Raphael Digital Art

Manual Ver. 1.0

User Interface

Summary

1. Introduction to Raphael Digital Art: Main Features

Raphael is a collaborative project aimed at building a platform for creating images and interactive installations for artists. The project's goal is to provide tools for artists who want to explore digital art, even if they do not have an established background in computer science or programming.
The three main features are block-based programming, a multi-level structure, and collaborative building.

Block-based programming allows users to easily and intuitively develop their work without needing to worry about technical aspects, focusing instead on the artistic ones. The "blocks" are data and image processing modules designed to be easily combined, similar to "Lego" constructions. This makes it extremely simple to use interactive devices such as cameras, image processing filters, video effects, etc.

The multi-level structure consists of three levels of complexity, allowing the artist to progress in their development of artistic language. These levels are:

- **Level 1 (Basic)**: Simple use of the interface to create works. No programming skills or extensive experience with the platform are required.

- **Level 2 (Advanced)**: Use of the advanced features of the interface. While programming knowledge is not required, a deep understanding of the platform and the portal is necessary to use plug-ins created by artists who contribute to the platform's development. At this level, users can create scripts (or shaders) with a low learning curve.

- **Level 3 (Developer)**: Allows users to develop their own "blocks" and decide whether or not to share their creations with other users according to the Living Lab rules. Development is facilitated by powerful modular templates, such as "graphic shaders," which are simplified for use through the platform's infrastructure. A basic understanding of C++ is required, though this can be easily acquired through an introductory course.

The platform is built on the principles of the Living Lab, encouraging collaborative construction through the continuous enrichment of processing blocks developed and shared by advanced users. A dedicated portal for the project allows users to follow the social life of the Living Lab, the resources developed collaboratively, and the works and projects created by artists using the platform. It also offers training resources (manuals, webinars, courses, workshops). The Living Lab's future roadmap includes the development of a "marketplace" where artist-developers can sell both digital artworks and complex, aesthetically-driven blocks.

1.2 Support Resources

The Living Lab portal can be accessed at: www.raphaeldigitalart.it
From the portal, you can download the latest version of the platform and the manuals.
You can also access the YouTube channel, which contains video tutorials.

1.3 Platforms and Language

Currently, Raphael Version 1.0, although developed using multi-platform technologies, runs exclusively on Windows. Future versions will also support macOS and Unix platforms.
The two supported languages are Italian and English.

1.4 Licenses

The Raphael platform is available in different versions:

- **Basic Version** (single license with no time limits, commercial use allowed) includes the core of the platform with many modules (blocks) for processing images, video, and installations. This includes the following module packages:

  System

  - **Blocks**
  - **Scalar**
  - **Graphs**
  - **Control**
  - **Flow Control**

  Math

  - **Int**
  - **Bool**
  - **Float**

  Effects

  - **Image-Video Effects**
  - **Filters**

  Composition

  - **Geometry**
  - **Color**

- **9 Plugin Packages** available for separate purchase (single license, no time limits, commercial use allowed). Each package includes multiple blocks that relate to the same theme. These include:

    Applications

    - Multivision
    - Mapping
    - Particles

    Interaction

    - Camera Interaction
    - Sound Interaction

    Art

    - Attractors
    - Artistic
    - Fractal
    - Music

In the future, new packages may be purchased, developed by both Raphael Digital Art developers and external users (partners).

- **Full Version** (single license, no time limits, commercial use allowed) includes the Basic Version plus all additional packages in a single installation.
- **Demo Version**: Same content as the Basic Version but with certain limitations, including non-commercial use and the ability to save images and videos at a resolution of about 1 Megapixel (WCGA = 1280x720).

1.5 Installation

Simply run the installer downloaded from www.raphaeldigitalart.it corresponding to the product purchased.

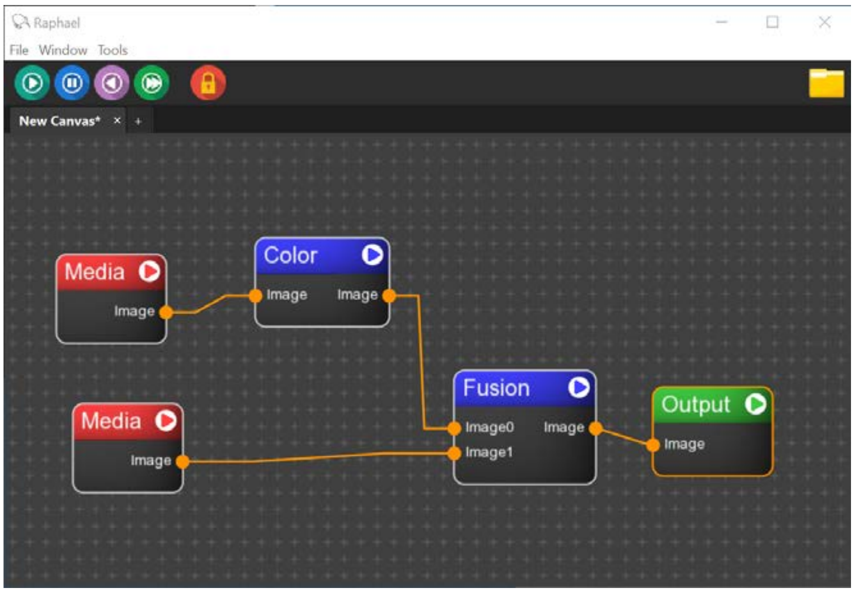
2. Block-Based Graphic Programming

To create an application, it is necessary to define the flow of operations that need to be performed in each "frame" of time. The duration depends on the complexity of the operations and is measured in fps (frames per second). In dynamic installations, the duration typically ranges from 10 to 60 fps (reference value).

The main interface of the platform describes the flow of the frame, which is continuously repeated. The flow of operations to be performed is called **Setup**.

A setup is a configuration of modular operations called **Blocks**. It can be created and saved to a file, and it can later be loaded from a file and modified (see Main GUI Interface).

The result of the processing is collected in the **Output** blocks, which represent the final step that will be sent to a display device (screen, monitor, projector). A setup can generate dynamically varying images based on the processing (blocks) contained within it.



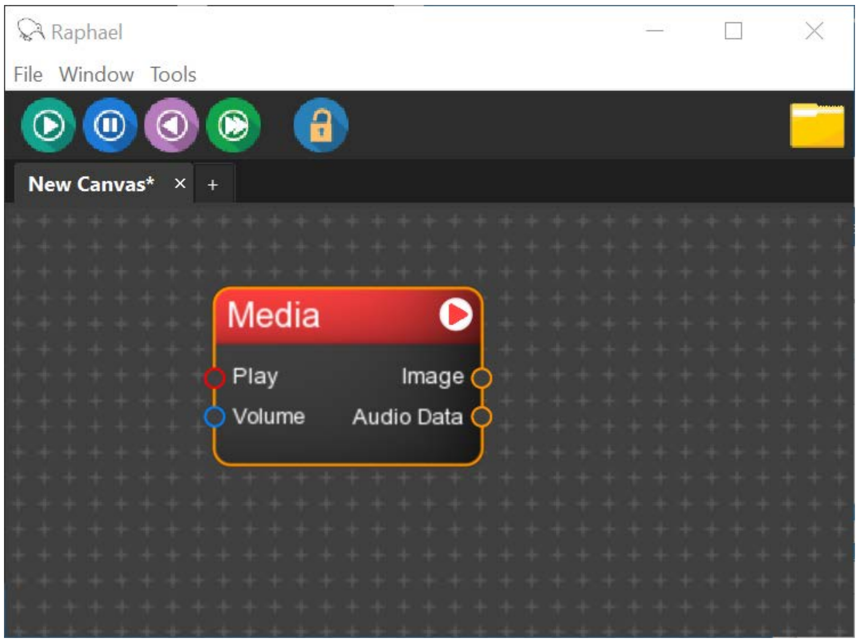
By convention, the flow of the setup starts on the left side and ends on the right side, where the outputs are placed. There are several types of blocks:

- **System Blocks**: Blocks that handle file loading (images, videos, audio clips), output to display screens, input from devices (cameras, microphones), control blocks (switch, play, resolution control, timer…), operations on scalars.

- **Mathematical Operators**: Blocks that operate on single variables (integers, floats, booleans, addition, subtraction, division, multiplication, logical operators).

- **Effects**: Blocks that perform specific image processing tasks.

Each block has input parameters, control parameters, and output parameters.



Input parameters are data coming from outside the block and cannot be modified within the block. Among these, there is generally at least one texture (image) as input.

Control parameters control the internal operation of the block. They can be modified through a specific interface (GUI Details) and saved in the setup.

Output parameters constitute the block's output, data available for subsequent blocks. Generally, the output includes at least one texture (image).

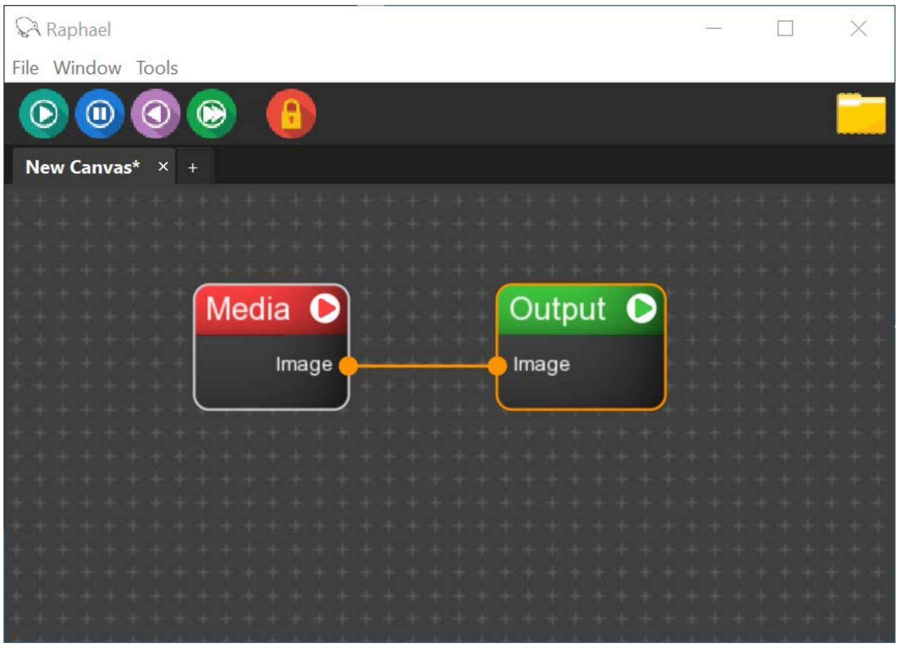
In Chapter 3, the main interface will explain how to display these parameters and modify the control parameters.

Input and output parameters can be "textures" (images) or variables in various formats (float, integer, boolean variables (0/1), strings, vectors of variables, or entire C++ classes).

2.1 Setup

A setup is a specific configuration of blocks that generates outputs.

The figure below shows a simple example of an image being loaded through the "Media" block and output to a display output ("Output").



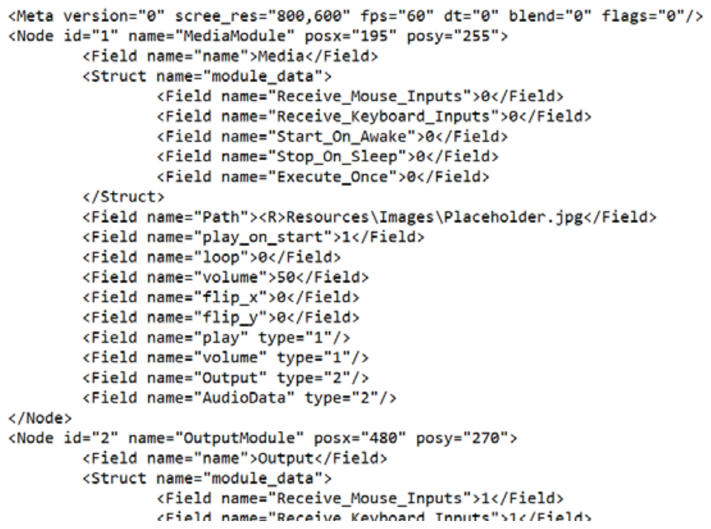
The creation of setups is simplified by block-based programming in the setup area, which allows you to define the flow of processing by placing various elementary blocks and connecting them with lines.

From the user interface (see 3.1.1), the setup can be saved to a file or loaded from a previously saved file. Setup files have the ".icf" extension.

The setup file contains:

- All blocks and their connections (connections, inputs, outputs).
- All parameters of the individual blocks.
- Display and resolution information.

The ".icf" files are in text format, making them readable. Here is a partial example:

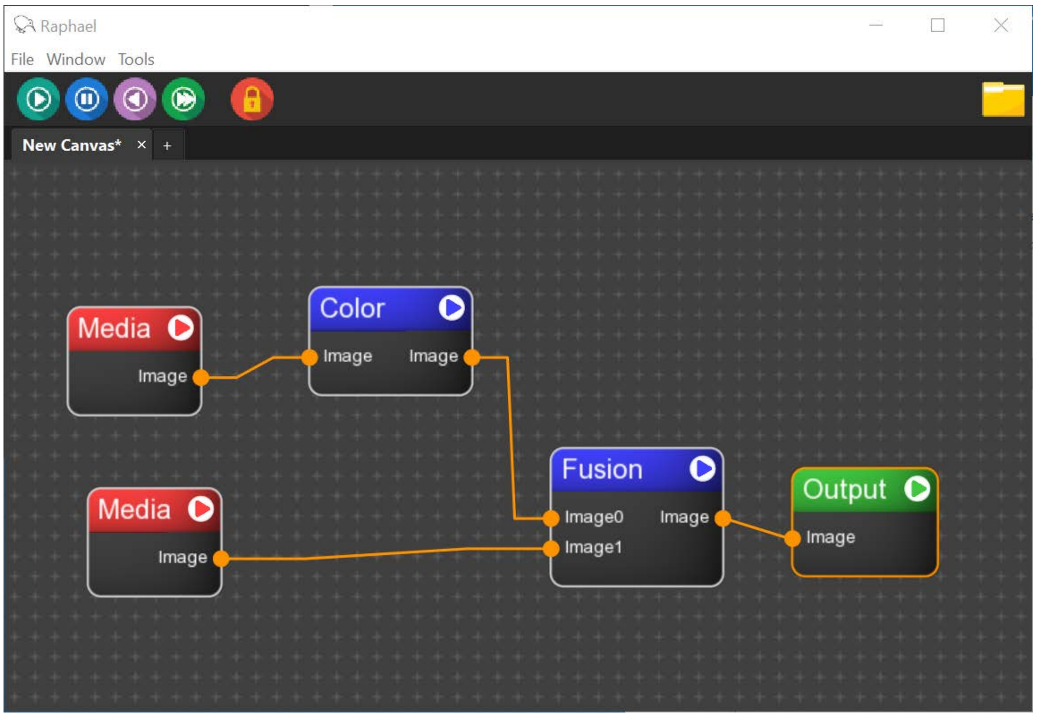


```
<Meta version="0" scree_res="800,600" fps="60" dt="0" blend="0" flags="0"/>
<Node id="1" name="MediaModule" posx="195" posy="255">
        <Field name="name">Media</Field>
        <Struct name="module_data">
                <Field name="Receive_Mouse_Inputs">0</Field>
                <Field name="Receive_Keyboard_Inputs">0</Field>
                <Field name="Start_On_Awake">0</Field>
                <Field name="Stop_On_Sleep">0</Field>
                <Field name="Execute_Once">0</Field>
        </Struct>
        <Field name="Path"><R>Resources\Images\Placeholder.jpg</Field>
        <Field name="play_on_start">1</Field>
        <Field name="loop">0</Field>
        <Field name="volume">50</Field>
        <Field name="flip_x">0</Field>
        <Field name="flip_y">0</Field>
        <Field name="play" type="1"/>
        <Field name="volume" type="1"/>
        <Field name="Output" type="2"/>
        <Field name="AudioData" type="2"/>
</Node>
<Node id="2" name="OutputModule" posx="480" posy="270">
        <Field name="name">Output</Field>
        <Struct name="module_data">
                <Field name="Receive_Mouse_Inputs">1</Field>
                <Field name="Receive_Keyboard_Inputs">1</Field>
```

Although these files can be edited with a text editor (XML format), it is recommended to always save them using the user interface to avoid errors that could render the setup unusable.

3. User Interface (GUI)

The figure shows a view of the main interface (Main Graphic User Interface).

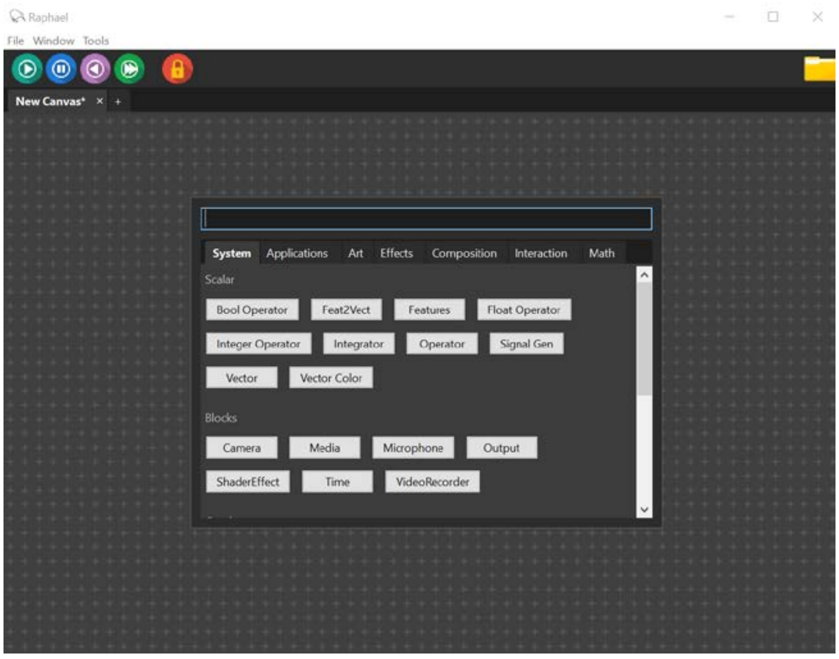


The Main GUI is composed of a central area dedicated to creating setups, a top toolbar dedicated to services, and an icon toolbar dedicated to controlling the program flow.

3.1 Setup Area

Block Insertion

To insert a block, click with the left mouse button in the setup area. The block table will appear as shown in the figure.



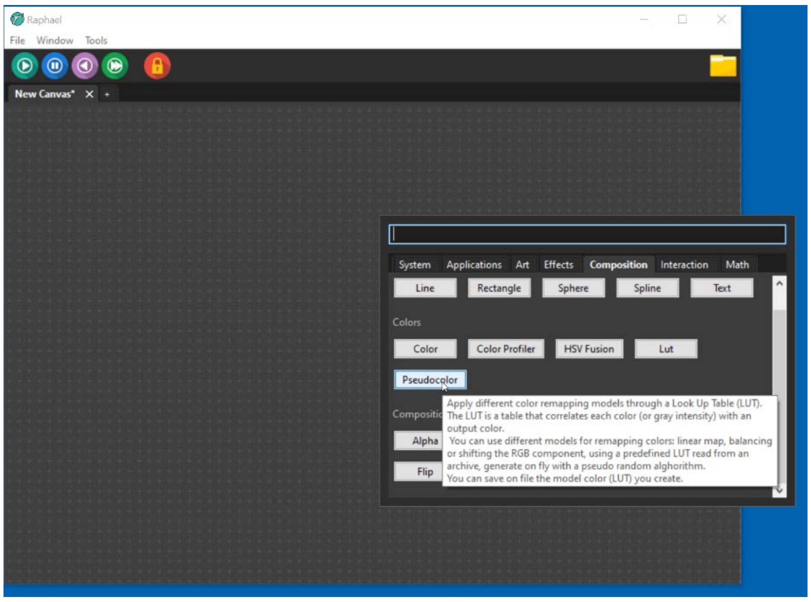
The table is divided into thematic tabs, each containing a set of blocks. By clicking on the selected block, it will be inserted into the setup area. By clicking on the tab name, other blocks can be chosen. If you know the name of the block, you can simply type it in the search area at the top.
For frequently used blocks, shortcuts can be activated. Specifically:

- Left-click + o = Output
- Left-click + m = Media

For inserting files (images, videos, audio clips), you can drag and drop a file directly from the Windows directory into the setup area. This allows for quick insertion of a Media block pre-configured with the file name.

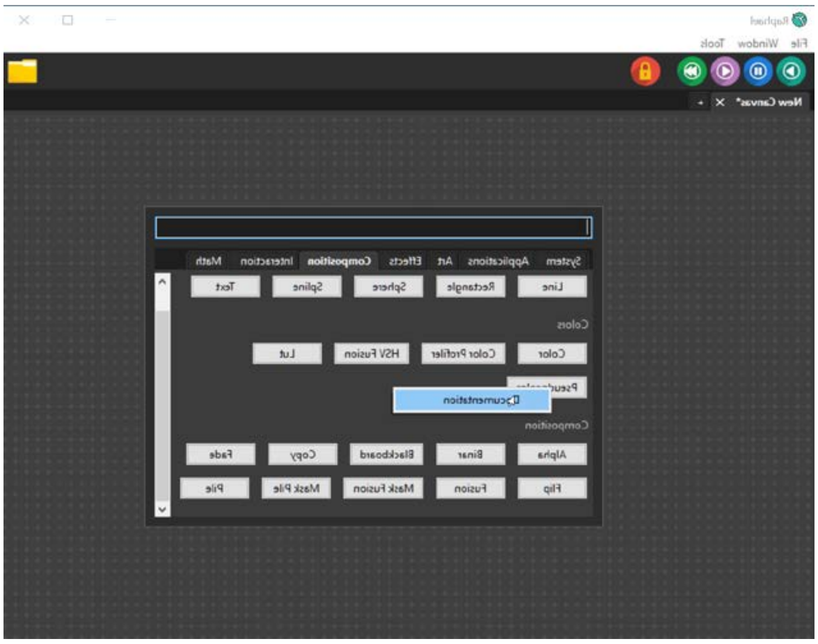
Inline Block Help

To facilitate the identification of the desired block and explore the potential of each block, you can hover the mouse over the block name in the table to display a concise help description of the block's objectives and functionalities.


Raphael
File Window Tools
New Canvas*
System Applications Art Effects Composition Interaction Math
Line Rectangle Sphere Spline Text
Colors
Color Color Profiler HSV Fusion Lut
Pseudocolor
Compositio
Alpha
Flip

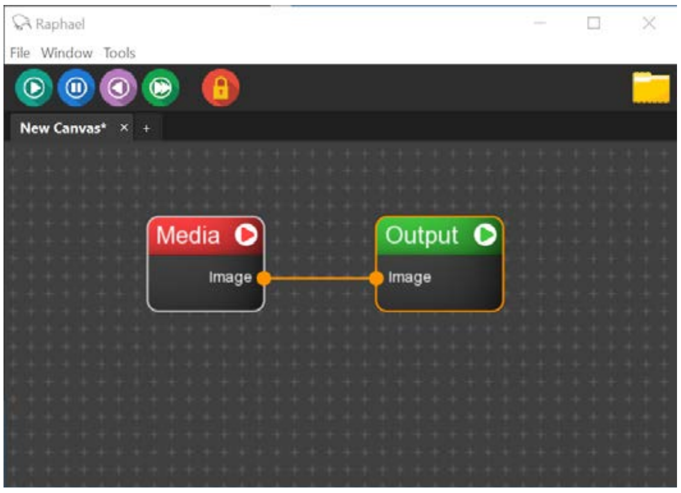
Apply different color remapping models through a Look Up Table (LUT).
The LUT is a table that correlates each color (or gray intensity) with an output color.
You can use different models for remapping colors: linear map, balancing or shifting the RGB component, using a predefined LUT read from an archive, generate on fly with a pseudo random alghorithm.
You can save on file the model color (LUT) you create.

Right-clicking on the selected block also allows you to access the full manual where the block's usage and parameters are documented.


Raphael
File Window Tools
New Canvas*
System Applications Art Effects Composition Interaction Math
Line Rectangle Sphere Spline Text
Colors
Color Color Profiler HSV Fusion Lut
Pseudocolor
Documentation
Composition
Alpha Binar Blackboard Copy Fade
Flip Fusion Mask Fusion Mask Pile Pile

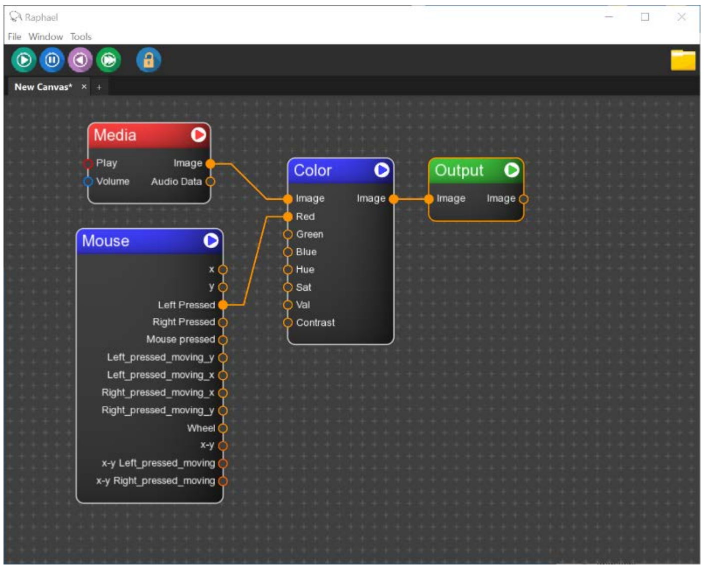
Connecting Blocks

Once two blocks are inserted, they can be connected via a connection arrow. To activate the arrow, click (left mouse button) on the triangle next to the block name and drag the mouse to the name of the next block. This will automatically connect the output of the first block with an input variable of the second block. In the following figure, the image output of the Media block will be connected to the image input of the Output block.

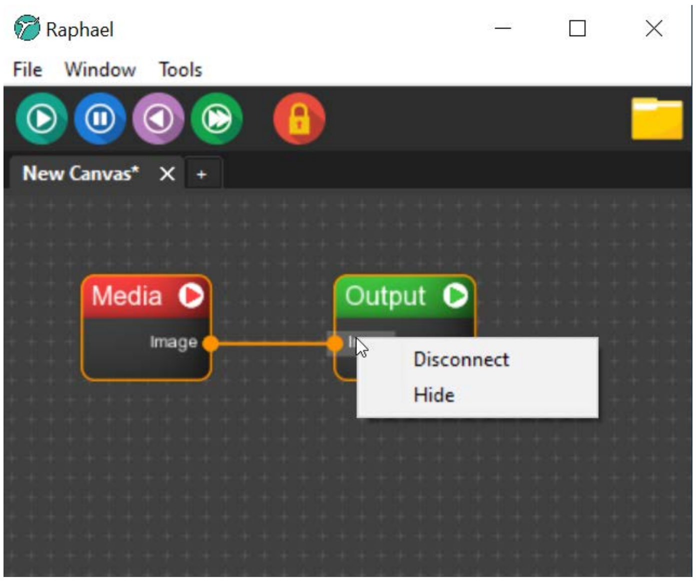


Connecting Parameters

To view all input/output parameters of the blocks, activate the "lock" icon in the icon toolbar. Once the parameters are explicitly displayed, it is possible to connect a specific output parameter of one block with an input parameter of another block (click the mouse on the output parameter dot and drag it to the input parameter dot of the next block). The following images show the parameter display.



In the example shown, right-clicking the mouse allows you to control the “Red” component of the “Color” block.
Parameters can only be connected if they are of the same type.
To disconnect a parameter, right-click on the parameter dot you want to disconnect. A menu with two options will appear: “Disconnect” to disconnect, and “Hide” to avoid tracing the connection line without disconnecting (in case of very complex setups). "Hide" connections can be displayed again by selecting "Show" from the menu.



Moving, Deleting, or Copying Blocks

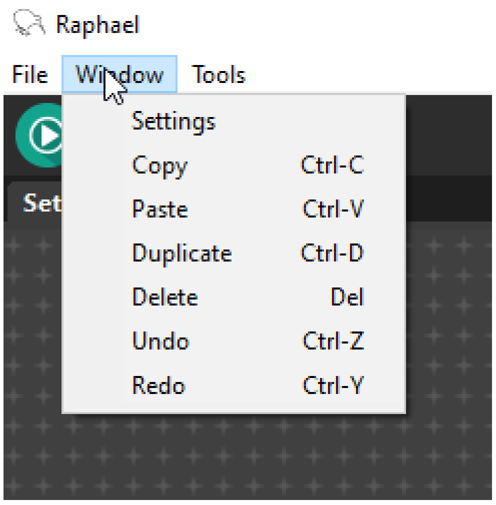
Clicking the left mouse button on a block will select the block you want to work on. The activated block will be highlighted with an orange outline. By clicking and holding the left mouse button on the selected block, you can move it to the desired position.

You can select multiple blocks by clicking and dragging the left mouse button to create a rectangle around the blocks you want to select. Alternatively, you can select multiple blocks by holding down the Ctrl key and left-clicking. This allows you to move multiple blocks at once.

Right-clicking in an area where there are no blocks and dragging the mouse while holding the button will allow you to move all blocks at once to better center them in the setup area.

Finally, Ctrl-C copies the selected blocks to memory, and Ctrl-V pastes them into the same setup.

To remove a block or a group of blocks, right-click on the block to remove. A menu will open with two options: "Delete" to remove the block, or "Open details" to open the block's parameter table.

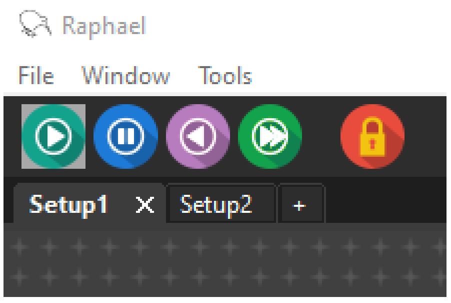


Alternatively, the functions for copying, deleting, duplicating, and pasting blocks can be accessed through the Window menu in the toolbar. This menu also allows you to activate Undo and Redo functions.

3.2 Setup Management

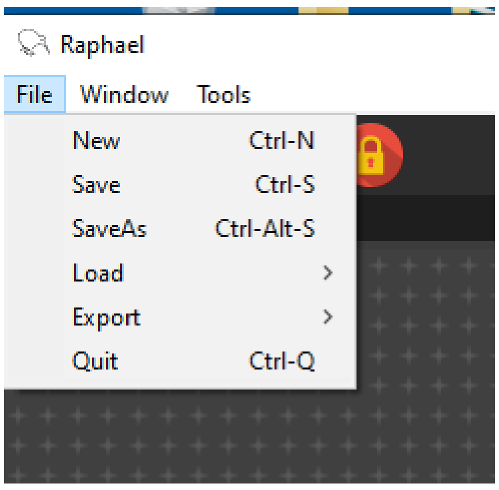
Setup Folders

Raphael allows you to work with multiple setups online. Various setups can be read and processed from different folders in the Setup area. In the example in the next figure, Setup1 is created in the first folder, and Setup2 in the second folder. This feature is very useful for comparing different setups, copying parts (with Ctrl-C, Ctrl-V), or examining a sample setup of the resources used by a particular block to use it in your own setup.



Creating, Saving, and Retrieving a Setup

In the toolbar shown in the figure, by clicking the "File" option, the service menu for creating, saving, and retrieving setups opens.



Below are the meanings of the various options:

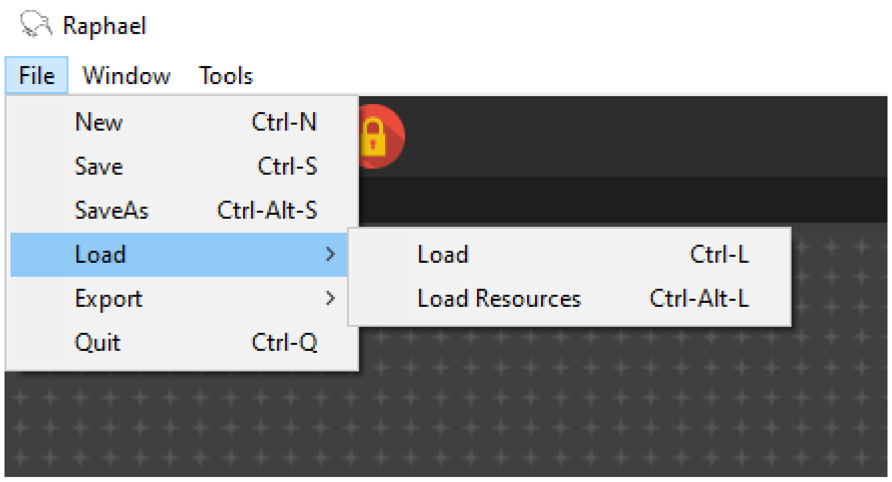
New: creates a new setup starting from an empty area

Save: saves the setup, overwriting the previous version

Save As: saves the setup with a new name and path chosen by the user

Quit: exits the program

- **Load**: loads a setup. There are two options: Load and Load Resources



With the **Load** option, you can choose a previously saved user setup.

Drag and Drop a Setup

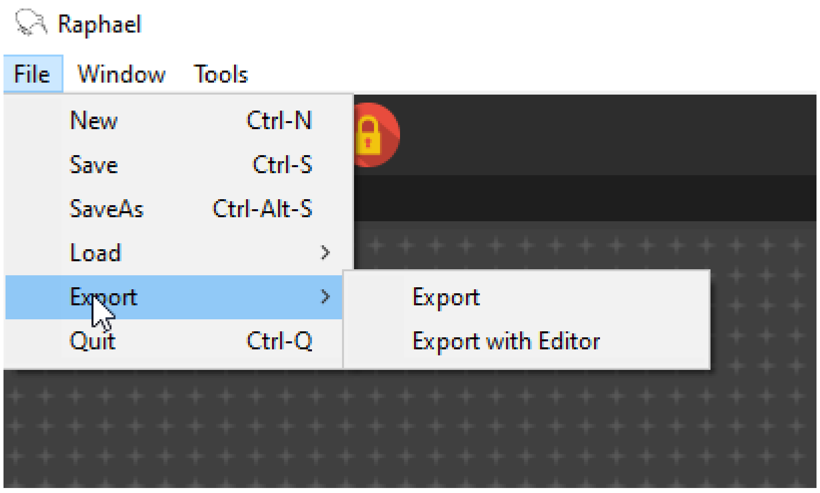
To load a setup file, you can use the drag-and-drop method by dragging a setup file (with the .icf extension) directly from a directory into the Setup area.

Example Setups in the Resources Area

The **Load Resources** option allows the user to access a large archive of example setups, which helps quickly understand how to use a block or a set of blocks. This option lets you access example setups. It is highly recommended not to overwrite a resource setup to avoid losing its integrity. Therefore, if you want to start from a resource setup and modify it for your needs, it is best to use Save As to save it in your own project area.

Exporting an Application

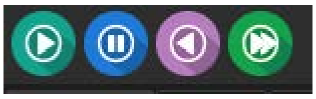
The **Export** option allows you to save not only the setup but also the entire program locked to a specific setup. This feature is useful for generating applications that can be launched standalone. The export includes all resources used by the specific application (images, videos, audio, etc.), an essential copy of Raphael, and the used setup. The export produces a resource folder and a launch file for the application that directly runs the Setup.



The Export function can export the application with no possibility for the user to change block parameters (Export option), or with the possibility for the user to change setup parameters during execution (Run). This is the ideal solution for easily moving an application to different computers.

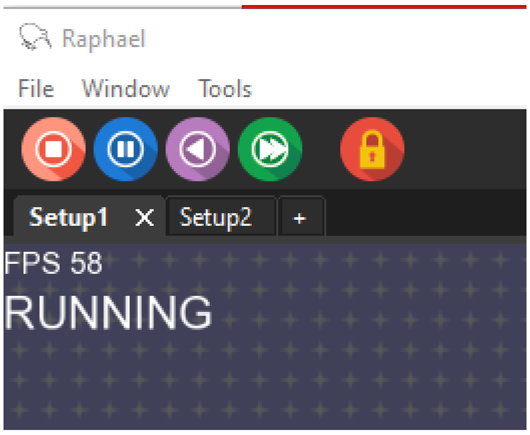
Running a Setup

The icon toolbar includes a series of features for running setups.



- The green triangle ("Play") starts the setup contained in the displayed folder.
- The pause icon ("Pause") pauses the execution.
- The purple triangle ("Restart") restarts the setup, restoring the original values of the setup parameters.
- The double green triangle ("Frame") runs the setup one frame at a time.

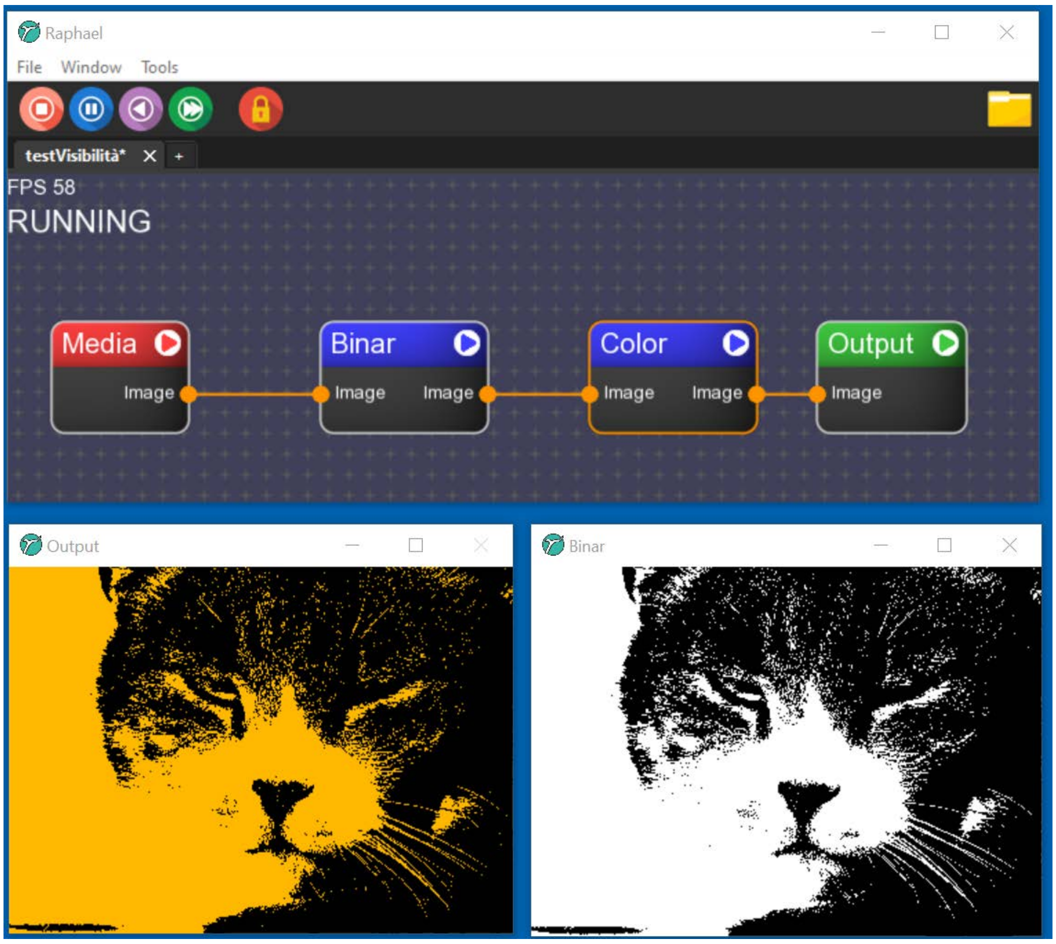
When the setup is running, the Play icon is replaced by a "Stop" icon, which is necessary to exit the execution. The label "RUNNING" and the FPS (Frames per second) will appear to monitor the execution speed in terms of frames per second. Computationally heavy setups may show lower FPS. For real-time applications, it is recommended not to go below 10 FPS.



While the setup is running, it is still possible to change the parameters of enabled blocks in real time, but the setup cannot be saved. To save it, you need to stop the setup and then save it.

Viewing an Intermediate Result

Some setups can be very complex, and it is very useful to check the actual output of a specific block, typically the image resulting from the output of a block. This is possible when the setup is running by clicking on the output dot of a variable.
In the example in the figure, an image (Media block) is first binarized (Binar block) and then colored (Color). Finally, it is sent to an Output block for display on the screen (image on the left).



Clicking on the dot of the image variable exiting from the Binar block opens a window where you can examine the state of the binarized image that will enter the Color block (image on the right).

Running from a Batch File or Command Line

You can run the Raphael program directly from the command line with several options:

```
-h            -> help
-f  setup.icf  -> run Raphael directly with the 'setup.icf' setup file.
-s            -> specifies that the file is a scheduler file (i.e., a sequence of setups).
-p            -> run Raphael directly in play mode.
```

These commands must be executed from the "Binaries" directory of the installation. Therefore, even when launching via a Batch file (*.bat), you must first switch to the correct directory. Below is an example of running from the command line:

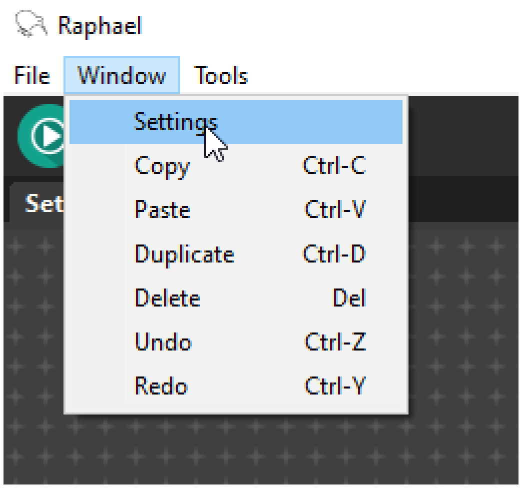
```
cd C:\Program Files\Raphael\Binaries  // moves to the program directory
RaphaelProject.exe -f setup.icf -p  // runs the program with the play option
```

Where "C:\Program Files" represents the area where the program is installed, and you should update it with the correct path of your installation.
This feature is useful for applications that need to run in an unsupervised mode (for example, a setup to be displayed in an exhibition area).
In general, keep in mind that in this solution, the execution is tied to the specific installation of Raphael and the global file paths, meaning the application is dependent on its environment.

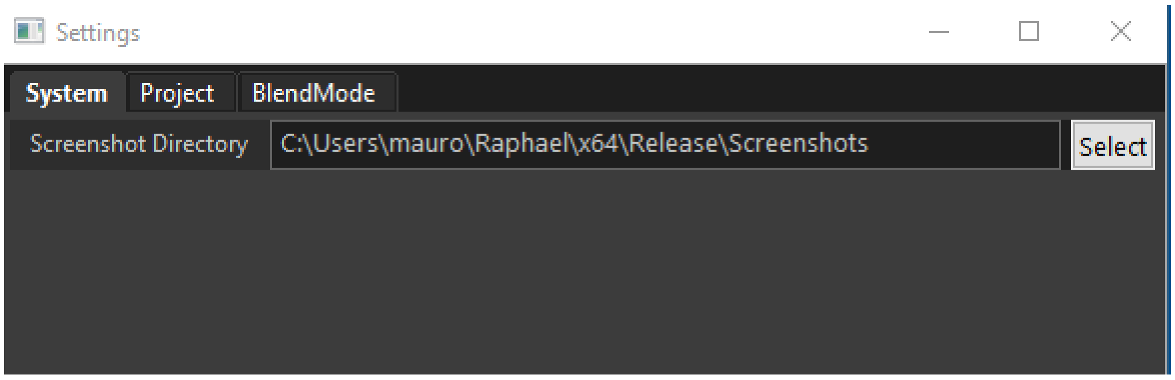
Per non incorrere in problemi di questa natura e permettere la portabilità della applicazione è bene utilizzare l'**export** (vedi sopra) che può essere lanciato agevolmente da computer diversi.

3.3 Settings

From the Window menu, you can access the general Settings.



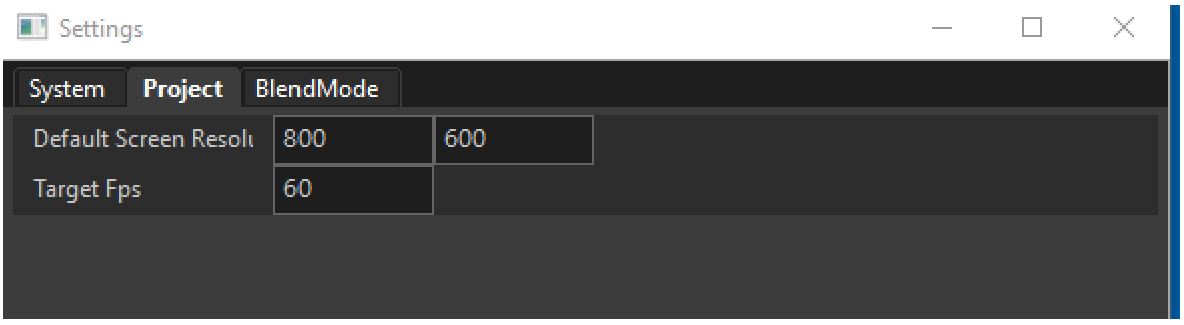
There are three classes of settings. The (**System**) settings refer to parameters related to the program's operation, including, for example, the directory definition where the user wants to save screenshots.



Screenshot Directory C:\Users\mauro\Raphael\x64\Release\Screenshots

A second and third class refer to specific setups and will be saved within their respective setups.

Specifically, the **Project** class contains the default resolution of the project (if not specified by the block that creates the image) and the requested FPS (Frames per Second). The requested FPS refers to the maximum value; for example, if FPS=30, the application will operate at a maximum of 30 fps. Conversely, if the computational load is very high, the actual fps might be lower.

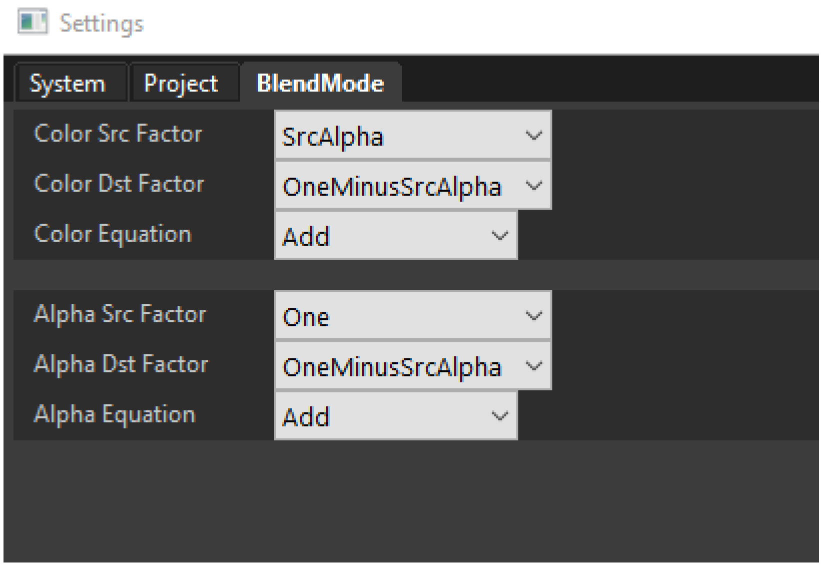

Settings

System | **Project** | BlendMode

Default Screen Resolu: 800 | 600

Target Fps: 60

The **BlendMode** class refers to the graphic model for OpenGL management (refer to the OpenGL manuals to understand the exact meaning of the various parameters - https://learnopengl.com/Advanced-OpenGL/Blending).


Settings

System | Project | **BlendMode**

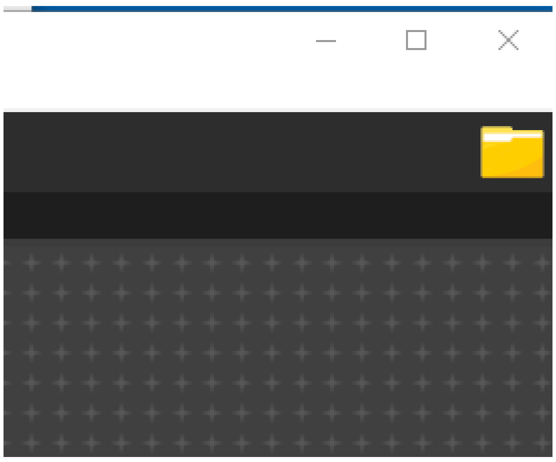
Color Src Factor	SrcAlpha
Color Dst Factor	OneMinusSrcAlpha
Color Equation	Add
Alpha Src Factor	One
Alpha Dst Factor	OneMinusSrcAlpha
Alpha Equation	Add

3.4 Project Organization: The Environment Variable for Portable Projects

The resources used in a setup (images, videos, audio clips) are generally referenced using "global paths" that directly point to directories on the specific computer being used. This makes it complicated to use the project on a different PC or in a different area of the same PC, as all the global file references need to be modified.

To avoid this difficulty, it is recommended to use the "**Environment**" variable, which makes the projects "portable," regardless of the computer used. To do this, you need to create a specific directory for each project where all files and setups will be stored and define this directory as the "Environment" variable. All files created downstream will use local addresses within the project directory instead of the global paths of the PC.

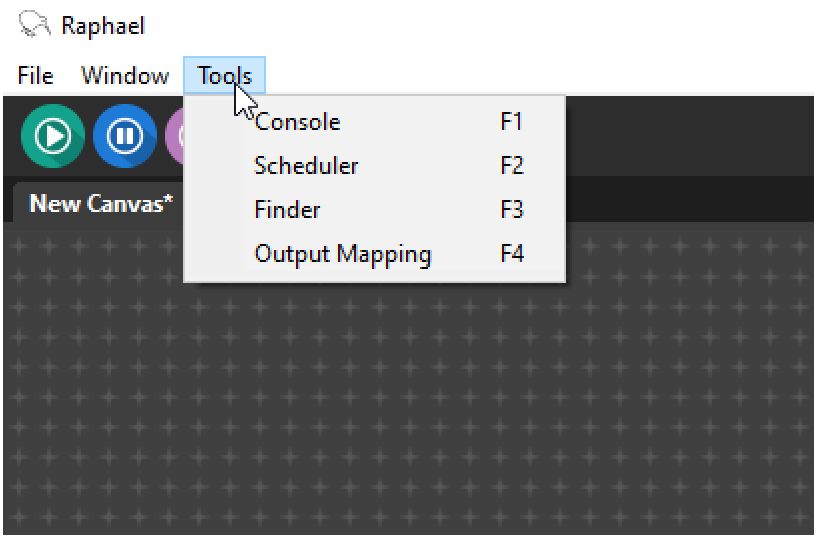
To set the Environment variable, click on the "Folder" icon at the top-right of the main GUI, as shown in the figure.



A window will open, allowing you to select the project directory, which will then be used as the relative reference for the setups.

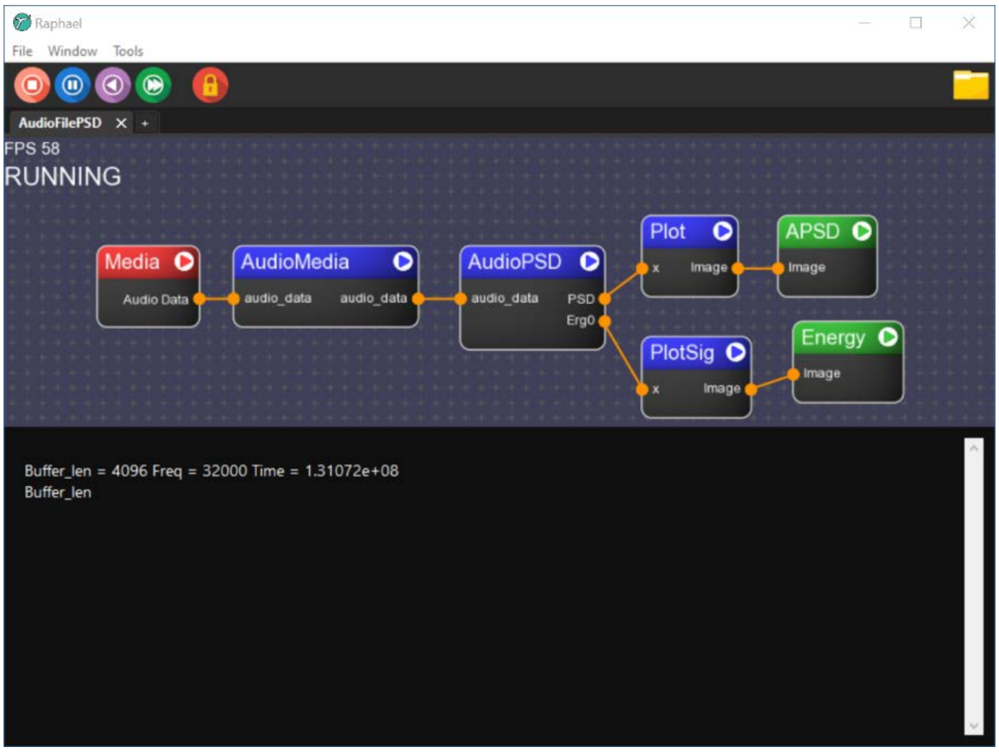
3.5 Tools

In the toolbar at the top, you can access a set of very useful **Tools** through the **Tools** option.



Console

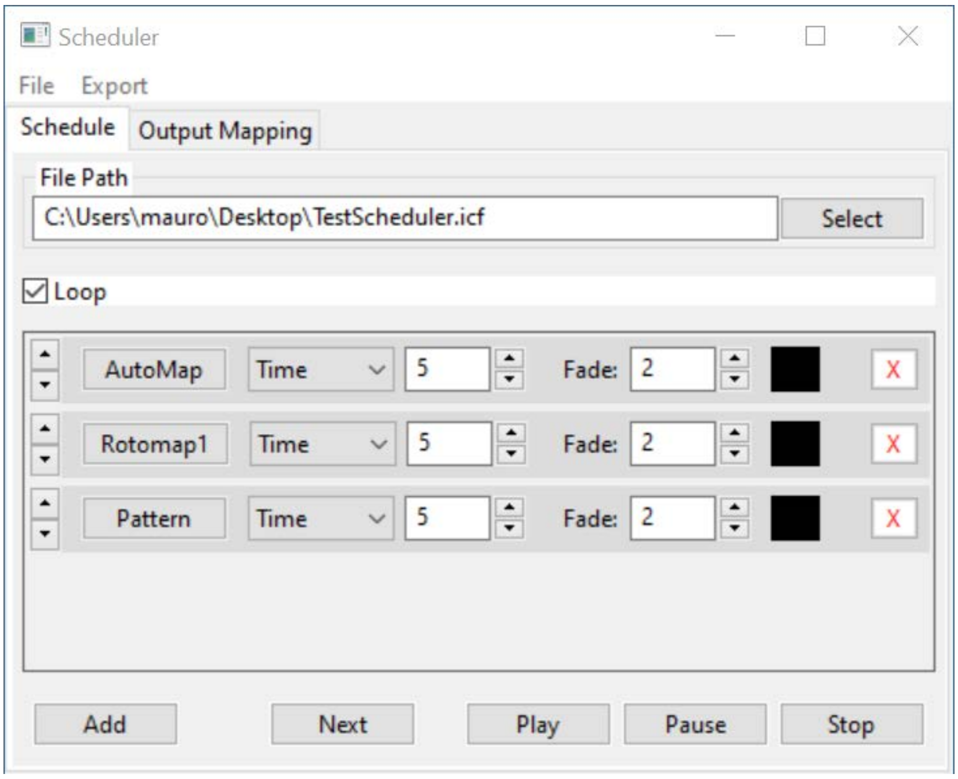
The **Console** tool allows you to open a console where operational messages about various activities performed by the program are displayed, including potential errors. This tool is useful for diagnostics in case of errors in the setups.



```
Buffer_len = 4096 Freq = 32000 Time = 1.31072e+08
Buffer_len
```

Scheduler

The **Scheduler** tool allows you to create a sequence of setups stored in separate files that need to be executed according to a scheme, which can be time-based (you can control the duration of each individual setup) or linked to an internal dynamic of the block (see the Sequence block). You can save and reload the sequence and control the fading time between setups.



C:\Users\mauro\Desktop\TestScheduler.icf

AutoMap	Time	5	Fade:	2	X
Rotomap1	Time	5	Fade:	2	X
Pattern	Time	5	Fade:	2	X

Menu Options:

File: save the sequence file (Save or Save As)

Export: allows you to export the scheduler application as a standalone application (useful for installations in non-supervised exhibition areas).

Output Mapping: in the case of multiple screen outputs, this allows you to map outputs to specific screens as desired (see further details below).

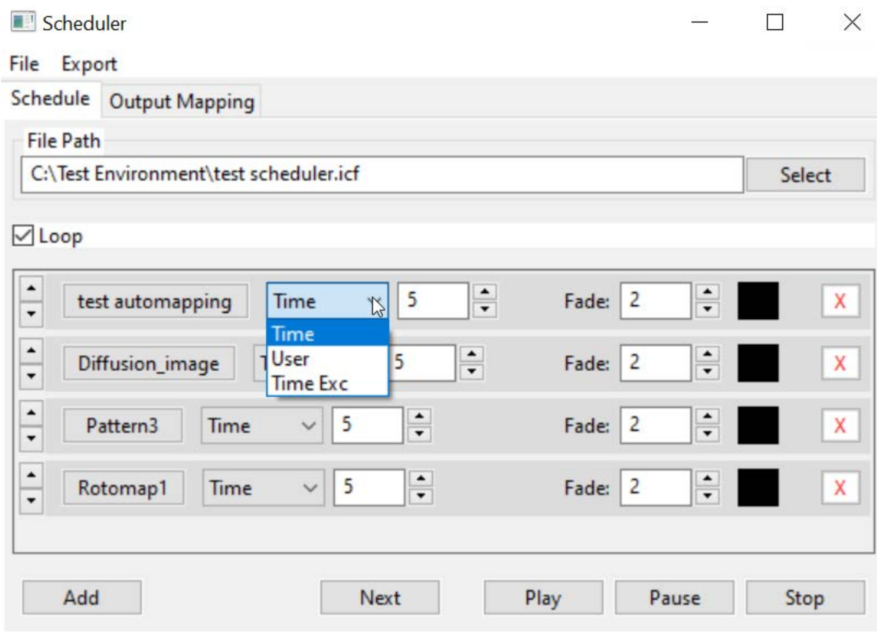
Select: input the name of the selected sequence file (reading a sequence from archive).

Loop: repeats the sequence from the beginning once it has finished.

Setup Row Options:

Up/Down Arrows: move the setup forward or backward in the sequence.

Time: defines the mode and possible duration time for the specific setup.



Scheduler

File Path: C:\Test Environment\test scheduler.icf

[✓] Loop

test automapping	Time	5	Fade: 2	■	X
Diffusion_image	Time	5	Fade: 2	■	X
Pattern3	Time	5	Fade: 2	■	X
Rotomap1	Time	5	Fade: 2	■	X

With the "Time" option, after a set time (input in the next field), the sequence will move to the next setup.

With the "User" option, the transition to the next setup is determined internally within the setup using the "Sequence" block. When this block receives an end setup command via the Scheduler, it moves to the next block. This option is useful when the setup duration is not predefined but depends on internal factors (algorithmic generation, user interaction, etc.).

With the "Time Exc" option, control is similar to the previous one but with a time limit beyond which it moves to the next setup.

Fade: defines the duration of the fade transition to the next setup. The fade happens through a color defined in the field to the right of the fade option.

Remove Box (X): removes the setup from the sequence.

Button Options:

Add: inserts a new setup file into the sequence.

Next: moves to the next setup in the sequence.

Play: starts the sequence. To play the scheduler setup sequence, you must use the play button in the scheduler, not the one in the main GUI.

Pause: pauses the scheduler play.

Stop: stops the sequence play.

Finder

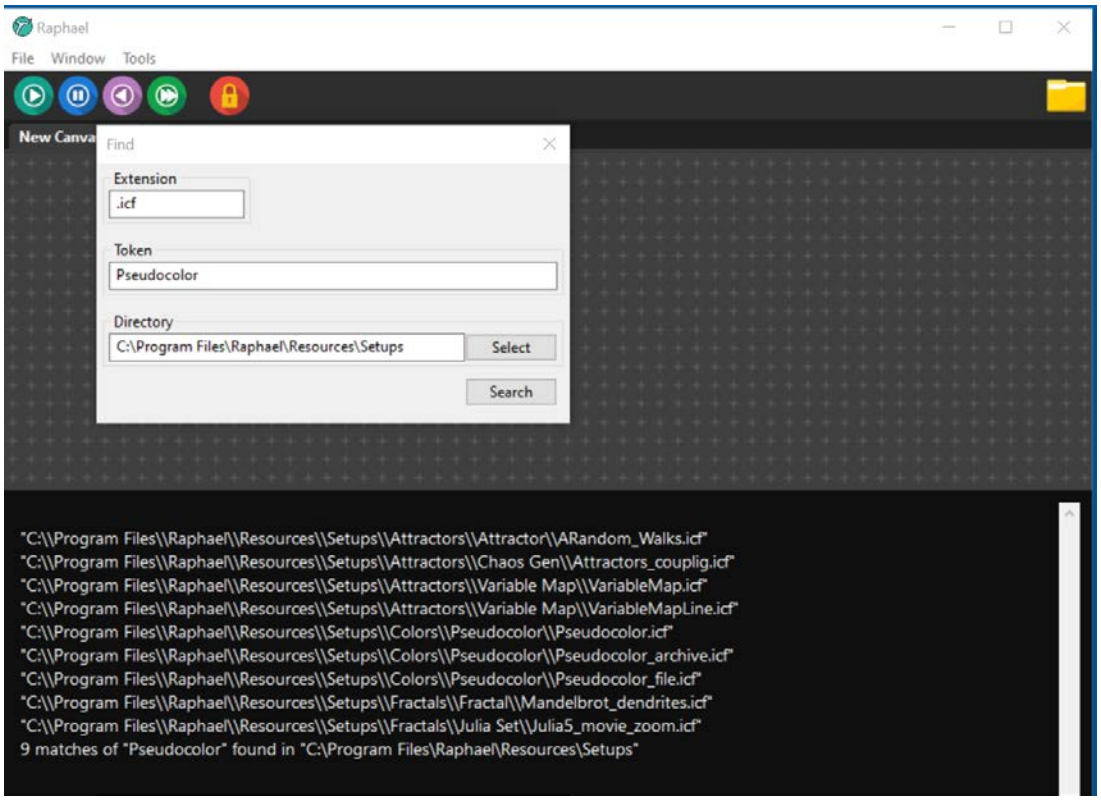
The **Finder** tool allows you to search for all files with a specific extension containing a certain word in a directory. This tool is very useful, for example, for finding all setups that contain a specific image or a certain type of block.

Parameters:

Extension: limits the search to files with a specific extension.

Token: the word being searched for.

Directory: the folder (including subfolders) to search within.

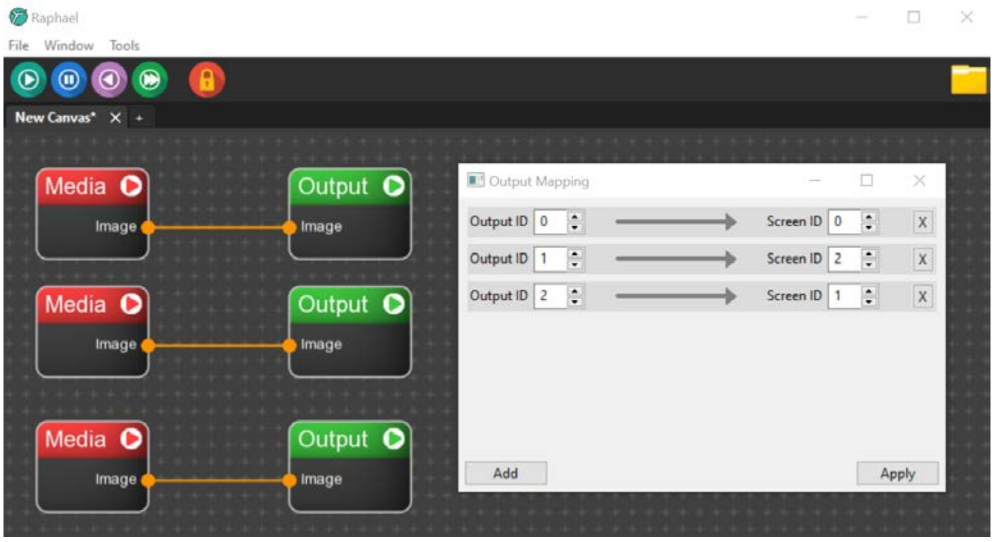


Extension: .icf
Token: Pseudocolor
Directory: C:\Program Files\Raphael\Resources\Setups

```
"C:\\Program Files\\Raphael\\Resources\\Setups\\Attractors\\Attractor\\ARandom_Walks.icf"
"C:\\Program Files\\Raphael\\Resources\\Setups\\Attractors\\Chaos Gen\\Attractors_couplig.icf"
"C:\\Program Files\\Raphael\\Resources\\Setups\\Attractors\\Variable Map\\VariableMap.icf"
"C:\\Program Files\\Raphael\\Resources\\Setups\\Attractors\\Variable Map\\VariableMapLine.icf"
"C:\\Program Files\\Raphael\\Resources\\Setups\\Colors\\Pseudocolor\\Pseudocolor.icf"
"C:\\Program Files\\Raphael\\Resources\\Setups\\Colors\\Pseudocolor\\Pseudocolor_archive.icf"
"C:\\Program Files\\Raphael\\Resources\\Setups\\Colors\\Pseudocolor\\Pseudocolor_file.icf"
"C:\\Program Files\\Raphael\\Resources\\Setups\\Fractals\\Fractal\\Mandelbrot_dendrites.icf"
"C:\\Program Files\\Raphael\\Resources\\Setups\\Fractals\\Julia Set\\Julia5_movie_zoom.icf"
9 matches of "Pseudocolor" found in "C:\Program Files\Raphael\Resources\Setups"
```

In the example shown, all setup files (.icf extension) in the resource setups folder containing the Pseudocolor block are being searched. To view the results, activate the **Console** option (shortcut F1).

The **Output Mapping** tool allows you to control and redirect the outputs to external screens for an application. It is especially useful in multi-screen or immersive applications.



Output ID 0	→	Screen ID 0
Output ID 1	→	Screen ID 2
Output ID 2	→	Screen ID 1

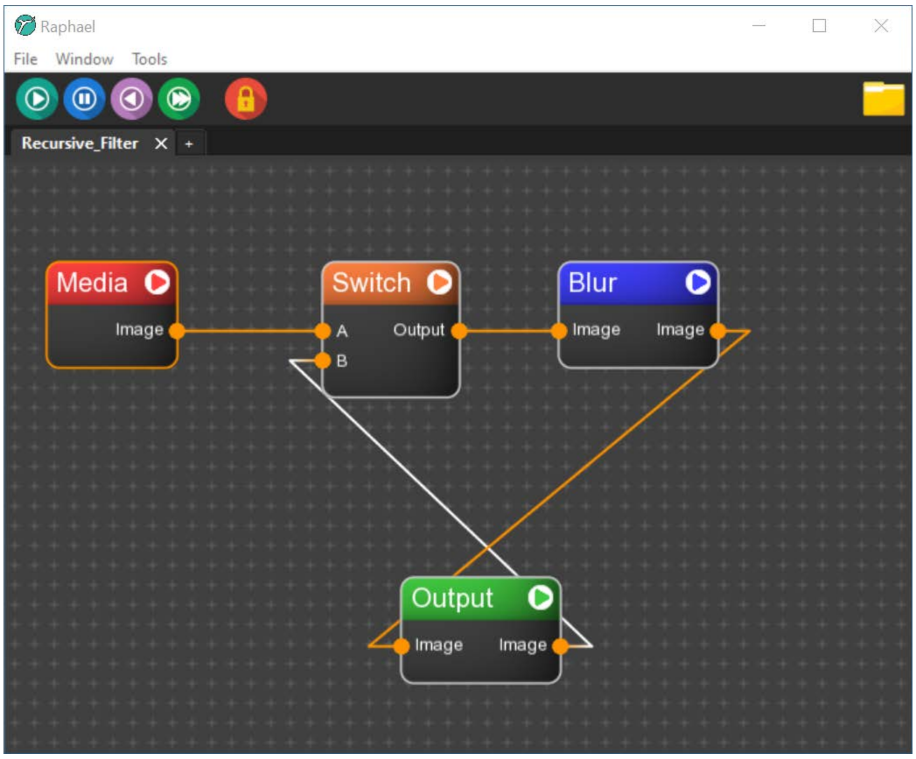
In the example shown, in a three-screen setup, output 0 is sent to screen 0, output 1 to screen 2, and output 2 to screen 1.

3.6 Recursion

The Raphael platform implements the possibility of recursive structures. Recursive structures are setups in which at least one input of a block within the setup receives data processed in the previous frame, particularly gathered from the outputs.

Recursion opens up many artistic applications, expanding the aesthetic and expressive potential of the platform.

The following shows a typical recursive structure. It involves recursively applying an image filter (Blur), which will be discussed in subsequent chapters.



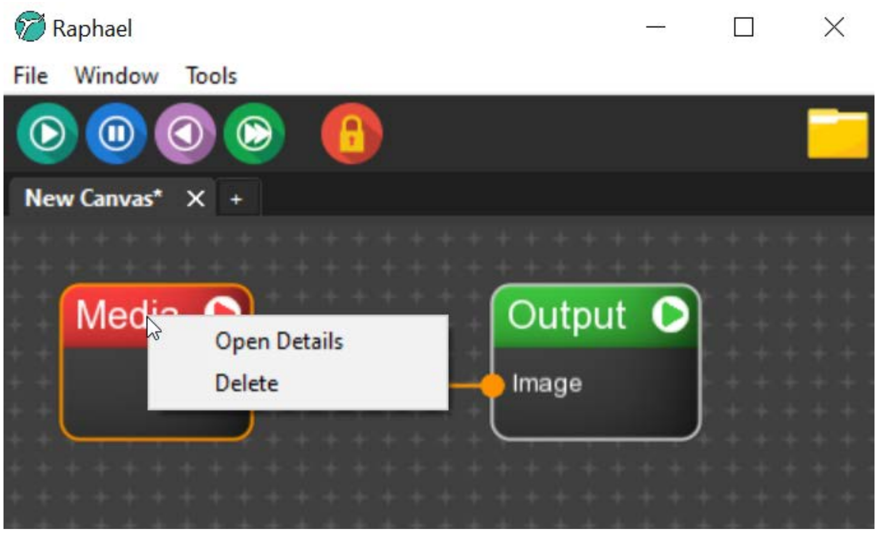
In the example shown, the Switch block is set to Once. Therefore, in the first cycle, it will take the image from the Imageblock, and in subsequent cycles, it will transmit the output's result. In this way, the filter is applied indefinitely to the original image.

The white color of the connection indicates that the image transfer will occur in the next cycle.

In complex setups, to reduce graphical complexity, you can use the Hide mode to keep recursive connections invisible.

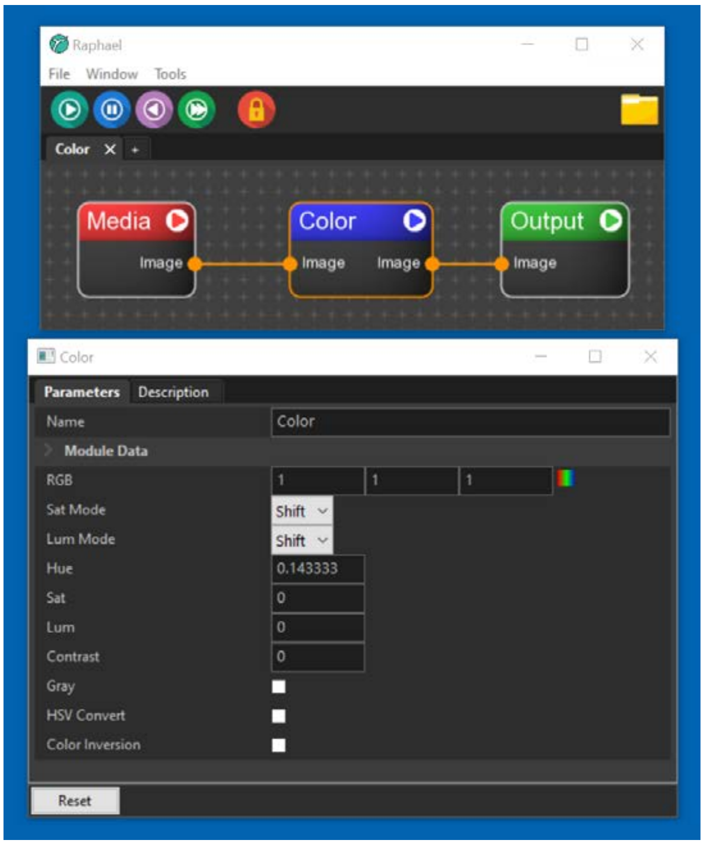
3.7 The GUI Details for Block Parameters

The GUI Details window is used to set the control parameter values for a specific block. It can be activated in two ways. The first is by double-clicking the left mouse button on the block's title in the Setup Area. The second is by right-clicking the block and selecting the Open Details option.



Block Parameters

Each block has its own GUI Details that describe its specific parameters. Some parameters require only a flag, others a label (e.g., a file name), others an optional choice, others a numeric code or a parameter vector.

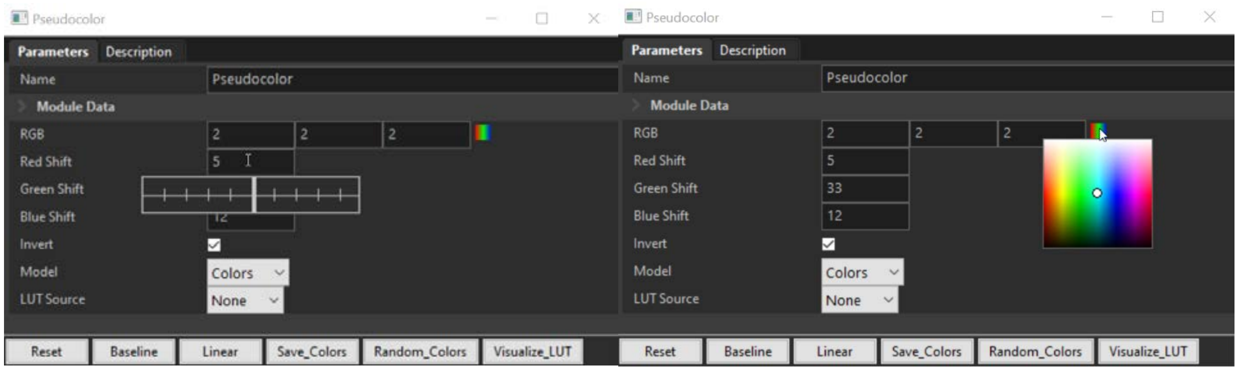


Name	Color
Module Data	
RGB	1 1 1
Sat Mode	Shift
Lum Mode	Shift
Hue	0.143333
Sat	0
Lum	0
Contrast	0
Gray	☐
HSV Convert	☐
Color Inversion	☐

In the GUI Details, there may be different models to set parameters depending on the type of parameter:

Buttons: (at the bottom) activate specific block functions. Usually, there is at least a Reset function to restore the default parameters (see next figure).

Sliders: allow quick setting of the parameter value either through number input or using the slider. If the parameter has multiple values (2 for xy coordinates, 3 or 4 for colors...), multiple sliders are presented in the same row. If it's a color, an option allows selecting the color itself.

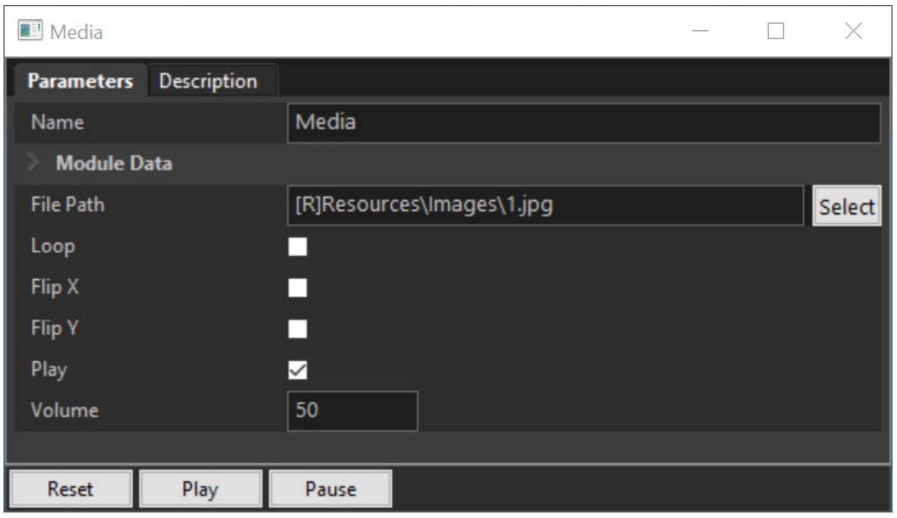


Name	Pseudocolor
Module Data	
RGB	2 2 2
Red Shift	5
Green Shift	33
Blue Shift	12
Invert	✓
Model	Colors
LUT Source	None

Checkboxes: refer to boolean variables (on/off). You can activate or deactivate them.

Menus: allow the selection between different options of an enumerated variable. Each option is identified by a label.

Strings: allow you to identify a file or directory on the computer. You can search for the file or directory in the PC using a dialog window activated with the Select button next to the file name input field.



Name	Media
Module Data	
File Path	[R]Resources\Images\1.jpg
Loop	☐
Flip X	☐
Flip Y	☐
Play	✓
Volume	50

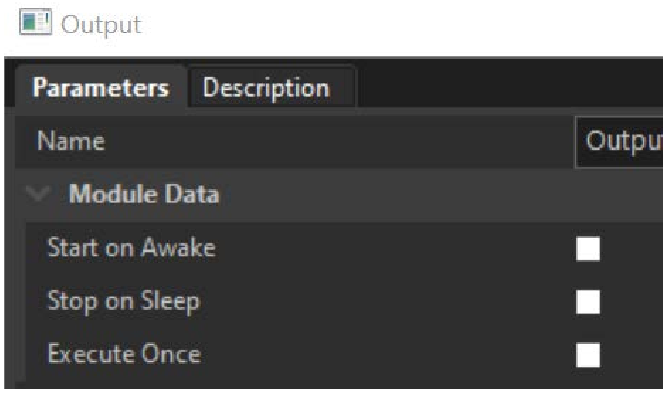
Module Data

Each block features a specific way of managing the start/play for advanced applications. In particular:

Start on Awake: starts the block when the program flow line is active.

Stop on Sleep: stops the block when the program flow line becomes inactive.

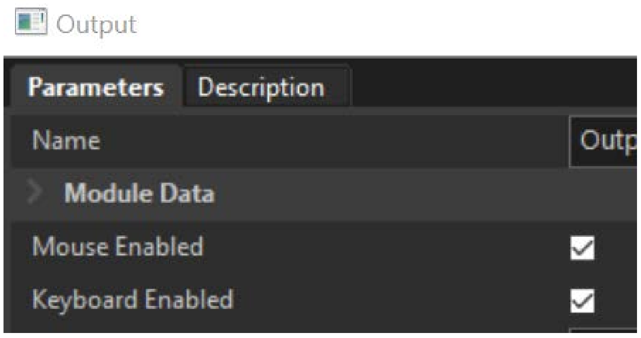
Execute Once: executes the block only once.



Name	Outpu
Module Data	
Start on Awake	☐
Stop on Sleep	☐
Execute Once	☐

Mouse and Keyboard Interaction Management

Each block allows separate management of mouse or keyboard interactions. This is important because it is possible that you want the mouse and/or keyboard to be active on certain blocks in the setup and not on others. By using the relevant flags, you can control the effects on each specific block.

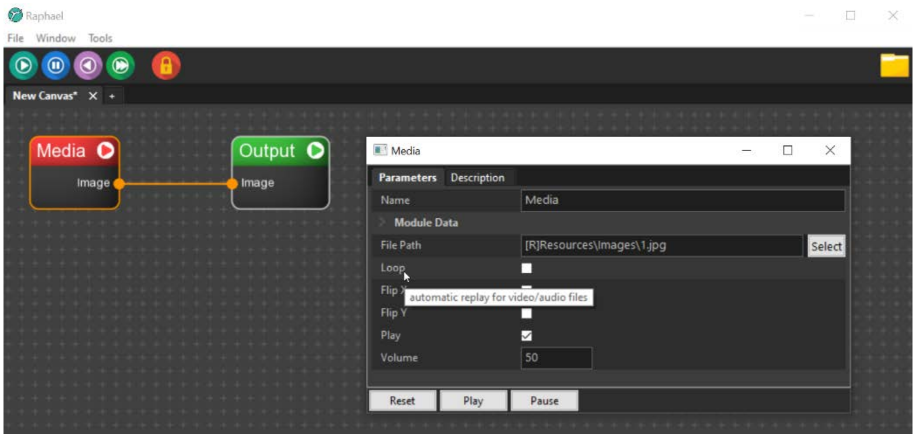


Name	Outp
Module Data	
Mouse Enabled	✓
Keyboard Enabled	✓

For blocks that don't require mouse/keyboard interaction, the flags won't be present.

In-line Help in GUI Details

Hovering the mouse over the label identifying the parameter will show a brief help explaining the function of the selected parameter within the context of the block it belongs to.



Name	Media
Module Data	
File Path	[R]Resources\Images\1.jpg
Loop	☐
Flip X	
Flip Y	☐
Play	✓
Volume	50

automatic replay for video/audio files

Clicking on the Description folder (top left) provides specific information about the nature, functionality, and documentation of the block. The description includes specific details, such as:

Creator: name(s) of the author(s) and/or developer(s) of the module.

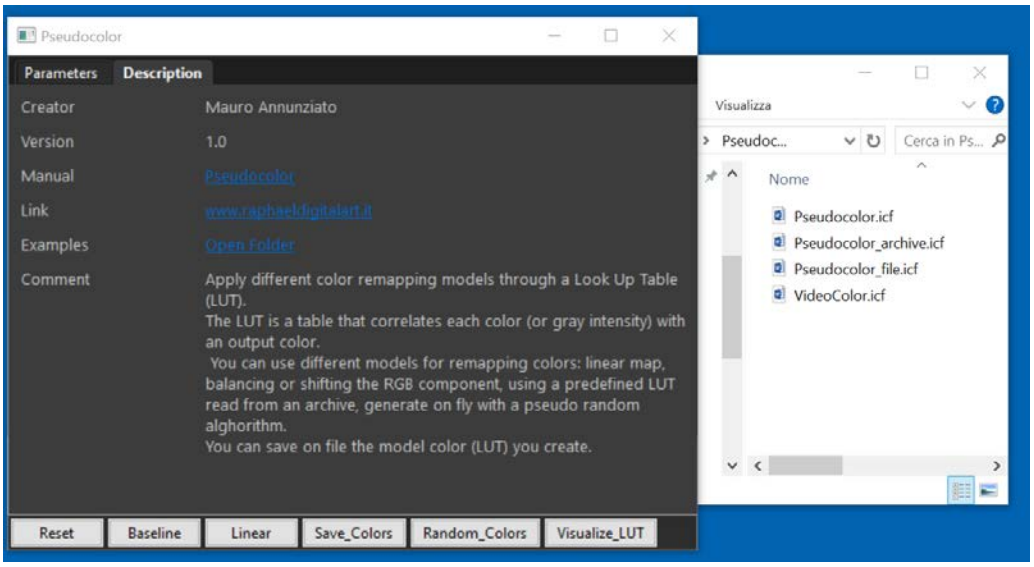
Version: version number.

Manual: link to the block's manual.

Link: link to the Raphael website.

Comment: help on the objectives, operation, and parameters of the block.

Examples: the link opens the **Resources** area, which contains example setups using that specific module (see figure).



Creator	Mauro Annunziato
Version	1.0
Manual	Pseudocolor
Link	www.raphaeldigitalart.it
Examples	Open Folder
Comment	Apply different color remapping models through a Look Up Table (LUT). The LUT is a table that correlates each color (or gray intensity) with an output color. You can use different models for remapping colors: linear map, balancing or shifting the RGB component, using a predefined LUT read from an archive, generate on fly with a pseudo random alghorithm. You can save on file the model color (LUT) you create.

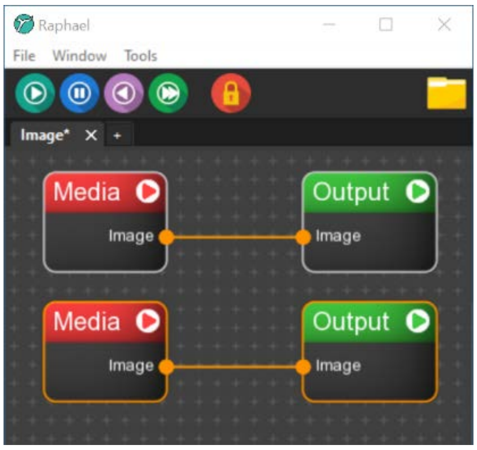
Pseudocolor.icf
Pseudocolor_archive.icf
Pseudocolor_file.icf
VideoColor.icf

For operational purposes, it's convenient to use a second folder in the **Setup Area** of the main GUI to place the example setup, allowing you to study its usage, read the parameters, and possibly copy them into the primary folder where you plan to build your own setup.

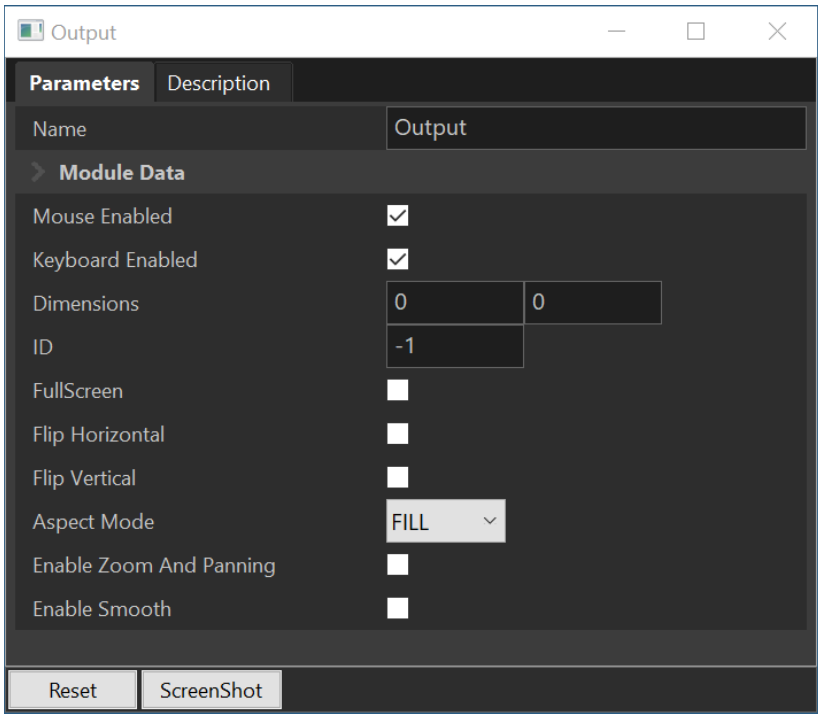
4 Main System Blocks: Output, Media, Camera

4.1 Output

A setup can have multiple outputs to send to multiple screens.



In the following figure, the GUI Details of the Output block are shown.



Name: Output

Module Data

Mouse Enabled: ✓
Keyboard Enabled: ✓
Dimensions: 0 0
ID: -1
FullScreen: ☐
Flip Horizontal: ☐
Flip Vertical: ☐
Aspect Mode: FILL
Enable Zoom And Panning: ☐
Enable Smooth: ☐

Parameters:

- **Dimensions:** Controls the size of the window in pixels. If the dimensions are 0,0, default dimensions are chosen that maintain the aspect ratio of the incoming image. It should be clarified that these dimensions refer to the display window size, not the input texture, which is not altered.
- **ID:** The identifying number of the output (used for Output Mapping). If set to -1, it is automatically chosen.
- **FullScreen:** Toggle for fullscreen.
- **Flip Horizontal:** Flips the image horizontally.
- **Flip Vertical:** Flips the image vertically.
- **Aspect Mode:** Controls the aspect ratio of the image and the ratio between the display window and the incoming image. There are 4 options:
  - **FILL:** Inserts the entire incoming image into the display window. The proportions are scaled according to the window.
  - **HEIGHT:** Inserts the image while maintaining the original proportions but making the height of the image match the window height.

  - **WIDTH:** Inserts the image while maintaining the original proportions but making the width of the image match the window width.
  - **NONE:** Inserts the image while maintaining both the proportions and the original resolution. Therefore, it is possible that the window shows only a part of the image when the incoming image resolution is higher than the window resolution.
- **Enable Zoom and Panning:** Toggle to activate zooming and panning within the display window. If enabled, the mouse scroll wheel controls the zoom, while dragging the mouse with the left button pressed controls the pan.
- **Enable Smooth:** If enabled, it applies image smoothing. Not recommended for applications requiring high computational power.

Among the functional buttons, besides Reset, there is the **ScreenShot** button, which allows saving the image currently on the output in the resolution defined in "Dimensions" to a file.

The block takes an image as input and provides an image as output.

4.2 Media

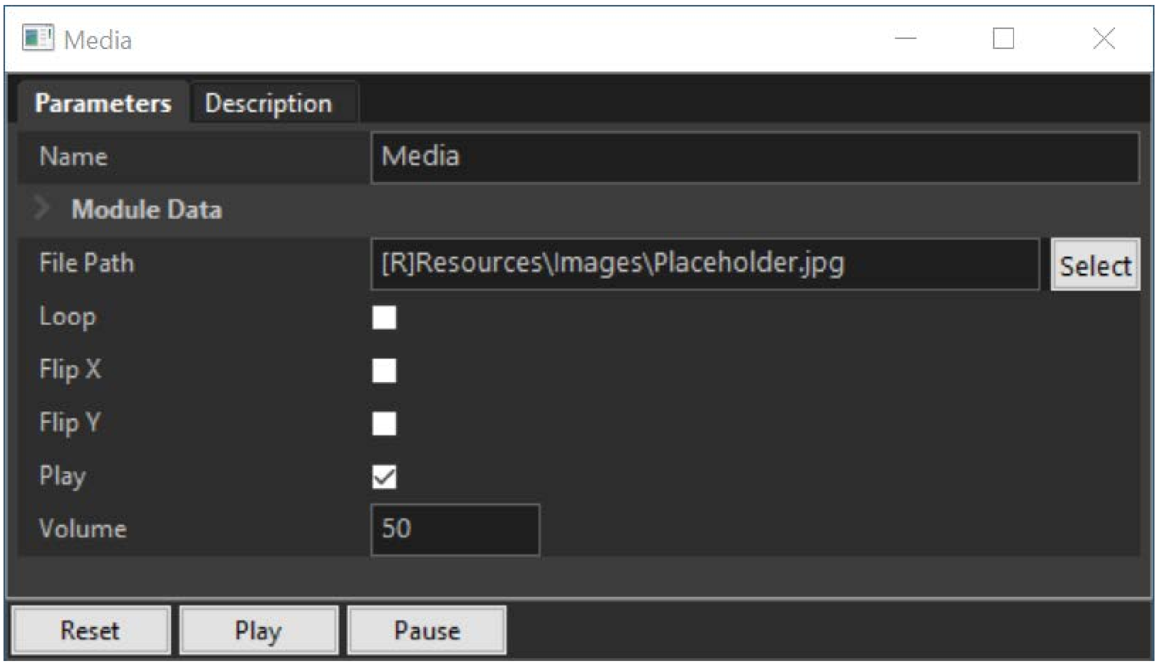
The **Media** block is used to access files in storage, such as images, video clips, and audio clips.

The block automatically recognizes the file type based on the file extension.

Options:

- **File Path:** The file name to insert.
- **Loop:** For video or audio files, allows restarting playback at the end.
- **FlipX:** Flips the image horizontally.
- **FlipY:** Flips the image vertically.
- **Play:** Plays the clip on start for video or audio files.
- **Volume:** Controls the volume (0-100).

The **Select** button opens a dialog window to search for the file in the computer's directories. The **Play** button starts and stops the clip, and the **Pause** button pauses it.



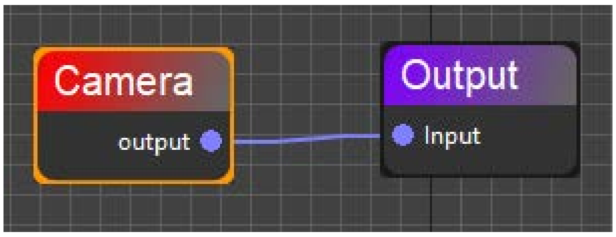
Name	Media
File Path	[R]Resources\Images\Placeholder.jpg
Loop	☐
Flip X	☐
Flip Y	☐
Play	✓
Volume	50

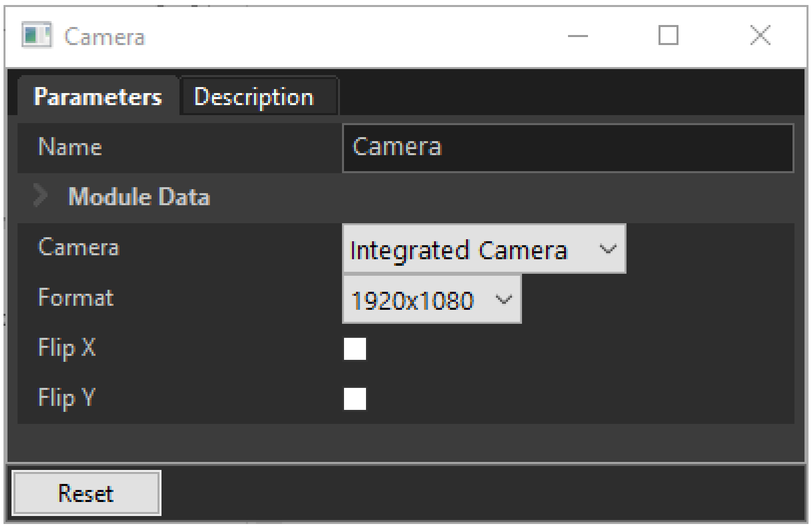
The block can take a **Play** command (boolean) or the **Volume** (integer from 0 to 100) as input. It provides image streaming (**image**) and audio data streaming (**Audio Data**) as output.

Finally, drag-and-drop functionality can be used by dragging a file (image, video, audio) directly from a directory into the Media block to insert files of this type.

4.3 Camera

The **Camera** block captures images from an acquisition channel (camera, webcam, etc.) and makes them available to subsequent blocks.





Name: Camera
Module Data
Camera: Integrated Camera
Format: 1920x1080
Flip X: ☐
Flip Y: ☐

Camera Control Parameters:

- **Camera:** Selects one of the devices recognized by the operating system (Windows or OS).
- **Format:** For devices that support multiple resolutions, the menu allows you to select the desired resolution.
- **FlipX:** Flips the image horizontally.
- **FlipY:** Flips the image vertically.

The block provides streaming output of the texture corresponding to the captured image (image).

Use the **Camera Control** block to adjust the capture area, zoom, panning, flip, grayscale conversion, gain, contrast, binarization, and inversion.