arranged at the bottom of the screen. The file must be named CHOICE and placed in the IMAGE folder.

The second step is to set the style via the Project Choice properties in the Text group.



Bubbles can contain an icon instead of text by specifying the name of the object in the Icon property of the choice.

Code integration

To start a conversation, just add a talk box by specifying the Dialogue Assembly and possibly identifying the start replica and the end replica. Adding a stop box or calling the shutup function will prematurely interrupt the conversation

```
shutup
```

Three other functions that will be very useful to you to control the visibility of choices.

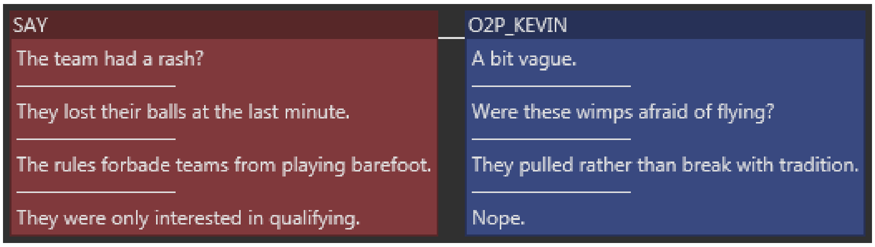
```
show_choice D_BOB 5 hide_choice D_BOB 8 if choice D_BOB 8
end
```

The said function allows to know whether a replica has already been said as its name indicates.

```
if said D_BOB 8 end
```

If a red box contains multiple replicas separated by an empty line and you want to know the player's choice, you can use the $_choice variable in the box script on say this way:

```
if var _choice 0 // first choice else_if var _choice 1 //
second choice end
```



SAY

The team had a rash?

They lost their balls at the last minute.

The rules forbade teams from playing barefoot.

They were only interested in qualifying.

O2P_KEVIN

A bit vague.

Were these wimps afraid of flying?

They pulled rather than break with tradition.

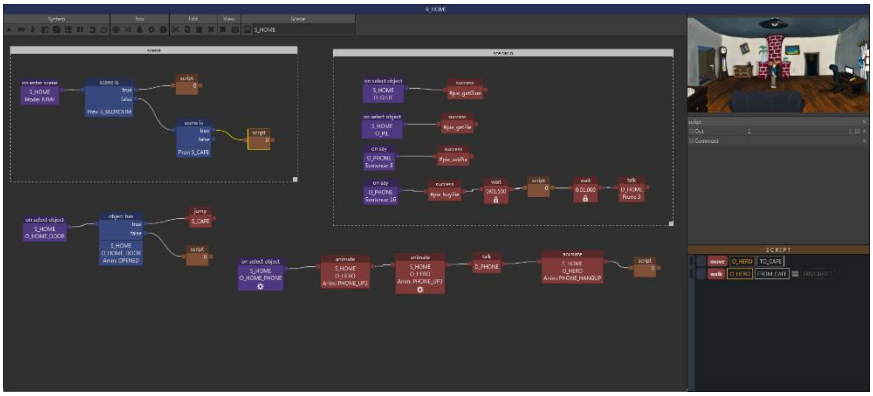
Nope.

This variable will be accessible in the event we say of the replica.

ROLE

Role editor allows you to visually program the synchronised actions, conditions and events of your game in a more intuitive way than by code lines.

It is possible and advisable to associate a role with a scene to interact between their respective editors. Moreover, at runtime the associated role will be automatically started at the loading of the scene, and stopped at its closing.



Language

The role editor offers five types of box: green system boxes, red action boxes, blue condition boxes, purple event boxes and brown scripts.

Connections have the same properties as in the scenario editor. If a box accepts multiple input connections, all signals must be sent to trigger the box execution.

START

The start box is called to launch the role. It is possible to set the call order of the start boxes by changing the value of the execution order parameter. The boxes will be sorted in ascending order.

UPDATE

The update box is called once per frame of the application (about 60 times per second for a 60 fps rotating application). You cannot control the refresh frequency. Please use either the on repeat box to create time-based intervals or the wait box to insert a pause in milliseconds. It is possible to set the call order of the update boxes by changing the value of the execution order parameter. The boxes will be sorted in ascending order. At the start of a role, they will always be called after the start boxes.

TASK

The task box is called by the task function and allows you to bridge between scripts and roles. You can use it as routines to reuse code without duplicating it.

OR

The gold box allows to set up a logical condition. If an input has several connections, the first signal received will suffice to run the box. The output has no particularity.

RANDOM

The random box allows to emit a single signal at random among the output connections. The input has no particularity.

SCHEDULE

The Schedule box allows you to plan the order of the outputs. The number of outputs is editable and can not be less than 2. Nothing prevents you from connecting multiple cables on the same output to combine the two methods. It is not mandatory to use all outputs.

SKIP

The skip box allows you to skip a frame from the program. It works as a pause of a single frame duration. Sometimes it can be useful to perform actions at the next frame.

RESTART

The restaurant box allows to restart the role by reset all states.

END

The end box makes it possible to stop the role.

SCRIPT

The script box allows you to run code and choose an output using the output function. By default, the first output is taken into account. The output function allows you to specify multiple outputs if needed. The box can return a different value of 0 to maintain execution and act as a loop. Thus, the script will be restarted to the next frame without transmitting signals.

Actions

In the role editor, the execution of an action is synchronized in contrast to scripts. This means that the signal comes out of the box when the action is considered finished. The notion of "action completed" depends on the action and its parameters.

An action can only have one input and only one output. It has a script that allows you to call other functions by code. This script is executed in three circumstances: when the signal enters the box (enter), when the signal persists on several frames (execute) and when the signal exits (exit). It is the same script and in order to differentiate these three possible cases, you must use the enter, execute and exit functions directly in the code.

```
if enter end if execute end if exit end
```

It must be understood that the script is only called once per frame regardless of the situation. If the three functions return true, this means that the signal enters and exits immediately.

It is possible to leave an action prematurely by returning a different value of 0. If the signal is already supposed to come out of the box, the returned value will not have any impact.

```
Return 1
```

Actions have a common parameter: Lock game. This parameter allows you to block interactions with the player as long as the signal is not out of the box. The lock function is independent and can be combined.

ANIMATE

The animate box allows you to play animation of an object.

Scene	The scene where the object is to be animated. If the field is empty, the object must exist in the current scene.
Object	The object to be animated.
Animation	The name of the animation to play.
Directorate	The name of the direction. If the direction is not specified and it exists for this animation, the current direction will be retained.
Reverse	Allows you to play the animation backwards. By default, the value set in the object editor will be taken into account.
Start frame	The index of the frame at which the animation should start. By default, the value defined in the object editor will be taken into account.
Action frame	The index of the frame to which a notification will be sent. By default, the value defined in the object editor will be taken into account.

CAMERA

The camera box allows to move the camera from one position to another without using the shots and to apply a zoom to the transition if necessary.

Scene	The name of the camera scene. If the field is empty, the indicated elements must exist in the current scene.
From shot	The name of the shot referring to the starting position.
From object	The name of the object referring to the starting position.
From cell	The name of the cell of the object referring to the starting position.
From zoom	The starting zoom value between 100 and 400.
To shot	The name of the shot referring to the end position.

To object	The name of the object referring to the end position.
To cell	The name of the cell of the object referring to the end position.
To zoom	The value of the end zoom between 100 and 400.
To offset X/Y	Allows a horizontal and/or vertical offset in pixels to be applied to the end position that will be maintained after the transition.
Axis	Allows to include or exclude axes. By default, both axes are taken into account.
Fade out	Allows to apply an output melt at the beginning of the transition.
Fade in	Allows to apply an input melt at the end of the transition.
Duration	The duration in milliseconds of the transition.
Endless	Allows to maintain the state of the camera with the end-of-transition values.

CINEMATIC

The cinematic box allows you to launch a video via an asset cinematic. The signal leaves the box when the video is stopped.

Cinematic	Name of the cinematic assembly to play.

EXAMINATION

The box examines allows a large display of an object to allow the player to observe an important detail of the game. During the observation, the game remains blocked until the player leaves the screen. The signal leaves the box when the player regains his hand.

Object	Name of object to be examined.
Animation	Name of the animation of the object to be played in loop.
Directorate	Name of the animation director.

Scale	The size of the object as a percentage of the screen size.
Opacity	The opacity of the object in percentage. The value 0 makes the object invisible.

INTERPOLATE

The interpolate box allows to interpolate an integer between a minimum value and a maximum value, and to recover the result in a variable.

Duration	The duration of interpolation in milliseconds.
Min/Max	The minimum value and the maximum value of interpolation.
Loop count	The number of loops to be performed.
Loop pause	The break in milliseconds between two loops.
Reverse	Allows to reverse interpolation with each loop.
Ease IN/OUT	Add an acceleration at the beginning and/or deceleration at the end.
Variable	The name of the variable that will contain the result of interpolation at each frame.

JUMP

The jump box allows you to change scene by applying a black melt transition. The signal leaves the box when the screen turns black.

Scene	The name of the new scene to load. Select BACK to reload the previous scene if it exists (equivalent to the jump_back function).
Fade output	Duration in milliseconds of the transition of the output melt.
Break	Duration in milliseconds of the break before the change of scene.
Fade indu ration	Duration in milliseconds of the inlet melt transition.

(b) (c) (d) (e) (e) (e) (e) (e) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f) (f)

The light box allows you to change the properties of a light attached to an object.

Scene	The name of the light scene. If the field is empty, the object must exist in the current scene.
Object	Name of the object where the light is located.
Visible	Change the visibility of the light to turn it on or off.
Ambient	Change the ambient value of the light between 0 and 255. This value adds to the current ambient value of the scene. To achieve an effect, it is therefore necessary to reduce the ambient value either of the scene or of the game.
Diffuse	Change the diffuse color of the light to RGBA format. To ignore this field, use the value 0 for the alpha component.
Angle	Change the angle amplitude between 0 and 360 degrees. If the angle is 360, the light is omnidirectional.
Directorate	Change the direction of light by specifying an angle between 0 and 359 degrees.
Distance	Change the distance to pixels of light. If the value is -1, the light illuminates infinity.
Detention	Change the light attenuation method. The value is a power between 0 and 10. If the value is 0, no attenuation is applied. If the value is 1, a linear attenuation is applied. If the value is 2, a square attenuation is applied.

PATH

The path box allows you to start the path of an object.

Scene	The name of the scene of the object. If the field is empty, the object must exist in the current scene.

Object	The name of the object where the path is located.
Path	The index of the path to be started.
Spot	The name of the spot where the displacement should start. If the field is empty, the first spot is used.
Loop count	The number of loops to perform. If the field is empty, the value defined in the path properties is taken into account.

PLAYER

The player box allows to modify the attributes of a player.

Player	The name of the player to change.
Scene	The scene name used for character change (equivalent to set_player_scene).
Object	The name of the object controlled by the player. (equivalent to the control function).
Switch	Allows you to change the current player if the value is true (equivalent to the switch function).
Empty	Allows to remove all objects from the inventory (equivalent to the empty function).
Transfer to	Allows to move all objects from the inventory to another player (equivalent to the transfer function).

POPUP

The popup box allows to display a message on the screen with a choice of response. Depending on the type of the window, several outputs are possible.

Title	The title of the window.
Message	The message from the window.
Type	By default, the window displays a OK button. It is possible to display two Yes/No buttons by choosing Question.

ROLE

The role box allows you to start a role. The signal will be blocked as long as the role is being played.

Role	Name of the role to be started.

SELECT

The select box allows you to reproduce the click on an object.

Player	The name of the player that must match the current player.
Scene	The name of the scene that must correspond to the current scene.
Object	The name of the object to be selected.
Subject	The name of the sub-object to be selected.

SHOW

The show box allows to change the visibility of one or more entities. The signal comes out immediately after changing the state of the elements.

Visible	Makes the specified entities visible or invisible.
Duration	Duration of the transition of the change of status if applicable.
Brightness	Allows to change the brightness of the scene between 0 (black) and 100.
Scene	The name of the scene concerning the specified entities. If the field is empty, the entities must exist in the current scene.
Layer	The name of the layer whose visibility will be changed.
Still	The name of the still whose visibility will be changed.
Object	The name of the object whose visibility will be changed.

Light	The name of the light via the object whose visibility will be changed.
Label	The name of the label whose visibility will be changed.
Cursor	Allows to change the visibility of the cursor. This property does not take into account the Visible property.
Bag	Allows to change the visibility of the inventory.
Menu	Allows to open or close the game menu. This property does not take into account the Visible property.
Credits	Allows to display credits or return to the game. This property does not take into account the Visible property.
Dialog	The name of the dialogue where the choice is to be changed.
Choice	The identification of the choice whose visibility will be changed.

SONG

The song box allows you to play music on a channel. The signal leaves the box immediately even in the event of a transition.

Song	The name of the music file without extension.
Channel	Channel index between 0 and 3. The -1 value acts on all channels, if applicable.
Volume	The volume of music between 0 and 100.
Loop	Single or loop reading.
Crossfade	The length in milliseconds of the transition between the music being played and the new music. Both musics must use the same channel.
Last song	Allows you to replay the previous music.

STOP

The stop box allows to stop the specified entities.

Object	The name of the object to stop (equivalent to the stop function).
Role	The name of the role to be interrupted (equivalent to the stop_role function).
Camera	Allows you to reset camera movements.
Dialog	Allows you to interrupt the conversion in progress.
Timeline	Allows you to interrupt the timeline.
Cinematic	Allows you to interrupt the kinematics while you are playing.
Song	Allows to stop the music while playing on a channel. If the channel is -1, all the music will be interrupted.

SUCCESS

The success box allows you to validate a riddle. This is the equivalent of the success function by the code.

Puzzle	The name of the riddle to be validated.

TAKE

The take box allows you to retrieve an object to place it in an inventory or in the basket.

Player	The name of the player who will receive the object. If the parameter is not specified, the current player will be used.
Subject matter	The name of the object to be recovered.
Silent	Allows to enable or disable visual effect.
Replace	The name of the object to be replaced if it is not an addition.
Animation	The name of the player's animation to play before triggering the action.
Directorate	The name of the animation director.

Action frame	The index of the animation frame used to trigger the action. -1 corresponds to the end of the animation. If the parameter is not specified, the index of the object editor will be used.

TALK

The talk box allows you to start a new conversation. The signal leaves the box when the conversation ends.

Dialog	The name of the dialogue to start.
Sentence (start)	Identifier of the reply of the beginning of the conversation.
Sentence (end)	Identifier of the reply that will interrupt the conversation. It is possible that the player leaves the conversation before reaching that reply.

TIMELINE

The timeline box allows you to launch a timeline.

Scene	The timeline scene name. If the field is empty, the timeline must exist in the current scene.
Timeline	The name of the timeline to start.

WAIT

The Wait box allows you to impose a waiting time in milliseconds before pulling out the signal.

Duration	The duration in milliseconds of waiting.

WALK

The walk box allows to move a stage object to calculate a path between the current position and the destination. This is the equivalent of the walk function. The signal leaves the box when the object has reached its target.

Scene	The name of the scene of the object. If the field is empty, the object must exist in the current scene.
Object	The name of the object to be moved.
From cell	The name of the cell of the object referring to the initial position. This parameter cannot be combined with the X and Y properties.
From x/y	The initial position coordinate. This parameter cannot be combined with the Cell property.
Cell	The name of the cell of the object referring to the destination position. This parameter cannot be combined with the X and Y properties.
To x/y	The coordinate of the position to be reached. This parameter cannot be combined with the Cell property.
Animation	Force the reading of an animation while moving.
Directorate	Force a direction while moving.
Straight	Moves the object along a straight line from its initial position to the destination without taking into account the running speed.
Straight duration	The duration of the movement in a straight line in milliseconds.
Straight ease IN/OUT	Allows to apply an acceleration at the beginning and/or deceleration towards the end of the movement in a straight line.

Conditions

The conditions allow to enrich the synchronised action sequences without going through the code, while keeping the logic of the visual programming. They have a specific input and several outputs. If the output determined by the result of the condition is not wired, the signal will then be lost and its progression interrupted.

A condition can take the form of a loop if the Loop box option is enabled. However, the loop will stop turning if the called output is connected to another box. "As long as the condition is false, the sequence must be blocked" would require plugging a cable only at the first output: in case the condition is true.

The most frequent conditions are accessible via the role editor but it is quite possible to call any function of the language via the result box which will return a boolean, i.e. the result of the function itself.

BOOLEAN IS

This condition makes it possible to compare the variable as a Boolean. The result cannot therefore be only true or false.

Variable	The name of the variable to be tested.

DIALOG HAS

This condition allows you to know the states of a dialogue. The condition will be true if all the parameters provided coincide.

Dialog	The name of the dialogue to be tested.
Choice	The identification of the choice that must be visible.
Said	Identifier of the replica that must be displayed at least once.

DIALOG IS

This condition allows you to know the dialogue and the replica while running. The condition will be true if all the parameters provided coincide.

Dialog	The name of the dialogue.
Sentence	Identifier of the replica.

OBJECT HAS

This condition allows you to know the states of an object. The condition will be true if all the parameters provided coincide.

Scene	The name of the scene being executed.
Object	The name of the object in the scene.
Animation	The name of the animation being played.

Directorate	The direction of animation.
Frame	The index of the animation frame.
Cell	The name of the cell where the object is located.
Flag	The flag value of the cell where the object is actually located. This parameter does not take into account the Cell property.
Sticker	The name of the sticker of the object.
Visible	The visibility of the object.

PLAYER HAS

This condition allows you to know the states of a player. The condition will be true if all the parameters provided coincide.

Player	The name of the player to test.
Object	The name of the object in the player's inventory.

PLAYER IS

This condition allows you to know the current player and the object controlled by the player. The condition will be true if all the parameters provided coincide.

Player	The name of the player.
Object	The name of the object controlled by the current player. This parameter does not take into account the Player property.

PUZZLE IS

This condition allows you to know the state of a riddle and has four outputs:

resolved: the puzzle was solved by the player.

Resolving: the riddle is in the process of resolution (the signal has entered the box but has not yet come out).This state considers that the riddle is not yet resolved.

unresolved: the puzzle is not solved (the signal has never come out of the box). This state can be combined with the resolving state.

catch-all: this is the case by default if the signal could not exit by the other sockets.

When a gold box excludes a puzzle, the puzzle becomes exclusively an unsolved puzzle.

Puzzle	The name of the riddle.

RESULT IS

This condition allows you to call a function of the language and compare the returned value of the Boolean type. You must not use the return keyword.

SCENE HAS

This condition allows you to know the states of a scene. The condition will be true if all the parameters provided coincide.

Scene	The name of the scene to be tested. If the parameter is not specified, the current scene will be used.
Still	The name of the still that is in the scene and must be visible.
Object	The name of the object is in the scene and must be visible.
Label	The name of the label is in the scene and must be visible.

SCENE IS

This condition allows you to know the scene being executed and the previous scene. The condition will be true if all the parameters provided coincide.

Scene	The name of the scene.
Previous scene	The name of the previous scene.

SQUENCE IS

This condition allows you to know the state of a sequence and has six outputs:

started: the sequence started (the signal has already entered into the first box of the sequence).

Playing: the sequence is still running (the signal is between the first and second boxes delimiting the sequence).

ended: the sequence started and ended (the signal came out of the second box of the sequence).

unstarted: the sequence never started (the signal did not enter into the first box of the sequence).

catch-all: this is the case by default if the signal could not exit by the other sockets.

Sequence	The name of the sequence referring to the two red boxes.

VALUE IS

This condition allows you to compare a variable and divert the signal according to its value. It's kind of a four-box switch with a default.

Value1	The value of the first output.
Value2	The value of the second output.
Value3	The value of the third output.
Value4	The value of the fourth output.
Variable	The name of the variable to be tested.

Events

Events allow to receive notifications when the player interacts with the elements of the game, and therefore to be able to react by performing actions in return. Events are specific to the elements of the game (objects, labels...).

When an event is triggered, a notification is sent to all boxes referring to the latter by browsing the running roles. However, there are two ways to spread the signal through the Consume property. The default behavior is to capture the notification, which ends the signal's propagation. If Consume is at False, the signal is propagated.

In order to bring more flexibility to the events, it is possible to define the order of execution of the called boxes. The combination of these two options is particularly effective: it allows you to specifically intercept the combination of the A object with the B object, as well as, if applicable, that of the A object with any other object. In this case, you will simply create two distinct boxes of the same event with different parameters and increase the value of the execution order property of the second box. If the values are identical, you will not be able to predict the order of call. Note that the Roles also have an order of execution.

FILTERS

Filters allow you to select the calling boxes upstream during an event. Some filters do not apply to all boxes. By default, no filters are applied.

Current player	Allows you to filter according to the current player.
Current scene	Allows you to filter according to the current scene.
Resolved puzzle	Allows you to filter according to the state of a puzzle that needs to be solved.
Resolving puzzle	Allows to filter according to the state of a puzzle that must be in the course of resolution.
Unresolved puzzle	Allows to filter according to the state of a puzzle that must be unsolved.
Started sequence	Allows to filter according to the state of a sequence that must be started.
Playing sequence	Allows to filter according to the state of a sequence that must be running.
Ended sequence	Allows to filter according to the state of a sequence that must be completed.
Unstarted sequence	Allows to filter according to the state of a sequence that has not yet been started.
Variable	Allows to filter according to the value of a variable. It is necessary to specify the name of the variable, the condition operator and the value to compare.

ON CLICK

This event allows you to receive a notification when the player clicks on the scene of the game. The click area is represented by a rectangle. By default, the rectangle corresponds to the size of the scene. By returning a different value of 0, the step will be ignored.

Scene	No, no. no,
X/Y	The position of the dot at the top left of the rectangle in pixels.
Width/Height	The size of the rectangle in pixels.

ON DETACH OBJECT

This event allows you to receive a notification when the player decides to separate two items from an inventory object. To use this feature, it is necessary to add a LAYOUT_DETACH image to the project.

Object	Name of object present in the inventory.

ON DRAG OBJECT

This event allows you to receive a notification when the player decides to start a drag and drop by selecting an object with the Source or Both property. By returning a different value of 0, the drag and drop is interrupted.

Scene	The name of the scene.
Object	The name of the source object that initiated drag-and-drop. From the box script, it is accessible via the $_obj variable.
Subject	The name of the sub-object present in the object. From the script of the box, it is accessible via the variable $_sub.

ON DROP OBJECT

This event allows you to receive a notification when the player deposits the object on a static area of the scene.

Scene	The name of the scene.
Object	The name of the source object that initiated drag-and-drop. From the box script, it is accessible via the $_obj variable.

WE'RE GOING SONG

This event allows you to receive a notification when a music ends before you close.

Song	The name of the music.

ENTER DIALOG

This event allows you to receive a notification when a conversation starts.

Dialog	The name of the dialogue.

WE'RE ENTERING SCENE

This event allows you to receive a notification when the player enters a new scene. In case of black melt, the notification is received just before the opening transition when the screen is in darkness.

Scene	The name of the scene.
Mode	JUMP: The event is called in the event of a change of scene. WALK: The event is called when the character reaches the TO cell after entering the scene. SWITCH: The event is called after a change of player. LOAD: The event is called when a game backup is restored.
Cell	WALK: The name of the cell TO.

EXIT DIALOG

This event allows you to receive a notification when the conversation ends.

Dialog	The name of the dialogue.

WE'RE EXITING SCENE

This event allows you to receive a notification when the player leaves the scene. In case of melt, the notification is received immediately after the transition.

Scene	The name of the scene.

IT'S INPUT

This event allows you to receive a notification when the player clicks on the screen.

Button	MAIN: the main button (left mouse click). ALT: the alternate button (right mouse click).

ON REACH CELL

This event is called when an object enters or exits an Event cell. Consecutive cells bearing the same name are considered as a group of cells or a single cell. If the object enters a cell, you can return a different value of 0 to cancel the last move and prevent entry.

Scene	The name of the scene.
Object	The name of the object present in the scene.
Cell	The cell name is present in the scene. The default CELL name is ignored.
Mode	IN : enters a cell. OUT : exits a cell.

ON REACH SPOT

This event allows you to receive a notification when an object reaches a spot placed on a path.

Scene	The name of the scene.
Object	The name of the object present in the scene.
Spot	The name of the spot present in the scene. The default SPOT name is ignored.

ON REPEAT

This event allows for periodic notification.

Duration	Duration of interval in milliseconds.

ON SAY

This event allows you to receive a notification when the player displays a choice or a reply.

Dialog	The name of the dialogue.
Sentence	Identifier of choice or replica.
Said	Previous state.
Sub-choice	The index of the paragraph after an election, valid only for replicas.

LABEL SELECT

This event allows you to receive a notification when the player clicks on a label.

Scene	The name of the scene.
Label	The name of the label in the scene.
Movement	WALK: The event is called when the player clicks the label for the first time to reach it. SKIP: The event is called when the player clicks the label a second time while the character is moving. With this option you will be able to give the player the opportunity to shorten his movement. Think about changing the label property.

WE SELECT LAYOUT

This event allows you to receive a notification when the player clicks on a customizable button on the interface.

Button	Button ID.

ON SELECT OBJECT

This event allows you to receive a notification when the player clicks on a decor object or using the select function.

Scene	The name of the scene.
Object	The name of the object present in the scene.
Subject	The name of the sub-object present in the object.
Movement	WALK: The event is called when the player clicks the object for the first time to reach it. SKIP: The event is called when the player clicks the object a second time during the character's movement. With this option you will be able to give the player the possibility to shorten his movement. Think about changing the property of the object.
Mode	USER: triggered after player action. CALL: triggered after call of select box.

ON SUCCESS

This event allows you to receive a notification when the player solves a puzzle.

Puzzle	The name of the riddle.

ON SWITCH PLAYER

This event allows you to receive a notification when the player changes the current player.

Player	The name of the new player.
Previous player	The name of the previous player.

ON UNLOCK CHOICE

This event allows you to receive a notification when the player unlocks a hidden choice through the replicas defined in the dialog editor.

Dialog	The name of the dialogue.
Choice	The identification of the choice.

ON USE LABEL

This event allows you to receive a notification when the player uses an inventory object with a label.

Scene	The name of the scene.
Object	The name of the object in the inventory.
Label	The name of the label in the scene.

ON USE OBJECT

This event allows you to receive a notification when the player combines an inventory object with a decor object or other inventory object.

Object A	The name of the first object in the inventory.

Object B	The name of the second object present in the scene or in the inventory.
Subject B	The name of the sub-object present in the second object.

ON WALK

This event allows you to receive a notification when the character starts or finishes moving.

Scene	The name of the scene.
Object	The name of the object present in the scene.
Mode	ENTER: to indicate the start of the step. EXIT: to indicate the end of the step.

Code integration

Roles can be initiated and defined by code.

```
start_role R_HOUSE stop_role R_GARDEN
```

To communicate with the role editor, there is the task function to send a task. The task functions as a routine. First specify the name of the role where the task is located, then the name of the task and finally the argument like this:

```
task R_HOUSE name "Argument"
```

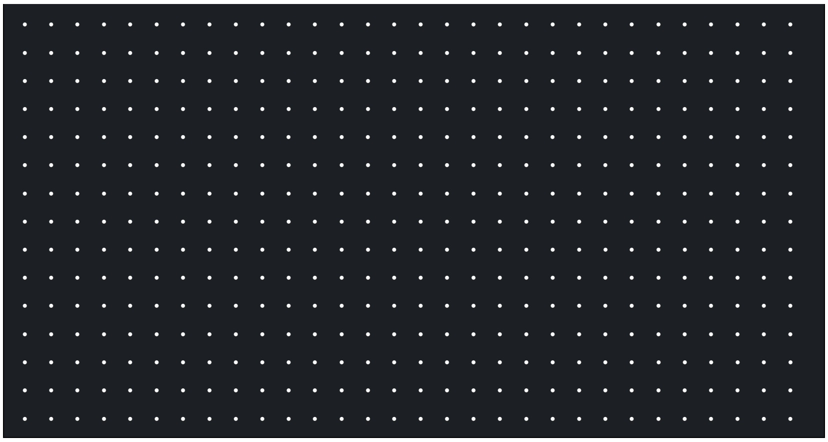
To search for a task among all the roles being executed, use the * character.

```
task * name $city
```

If there are several task boxes with the same name, only the first one found will be executed. Using the same task name in different roles may have an interest in genericity since not all of them are necessarily en route at the same time.

To recover the argument in the script of the task box, you just need to use the $_arg variable. By default, the argument is an empty text.

```
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
```



To return a boolean value to the task function, you must change the value of the $_res variable by indicating 1 or true. By default, the task function returns false.

```
set _res 1 set _res true
```

Regarding the role editor's task box, you can choose the output by calling the function out and specifying an index between 0 and 9.

```
Out 2
```

Returning a different value of 0, the n-catalogue will send no output signals.

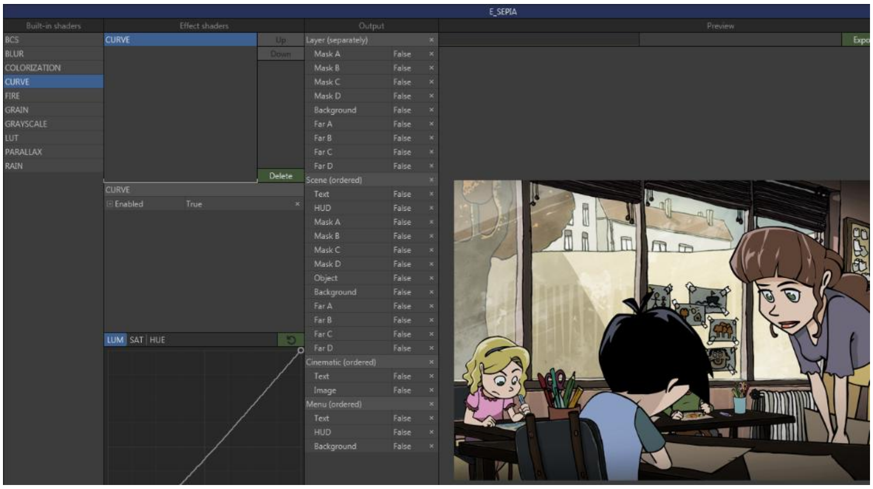
```
Return 1
```

The role must be running to receive a task, otherwise the task function returns false.

EFFECTS

The effect editor makes it possible to apply visual effects to the screen in real time by requesting the graphic card through the shaders. An effect can be associated with an entire scene or a layer from the stage editor or the whole game from the project parameters.

An effect is made up of a list of shaders because it is possible to stack them to create a succession of visual effects. To be used sparingly as this can significantly reduce the performance of the game in case of exaggeration. For a mobile game it is recommended to check the performance regularly by testing the application on several models of device.



CURVE
Enabled True

Layer (separately)
Mask A False
Mask B False
Mask C False
Mask D False
Background False
Far A False
Far B False
Far C False
Far D False

Scene (ordered)
Text False
HUD False
Mask A False
Mask B False
Mask C False
Mask D False
Object False
Background False
Far A False
Far B False
Far C False
Far D False

Cinematic (ordered)
Text False
Image False

Menu (ordered)
Text False
HUD False
Background False

Output

Some elements constituting the scene are independent such as the main decoration, the four distant decorations and the four masks in the foreground. These elements are only PNG files. By activating these layers, you will apply the effect only on the layers in question.

For the other categories, the operation is different and the effect is applied to the entire screen. The order of the parameters corresponds to the order of the display of the elements. Regarding the scene, the order of the display is as follows:

Far	Layers D, B, C and A.
Back	The main decor.
Object	Objects and labels.
Mask	Masks D, C, B and A.
HUD	Interface elements.
Text	Texts (except labels).
Cursor	The mouse cursor.

It is therefore impossible to apply the effect, for example, to masks without applying it to objects, to the main decoration or to distant decorations.

Brightness, Constrast, Saturation

This effect makes it possible to control the brightness, contrast and saturation of the image.

The brightness is between -1 and 1. Its default value is 0.

The contrast is a positive value. Its default value is 1.

Saturation is a positive value. When its value is 0, the image is completely unsaturated. Its default value is 1.

Blur

This effect allows a Gaussian blur to be applied.

The initial intensity is between 0 and 1. The max zoom allows to increase the intensity depending on the zoom of the scene to simulate a depth of field effect. If the value is equal to 1, only the initial intensity will be used. This technique is more suitable for Visual Novels when the decor is further away from the characters.

Colouring

This effect allows to apply two shades to an image in shade of grey, based on the high and low lights. If the image is in colors, it is first converted to black and white by the shader.

The Highlight color value is applied to the high lights and the Shadow color value to the low lights. By default the values are set to FFFFFF (RGB), i.e. the value of the white.

The Color smoothless value softens the transition between the two shades. To disable the transition, simply specify the value 0.

Curve

This effect makes it possible to modify the brightness, saturation and color of the image using three curves.

Fire

This effect allows to generate a realistic rendering of fire flame over the entire surface of the image with the possibility of adjusting the animation speed.

The Fusion value allows you to control the fusion mode. By default, the fire is rendered on a black background. In Addition mode, it is superimposed on the background image. Mask mode is slightly more complex to set up because the background must be a shaded image of grey, where black represents transparency.

Grain

This effect allows a dynamic monochrome grain to be applied to the image taking into account the bokeh effect if necessary.

The Gain value allows to change the size of the grain to the image.

Grayscale

This effect removes the colors from the image. This shader is faster than the BCS shader if you only want to convert the image to black and white.

Look Up Table

This effect adds a look to your image. Just choose a style from the list of available filters.

Filters can also be applied directly to the images upstream at the time of build. In this case, the graphics card no longer needs to perform a runtime processing. The major drawback of this method is the loss of quality since each transparent image is treated independently, while the effects can be applied to the entire scene. It is sometimes better to statically process images for mobile versions in order to find a good compromise between the quality and performance of your game.

Parallax

This effect allows to apply a parallax effect by providing a version of the decor (DEPTH.png) containing the distance of each pixel from the camera. White represents the elements closest to the camera.

Rain

This effect allows to generate a realistic rendering of rain over the entire surface of the image with the possibility of adjusting the intensity and colour of the drops.

The Fusion value allows you to control the fusion mode. By default, the rain is rendered on black background. In Addition mode, it is superimposed on the background image. Mask mode is slightly more complex to set up because the background must be a shaded image of grey, where black represents transparency.

Integration by publishers

After you have indicated with which elements the shaders should interact, you must associate your asset to make it visible.

For scenes, use the Effect property of the Scene group. Same for kinematics.

For the menu, the setting is in the project properties on the Effect line of the Menu group.

You will find another line Effect in the Graphics group that will allow you to associate an effect with all the scenes of the game. Note that this property will be ignored if the Effect field of a scene is indicated. It is actually a matter of defining a global effect to the game and then applying a specific effect to certain scenes. This avoids you informing the Effect field of all the scenes.

Code integration

The change of effects is made by the set_effect and set_scene_effect functions. The first function changes the effect of the game and the second function changes the effect of a scene. If the parameter is empty, no effect will be used.

```
set_effect E_NIGHT set_scene_effect S_GARDEN E_DAY
set_effect
```

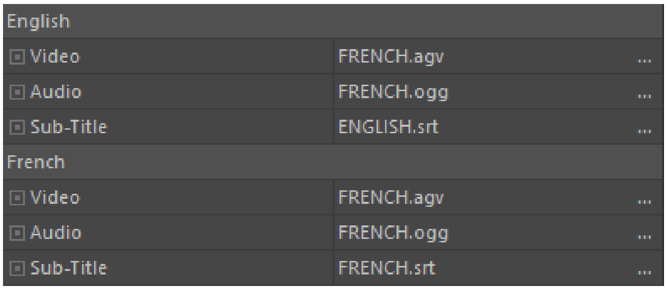
CINEMATIC

In an adventure game, cinematographic sequences or cutcenes are often used in the full screen to present the story, the characters, the environment or strengthen the plot at key moments of storytelling. These are passages that bring a lot of additional information to the game and that can hardly be integrated into the gameplay and puzzles. These sequences are also used to reward the player after a long stage of riddles solved to give him a moment of respite, and to make him want to pursue the adventure towards another goal.

There is an alternative method for making in-game cutscenes directly in the stage editor using the timeline, which will not be dealt with in this chapter. Please refer to the chapter on the stage editor.

Video, Audio and Subtitle

A kinematic consists of a video file in MP4 format, an audio file in OGG format and a subtitle file in SRT format.



English		
⊡ Video	FRENCH.agv	...
⊡ Audio	FRENCH.ogg	...
⊡ Sub-Title	ENGLISH.srt	...
French		
⊡ Video	FRENCH.agv	...
⊡ Audio	FRENCH.ogg	...
⊡ Sub-Title	FRENCH.srt	...

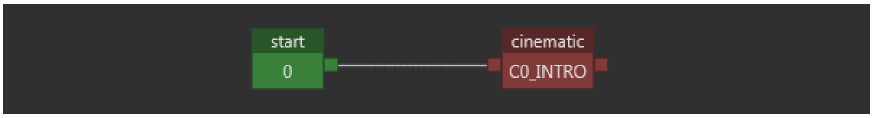
After creating a new Cinematic asset from the main toolbar or the ALT+C shortcut, a folder is added to the project to place the video, audio and subtitles of all languages. You have the possibility to associate a file with each element of a language and thus reuse the same file for different languages. It may be that the video is unique and the sound files are specific to each language.

Encoding an MP4 video file

For reasons of optimal compatibility, it is preferable to encode videos in H.264 Baseline 3.1 format in 1280x720 pixels and 24 images per second.

Code integration

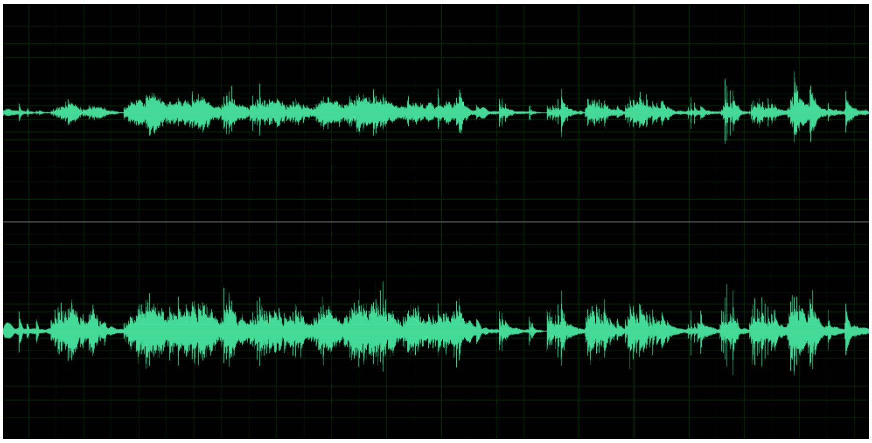
In order to launch a kinematic in your game, you need to modify a role by adding a new Cinematic box. Change the Cinematic parameter to specify the set of your video. Link it to the other boxes according to the composition of your story. The program will leave the box at the end of the video.



To force the player to watch the video in full, open the assembly and change the Skip option to None.

AUDIO

Audio has an important place in adventure games: especially music and voices. That's why seccia.dev offers simple but essential features for sound integration.



Music and sound atmosphere

The music and sound ambiences are in OGG (Windows) and MP3 (iOS, Android, Web) formats are placed in the SONG folder at the root of the project. Four tracks are proposed, allowing to play up to four files simultaneously in order to dissociate the sound ambience from the music. You can string two musics or sound ambiences on the same track using the song box.

Noise

The noises are short audio files in WAV format and located in the SOUND folder at the root of the project. These sounds are played on independent tracks.

```
play_sound "BOOM"
```

Votes

The voices in OGG (Windows) and MP3 format (iOS, Android, Web) are placed in the VOICE folder of the Dialogue Assets. This folder contains one file per replica. To correctly name your files, you need to recover the box identifier in the editor and specify the name of the language as follows:

D_MYDIALOG\VOICE\EN_16.ogg D_MYDIALOG\VOICE\EN_16.ogg

If several paragraphs separated by a line are present in the same box, you must add its index after the identifier except for the first paragraph:

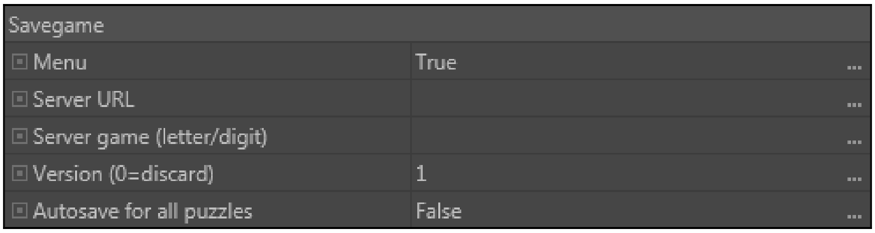
D_MYDIALOG\VOICE\EN_16.ogg D_MYDIALOG\VOICE\EN_16_1.ogg D_MYDIALOG\VOICE\EN_16_2.ogg

Identifiers are accessible from properties and stored in HTML files generated via the TEXT/Export to HTML menu.

SAFEGUARD

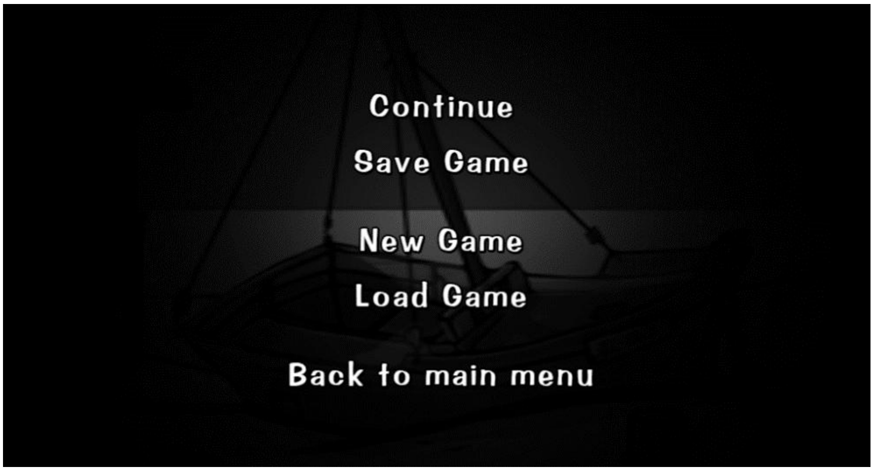
Games created with seccia.dev can contain up to six backup locations to allow players to retake their game whenever they like. There is a need to differentiate between local backups and more complex online backups to set up.

To make the backup menu appear in the game, you must enable the Menu property in the project settings in the Savegame section.



Savegame	
Menu	True
Server URL	
Server game (letter/digit)	
Version (0=discard)	1
Autosave for all puzzles	False

Otherwise only the functions of the language will allow to manage the backups.



Local backup

Location 0 is reserved for automatic backup activated by the save function. It's up to you to call this function when you want to save the player's progress. There can only be one automatic backup per game. In case of new games, the old backup will be overwritten by the new progression. If you want to load this game following an action without going through the menu, you can do it by calling the load function.

Locations 1 to 4 are reserved for manual backups. If the menu is disabled or you use a custom menu, you can still access these locations by calling the load_game and save_game functions.

The last location with number 5 is reserved for online backups.

To display an icon on the screen during backup to inform the player, place a LAYOUT_AUTOSAVE file in the IMAGE folder of your project.

Online backup

Online backup consists of keeping the game progress on a web server to continue the game on another platform or not to lose data in case of re-installation of the application or loss of the device. Only one backup per user account is allowed.

The implementation of the online backup system requires a website with at least version 5.5 of PHP installed. No database is required. Android requires an SSL certificate to establish a secure connection via HTTPS protocol. I advise you to opt for a secure solution on all platforms. Some hosting providers provide one for free.

You will find explanations for installing scripts on a web server in the Server folder of the software.

PHP scripts will allow you to store on the same disk space of your site, user accounts of all your games created with seccia.dev.

Since the scripts collect personal information from your players, you must comply with the general GDPR data protection regulations by providing, in the project settings, the link to the page of your privacy policy. Please direct yourself to the official bodies to read the regulations in order to properly write your document.

Chapters

With regard to an episodic game, you may have to unblock chapters from a custom menu. In order to generate backup files per chapter, you need to launch a part of your game, save the progress to the desired place, recover the SAV file produced by the application and transfer it to the SCENARIO folder by naming it as follows:

SCENARIO\CHAPTER1.sav SCENARIO\CHAPTER2.sav

These read-only files will be integrated into the application when building and the load_chapter function will allow you to load them.

```
load_chapter 1
```

Pay attention to the compatibility of backups between two versions of your application. On the other hand, the format is universal and will work on all platforms.

Shared data

It is quite possible to save additional data that will be shared by all parts of the device. These data are game-specific, such as unlocking a chapter or an event. They cannot be saved online.

Here are some examples of the function code to use:

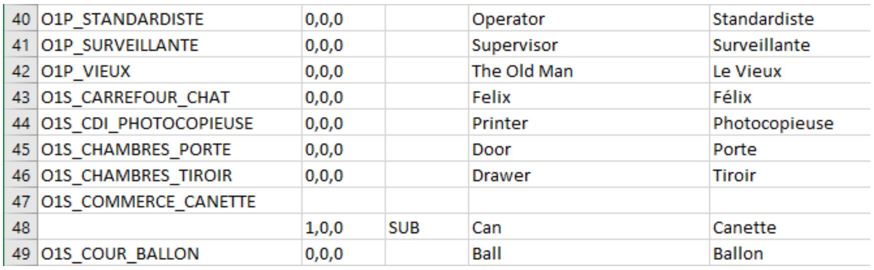
```
write_game_value "CHAPTER2" 1 write_game_value "CHAPTER3" 0
write_game_value "SCORE" 10 read_game_value variable
"SCORE" alternative $variable
```

LOCATION

I explained to you at the beginning of the book that one of the particularities of the narrative adventure game is the importance of history, dialogue and staging, which play a key role in enriching gameplay at the same level as puzzles. Therefore, if narration is one of the pillars of the structure of your game and dialogues are awkward, the gaming experience will be completely wasted and the players will miss out on your universe. This implies that the texts must be well written and translated.

seccia.dev offers two editing modes to facilitate the writing of dialogues and translation into many languages, without typing a single line of code.

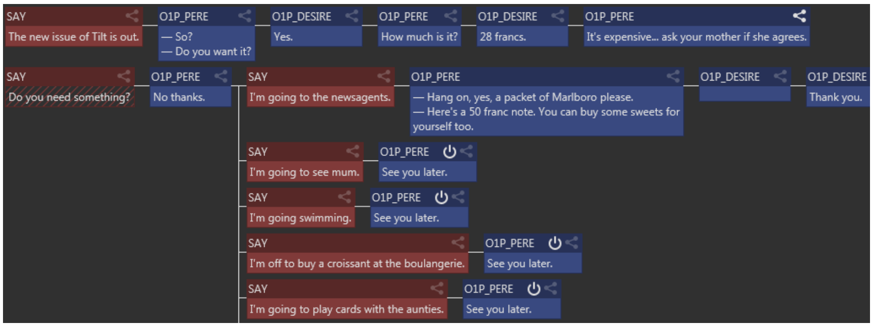
Before you start, you must first activate the desired languages in the parameters of the project. For more information on this I refer you to the chapter Project.



40	O1P_STANDARDISTE	0,0,0		Operator	Standardiste
41	O1P_SURVEILLANTE	0,0,0		Supervisor	Surveillante
42	O1P_VIEUX	0,0,0		The Old Man	Le Vieux
43	O1S_CARREFOUR_CHAT	0,0,0		Felix	Félix
44	O1S_CDI_PHOTOCOPIEUSE	0,0,0		Printer	Photocopieuse
45	O1S_CHAMBRES_PORTE	0,0,0		Door	Porte
46	O1S_CHAMBRES_TIROIR	0,0,0		Drawer	Tiroir
47	O1S_COMMERCE_CANETTE				
48		1,0,0	SUB	Can	Canette
49	O1S_COUR_BALLON	0,0,0		Ball	Ballon

Internal editing

The first step is to write the original text using the software interface. The titles of the elements are editable in the stage and object editors. The dialog editor allows to build a full conversation between one or more characters, while offering choices of replica to the player. Unlike the scenario, the conversations are not graphs but arborescences: a box can have only one direct parent but several children.



SAY
The new issue of Tilt is out.

O1P_PERE
— So?
— Do you want it?

O1P_DESIRE
Yes.

O1P_PERE
How much is it?

O1P_DESIRE
28 francs.

O1P_PERE
It's expensive... ask your mother if she agrees.

SAY
Do you need something?

O1P_PERE
No thanks.

SAY
I'm going to the newsagents.

O1P_PERE
— Hang on, yes, a packet of Marlboro please.
— Here's a 50 franc note. You can buy some sweets for yourself too.

O1P_DESIRE

O1P_DESIRE
Thank you.

SAY
I'm going to see mum.

O1P_PERE
See you later.

SAY
I'm going swimming.

O1P_PERE
See you later.

SAY
I'm off to buy a croissant at the boulangerie.

O1P_PERE
See you later.

SAY
I'm going to play cards with the aunties.

O1P_PERE
See you later.

Regardless of the editor used, you can edit texts from all languages at any time without leaving the software environment. This method is convenient to quickly correct errors or make minor changes to translations.

On the other hand, I advise you not to proceed this way to fully translate your game. Instead, opt for external editing.

External editing

External publishing has three main advantages:

• ensure efficient translation • work with external translators • benefit from the UTF-8 format

This method consists of exporting the entire text to CSV files in UTF-8 format editable in Excel, and reimporting them after modifications. It is important to keep the UTF-8 format at record and keep the comma as a column separator.

Why is it more efficient? Simply because it is more convenient to translate a single CSV file on two columns than to go open all the assets within the software. You get better visibility of the work to provide and you can use your usual spreadsheet.

The second advantage allows you to send texts to translators without having to transmit the sources of your game. Furthermore, seccia.dev generates a CSV per language, which facilitates file exchange when several translators work on the same project.

Finally, the third advantage is to be able to translate the text using other alphabets not recognized by the software thanks to the UTF-8 format of CSV files.

This method can also be used for rereading the original text before the localization phase. HTML export is also a good alternative and the format is more easily converted to PDF.

Latin Alphabet, Cyrillic and others

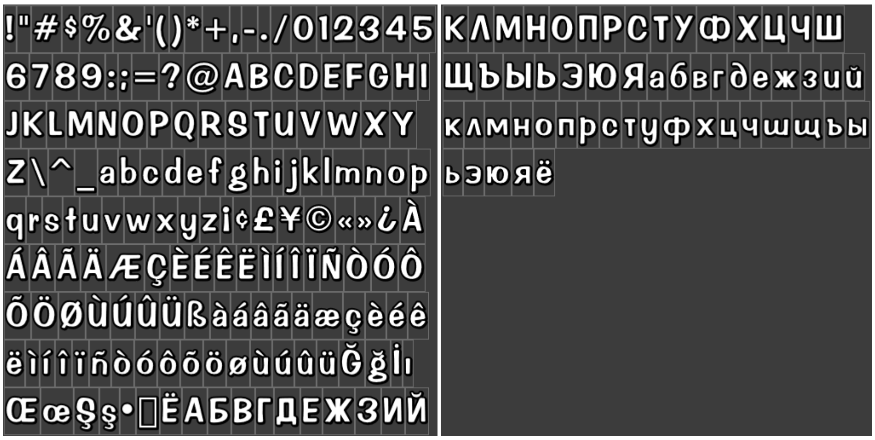
The seccia.dev interface does not support Unicode, only ASCII format. Therefore it is necessary to go through CSV files to translate your text into Russian or Chinese for example.

Unicode characters are coded to hexadecimal like this: ~FFFF. To convert a Unicode text into this format, just copy the text into the clipboard and click on the TEXT/Unicode to seccia.dev menu. The content of the clipboard will then be modified and you can paste the new text into a field of the software.

For Cyrillic languages, remember to include the Cyrillic alphabet in the generation of your font. Note that existing fonts were generated with only one alphabet for graphic optimization reasons.

As far as Asian languages are concerned, the way to proceed is slightly different because it is not possible to generate all the symbols of the language in advance. At the time of the build of your game, seccia.dev only recovers the symbols present in your texts in order to limit the number of textures to be generated. Asian fonts are used by default but you can choose others from the properties of the project by writing their exact name.

Turkish letters not included in the ASCII character set are included in the Latin alphabet of seccia.dev.



SCRIPT

We've come to the sensitive subject. Even if you're not a developer in your soul, learning seccia.dev code should not slow you down, but rather encourage you with its minimalist programming language.

The language is case sensitive only for commands and functions. For example if can't write If, iF or IF.

Comments begin with two slash bars and end at the end of the instruction line.

```
// if the current scene is S_PARK or S_SUBWAY if scene
S_PARK S_SUBWAY // change the current scene jump S_HOME end
// test variable value set pseudo "Sylvain" if var pseudo
"Sylvain" "Mike" "Bob" // here end // play animation
animate O_HERO FIGHT * animate O_HERO FIGHT
```

The asterisk keeps the default value of the parameter. In the example, the direction parameter is ignored to play an animation without changing the direction of the object. Since it is the last parameter of the function, it is possible to exclude it.

Syntax

Only one instruction per line is allowed. One instruction consists of three elements: the command, the function and the parameters. The command and the parameters can be optional. The return command does not require any function or parameters. The else, end, break and continuous commands do not accept any function or parameters.

```
command function param1 param2 ...
```

A command allows you to apply conditions to a block of instructions, to perform loops of instructions and to return an integer. To define a block, simply place the instructions below the command. The indentations are inserted automatically by the code editor but are not mandatory. Here are some examples:

```
if function param // instructions else_if function param //
instructions else // instructions end
```

The if, if_not, while, while_not commands are used to open a new instruction block. To close this block, you must add the end command.

```
if function param // instructions end
```

The else, else_if, else_if_not commands are optional and used to add a condition to an if or if_not block before end.

```
if function param // instructions else // instructions end
```

The while and while_not commands must always be closed with end. They can also contain break and continuous commands.

```
while function param // instructions end
```

As for the return command, it does not need a block of instructions and is used on a single line. The interpreter will then immediately leave the script without executing the remaining instructions. In the code below, the first instruction will prevent execution of the following two instructions.

```
return return 1 return android
```

Here is a summary of the orders:

if	Allows to run the block of instructions below if the expression is true.
if_not	Allows you to execute the block of instructions below if the expression is false.
Else	Allows to run the block of instructions below if the previous expressions are false.
else_if	Allows you to run the block of instructions below if the expression is true and if the previous expressions are false.
else_if_not	Allows you to execute the block of instructions below if the expression is false and if the previous expressions are false.
while	Allows to run the block of instructions below as long as the expression is true.
while_not	Allows to execute the block of instructions below as long as the expression is false.

end	Allows you to close a block if, if_not, while or while_not.
Break	Allows you to interrupt the while or while_not loop and continue running the script after the end, i.e. after the block is closed.
continued	Allows you to return immediately to the while or while_not statement of the current loop. If the condition is wrong, the loop will end normally.
return	Allows you to leave the script immediately by returning an integer. By default, the scripts return the value 0.

Variable

The language uses only one type of variable: strings of characters, i.e. texts. When functions expect a numerical value, the text is then converted into numbers. Quotes become optional for numerical values and text without space.

To define a new variable or change its value:

```
set foo "monday" set foo 10
```

To concate two texts, two possibilities:

```
// Method 1 set foo "the sky is " " blue" // Method 2 set
foo "the sky is " cat foo "blue"
```

To manipulate numbers:

```
set foo 10 // foo is 10 add foo 1 // foo is 11 sub foo 3 //
foo is 8 mul foo 2 // foo is 16 div foo 4 // foo is 4 mod
foo 2 // foo is 0
```

To compare two texts:

```
set foo "monday" if var foo "monday" // if foo is monday
end if var foo "monday" " Tuesday" // if foo is monday OR
Tuesday end
```

To compare two numbers:

```
set foo 10 if equal $foo 10 // if foo is 10 end if less
$foo 10 // if foo is less than 10 end
```

If you take a look at the documentation of these functions, you will notice the difference between the name parameter and the value parameter. For name it is the name of the variable without quotation marks and for value it is the value of the variable. But is it possible to specify the value of another variable for the value parameter? Yes of course and in two different ways.

The first method allows you to recover the value of a variable by its name using the $ character as in PHP.

```
set a "monday" set b $a
```

This technique is obviously valid for all functions and in the role editor.

```
set foo "THEME" play_sound $foo
```

The second method is slightly more complex but in any case it will be less useful to you. When a variable contains the name of another variable, it is possible to recover its value with the & character.

```
set a "monday" set b "a" set c &b
```

The first instruction changes the value of the variable a. It goes the same way for the second instruction that changes the value of the variable b. So we have a="monday" and b="a". Using the character &, the third instruction actually recovers the value of the variable a. The big advantage of this method is to be able to form variable names when running the game to modify or recover their value. Here is an example that displays the Italian text on the screen.

```
set countries_0 "France" set countries_1 "Italy" set index
1 set variable "countries_" $index alert &variable
```

The functions prefixed by list_ make it easier to manipulate texts in a list. A list is nothing but a text containing values separated by a comma. Here is an example that displays the French text on the screen.

```
set countries "Italy,France" list_add countries "UK"
list_sort countries list_get countries 0 first_country
alert $first_country
```

The name of a riddle is prefixed by the # character and can only be used by functions referring to the scenario.

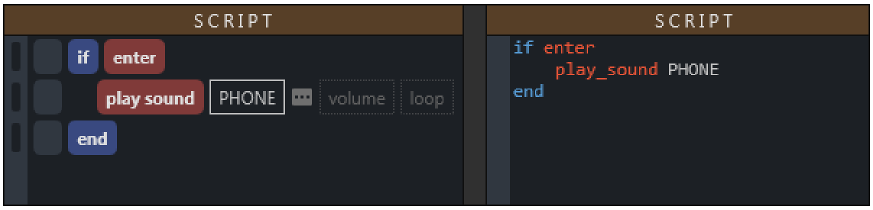
```
success #get_key if resolved #open_door end
```

Tags must be prefixed by the @ character.

```
if scene @prologue end
```

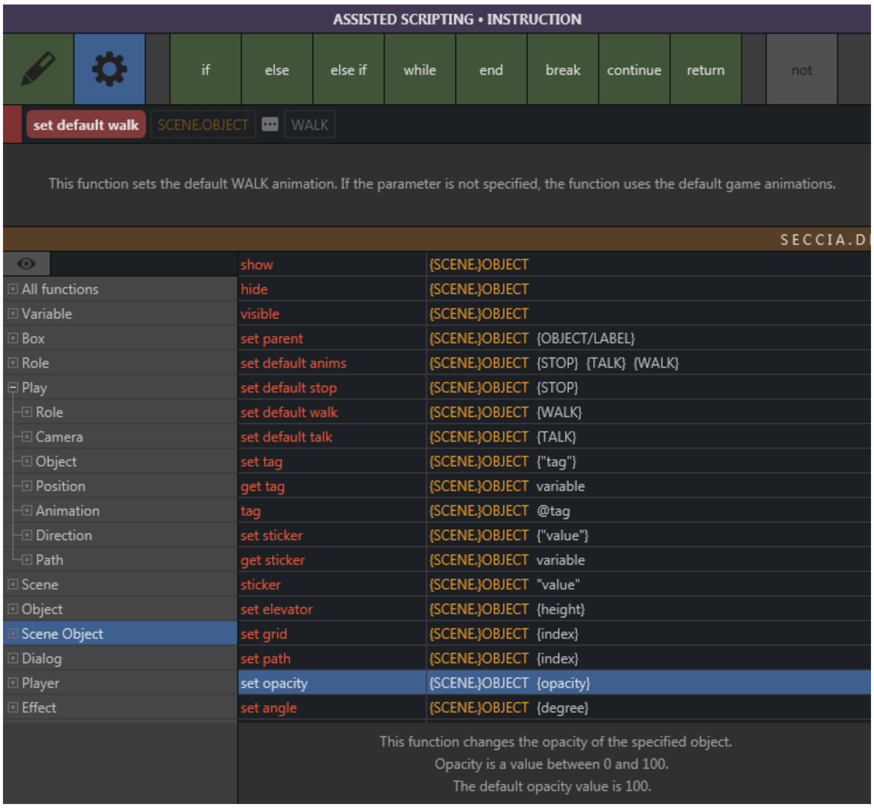
Assisted Scripting

seccia.dev integrates a more intuitive mode to edit scripts: assisted Scripting. This mode is enabled by default at the first installation of the software and aims to facilitate the editing of scripts without going through the classic code editor, i.e. typing the instructions on the keyboard. It is only a different graphical representation of the script. The code remains unchanged. The transition from classical mode to graphic mode is possible at any time by clicking on the Assisted Scripting icon of the main toolbar. You can compare the two modes on the image below.



```
if enter
    play_sound PHONE
end
```

Even if you're a coder, Assisted Scripting can help you develop your game more effectively with the help of dialog boxes.



This function sets the default WALK animation. If the parameter is not specified, the function uses the default game animations.

All functions	show	{SCENE.}OBJECT
Variable	hide	{SCENE.}OBJECT
Box	visible	{SCENE.}OBJECT
Role	set parent	{SCENE.}OBJECT {OBJECT/LABEL}
Play	set default anims	{SCENE.}OBJECT {STOP} {TALK} {WALK}
Role	set default stop	{SCENE.}OBJECT {STOP}
Camera	set default walk	{SCENE.}OBJECT {WALK}
Object	set default talk	{SCENE.}OBJECT {TALK}
Position	set tag	{SCENE.}OBJECT {"tag"}
Animation	get tag	{SCENE.}OBJECT variable
Direction	tag	{SCENE.}OBJECT @tag
Path	set sticker	{SCENE.}OBJECT {"value"}
Scene	get sticker	{SCENE.}OBJECT variable
Object	sticker	{SCENE.}OBJECT "value"
Scene Object	set elevator	{SCENE.}OBJECT {height}
Dialog	set grid	{SCENE.}OBJECT {index}
Player	set path	{SCENE.}OBJECT {index}
Effect	set opacity	{SCENE.}OBJECT {opacity}
	set angle	{SCENE.}OBJECT {degree}

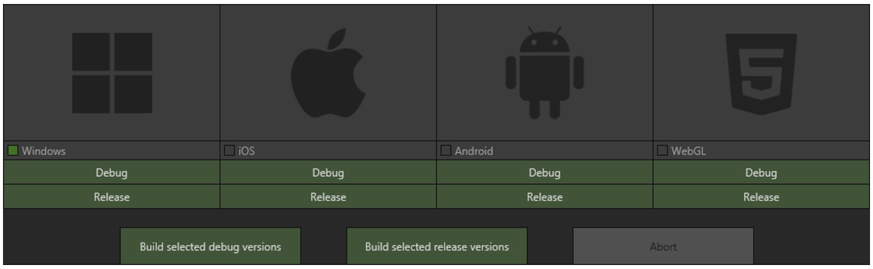
This function changes the opacity of the specified object.
Opacity is a value between 0 and 100.
The default opacity value is 100.

The main dialog box allows you to compose your instruction by going to search the elements in a totally intuitive way. Unlike the classic code display, all the available parameters of a function are visible in a blink of an eye and are obviously interactive. The language keywords are defined at the top of the window and all the functions on the left are classified by category. After you have validated a function, you simply fill in the parameters by clicking on it and following the indications provided on the screen. The instructions are editable directly from the scripts without reopening the main dialog box.

BUILDING

If you have correctly followed the steps of the first chapter, you will be able to export your game to these platforms, otherwise you will have to settle for the Windows version.

Apart from the graphic assets located in the PLATFORM folder, a seccia.dev game does not need any port to run on all the proposed platforms.

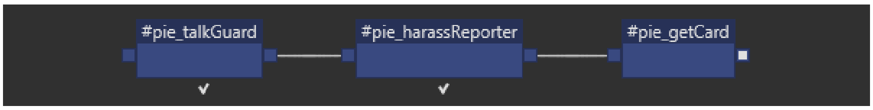

Windows
iOS
Android
WebGL
Debug
Release
Build selected debug versions
Build selected release versions
Abort

Live, Debug and Release

For each platform, you can export your game in two formats: Debug or Release.

Release is reserved for distributable versions. These versions contain all the features and optimizations of your game that can take a lot of time at the time of generation.

Debug is exclusively for development purposes. These buildings allow you to test the game by simulating the state of a game. If you select a scenario box and press the space bar, you will see a symbol appearing below the box. This marker tells you that the puzzle is supposed to be solved in Debug mode.



Live works like Debug but allows you to launch the game faster by avoiding compression of textures. PNG images are read directly by the graphics card without optimization, which can result in a difference in rendering and performance. This configuration is very useful to improve your project efficiency by limiting the generation time of Windows and Web buildes during development. These builds are saved in the respective play_windows and play_web folders.

Android

Before you can generate the Android build, you first need to create a Keystore file with the JDK keytool. This file will be used for all your projects. You then need to fill in the Signature group fields in the project properties: the file path, the name and passwords you used with the keytool. Depending on the configuration, you will either get an APK file or an AAB file.

iOS

seccia.dev generates a folder that will have to be opened and compiled on macOS with the Xcode app distributed for free by Apple. On the other hand, a paid developer account is needed both to publish your game on the AppStore and to test it on tablets and phones.

Since the appearance of M chips, you can run an iOS app on macOS. I invite you to check with Apple to follow the steps of publishing an iOS game on macOS.

As an independent, I advise you not to publish two versions: one for iOS and another for macOS. The production is time-consuming.

Web

Two distributions are possible:

public	The default version to upload to your website.
Local	The local version works with a local server accessible at: http://localhost/

As for the local version, now as a security measure browsers systematically block connection to the web application when you use the file protocol to specify the folder path. It is therefore necessary to pass through a local server on port 80. seccia.dev s的en load if you test your game by clicking Play release game. The server will be automatically disconnected when the project or software closes.

Post-build

It is possible to run a command line program for each platform at the end of the generation. This feature only applies to build release.

To change the Windows executable icon, download the Resource Hacker application at the following address:

https://www.angusj.com/resourcehacker/

Then add the following command line to the properties of your project by replacing "exe" with the full path of the executable generated by seccia.dev and "ico" by the full path of the icon file in .ico format:

```
ResourceHacker.exe -open "exe" -save "exe" -action modify -res "ico" -
mask INGOROUP,103,
```

Command line

You can generate your build release on the command line, simply adding the -build argument after the file name.

```
seccia.exe "d:\game\game.seccia" -build
```

All platforms selected and registered in the project will be taken into account.

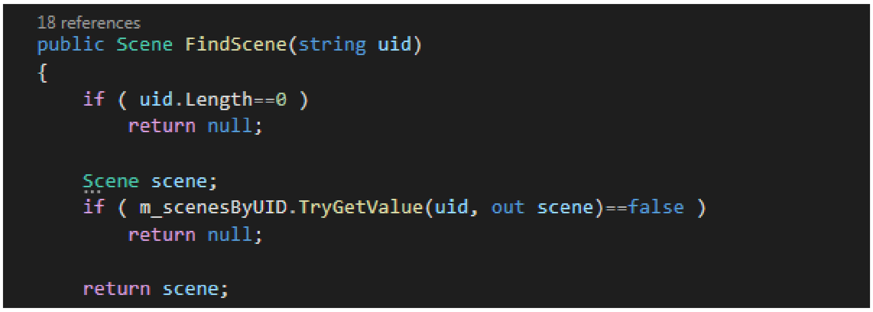
The executable seccia.exe allows you to edit JSON files by modifying string-like values in the command line. You can thus modify the project settings to generate alternative build through a .BAT file. For example if your project includes only the Windows platform and you want to generate an Android build at the command line, you can make a temporary copy like this:

```
[BILD.bat]

copy d:\game.seccia d:\temp.seccia

seccia.exe -json d:\temp.seccia "project build windows" "0" "project
build android" "0"

seccia.exe d:\temp.seccia -build

del d:\temp.seccia
```

The tool will not be able to add new values but only modify them.

UNITY PROJECT

The runtime of games designed with seccia.dev is fully developed with Unity. The Unity project is therefore provided with the software so that you can compile it with your Unity license and possibly add new features. The only drawback is to have to postpone the changes to each new update of the software. Therefore it is better to add files and only modify AgePlugin.cs to facilitate maintenance of the updates.



```
18 references
public Scene FindScene(string uid)
{
    if ( uid.Length==0 )
        return null;

    Scene scene;
    if ( m_scenesByUID.TryGetValue(uid, out scene)==false )
        return null;

    return scene;
```

Exposed functions

To avoid changing the source code of the project and breaking something, I set up an AgePlugin.cs source file by setting up a list of functions and events. Simply change the code of these functions.

```
public static bool OnAppInit() { return true; } public
static void OnAppQuit() { application.Quit(); } public
static void OnAppUpdate() { } public static bool
IsLoadEnabled() { return true; } public static bool
IsSaveEnabled() { return true; } public static bool
OnUserButton(int index, string url) { return true; }
```

Callback

The Callback function of the C# code is called when you use the callback function in your game. Both parameters are optional and have no particular meaning. You can use the first parameter to designate an event name and the second parameter for its value. The return Boolean value is sent to your game synchronously. Here is an example of using the Callback function.

```
public static bool Callback(string param1, string param2) {
switch ( param1) { box "unlock": if ( param2=="1" ) // code
here break; } return true; }
```

Interlude

You can start an action from the Unity project as soon as the player recovers his hand on the game. Unlike the Callback function that is called immediately, the Interlude function is asynchronized.

```
public static void Interlude(string param) { return true; }
```

If the application is waiting to call the Interlude function, it is possible to cancel the current query using the cancel_interlude function.

If you want to block the application during the interlude, you should call the static functions G.InterludeLock and G.InterludeUnlock of the Unity project. If you specify a duration at the first function, you will not need to call the second function unless you shorten the initial delay.

When the application is blocked, you can customize the game screen using the OnInterludeDraw function. This function must return the true value to be able to use it correctly. The following code will display a white screen.

```
public static bool OnInterludeDraw() { G.FillScreen(new
Color(1.0f, 1.0f, 1.0f, 1.0f)) return true; }
```

Finally, if you need to validate a riddle since the Unity project, you can use the G.Success function by specifying the box identifier. This identifier is to be recovered in the scenario editor.