

Introduction

Like in scientific publishing, levels are peer-reviewed, sent back for possible revisions, and once accepted, packaged into magazine issues that I hope to release on a regular basis. To submit a level, you will need to purchase the Flaw Forge Editor (19.99€, one-time purchase). This may seem a little expensive at first, but I hope you will enjoy the Flaw Forge concept and its creative freedom enough to want to pursue the adventure. Note that for each accepted level, the author is credited at the bottom of the Player. Accumulated recognition will be further rewarded: the best contributors will gain early access to future issues or may even be invited to join the editorial board.

Please keep in mind that I am an independent developer who spent nearly a full year building these apps and creating their initial content. I apologize to iOS-only users — the Editor is currently available on macOS only.

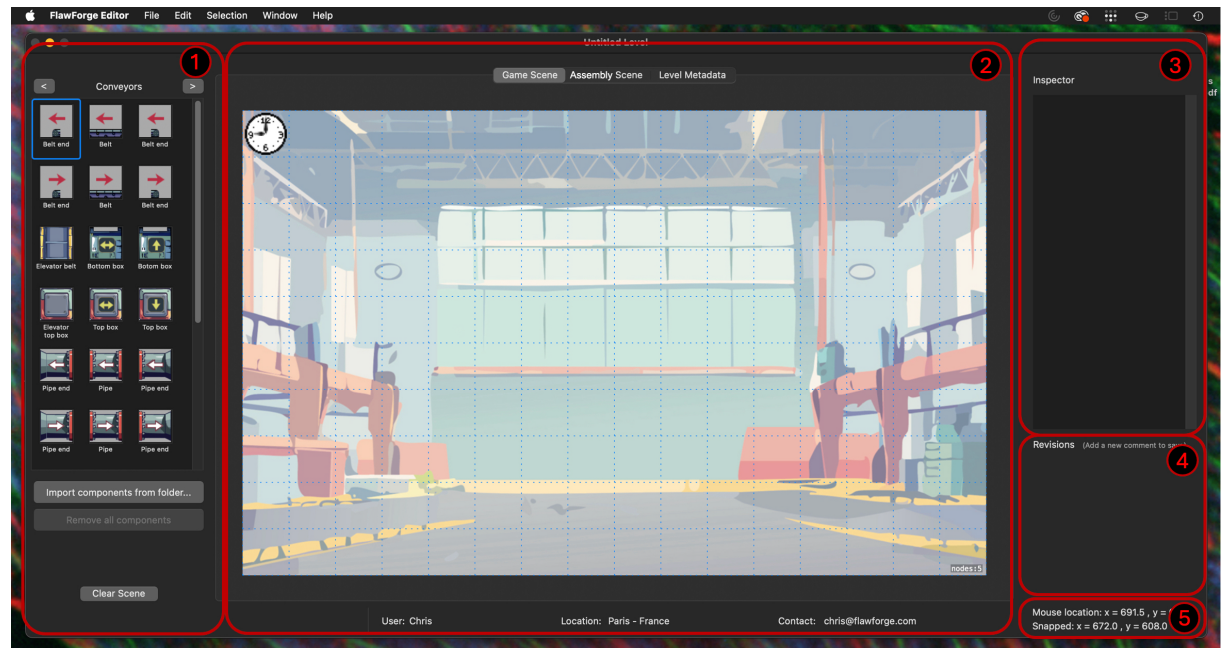
The Editor

A- First Launch: You will be asked only once to fill your name, location and contact address. This is only for recognition purpose, so that I, other players and level developers know just a little bit who you are and may contact you directly. Of course, this information is optional, and you can use a pseudo and/or a junk mail if you want to stay anonymous. If something is lost or if you want to update the author information, just call the Preferences item in the Flaw Forge Menu.

B- The Window Zones:

- 1- The Module & Components zone. Displays a full catalogue of available modules grouped into logic sets. *Refer to the module list in the Flaw Forge Player manual for each module function.* It also allows you to choose a set of components i.e. the mobile objects that are injected into the assembly line and that you should assemble into a final product.
- 2- The Scene and Metadata editors where you will set the factory assembly line, the simplified assembly guide and all the data like the component entry order, the quota, the truck capacity etc...
- 3- The Sprite Inspector: it will display all the properties of the sprite selected. Mostly for debugging purpose.
- 4- The revision data: Add revision information before you save a level. Keep track of all the previous changes.

5- The mouse location data, also for debugging help.

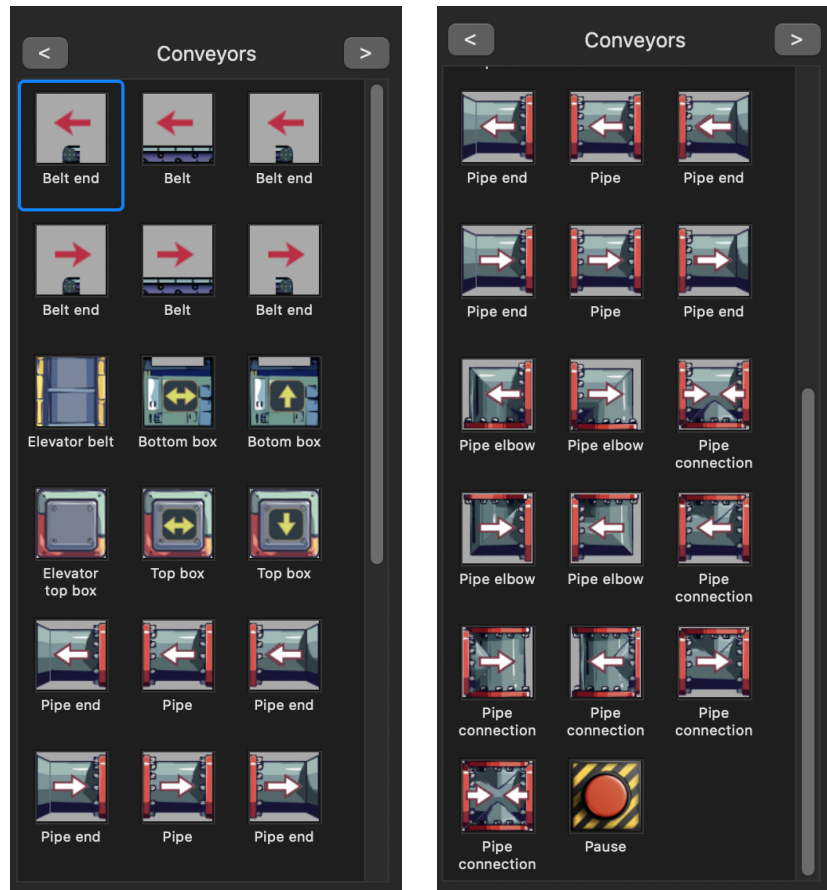


1- The Module & Components zone

The Conveyor category:

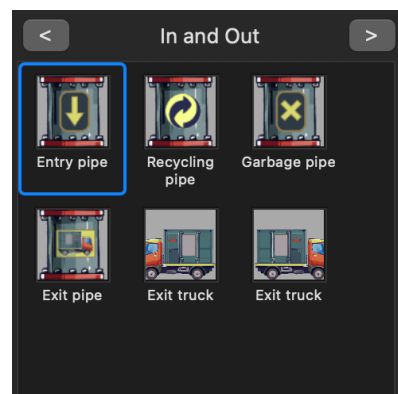
There you find belts and pipes that go in the indicated direction, the Elevator Module and Exit Box, The Bottom and Falling Boxes, The Emergency Stop button.

Note that you should chose the state of the box to start the level with, either horizontal or vertical exit direction.



The In and Out category:

Here are the Entry Pipe and different Truck exits. One of each is mandatory in a new level. You will also find the Recycler and the Trash here.





The Storage and Transformers set:

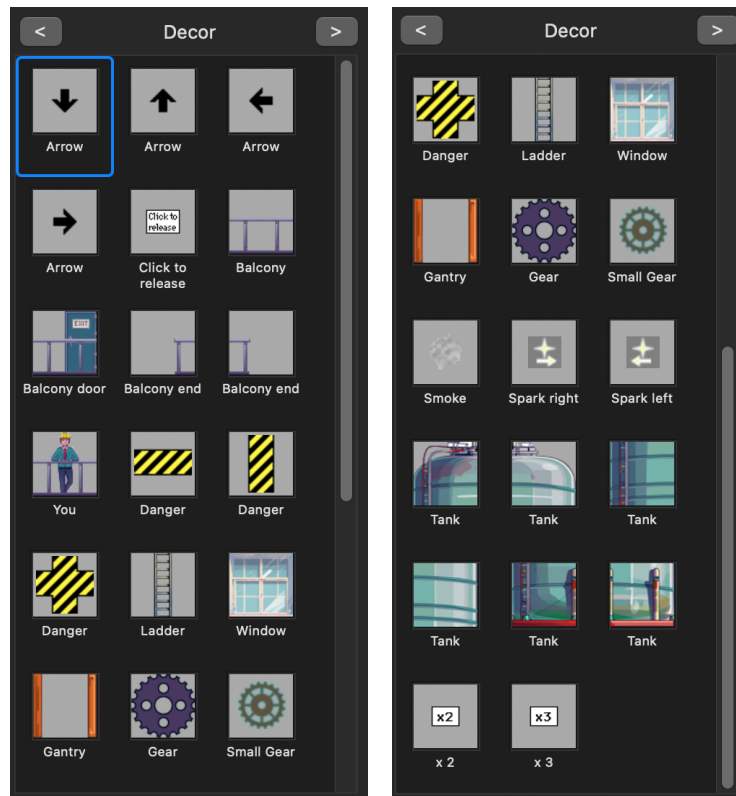
There you can pick Single Storage units and Control Switches as well as Oven and Washer modules. They all come in two states Close or Open / On or Off. Chose the one you want your level to start with.

You can also chose a Filler, or a Wrapper or Multiple Accumulators for components or for liquid material.

Notice that the liquid accumulator was absent of the original "Factory" game. It was added here to provide more control component liquefaction and distribution. It works like the Accumulator for solid components.

The Decor Modules:

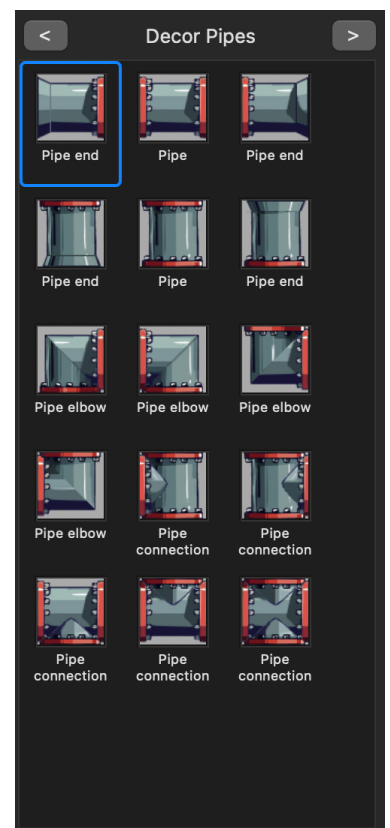
You can use them to personalize your level and make it more fancy or more self-explanatory. Arrows will immediately guide the player to indicate where components should be stored or dropped. Click To Release and x2 or x3 stickers indicate how to proceed with Multiple and with Liquid Accumulators.

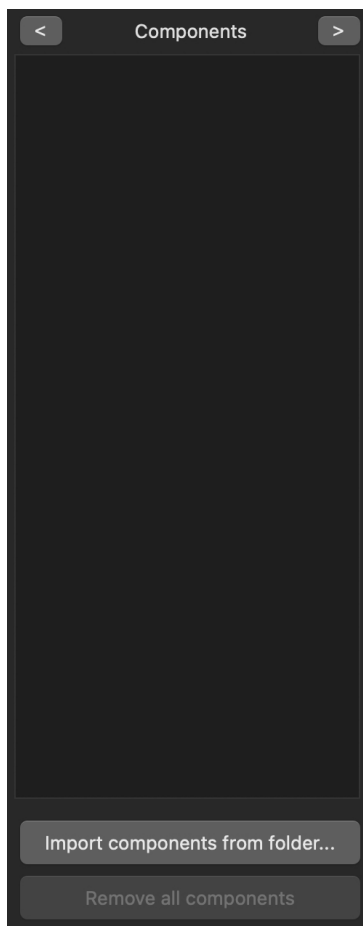


The Decor Pipes:

They allow you to place pipes that do not have conveyor functions.

Note that the vertical pipes here are just masks for falling components, but they can also guide them vertically to prevent collision with a crossing belt.

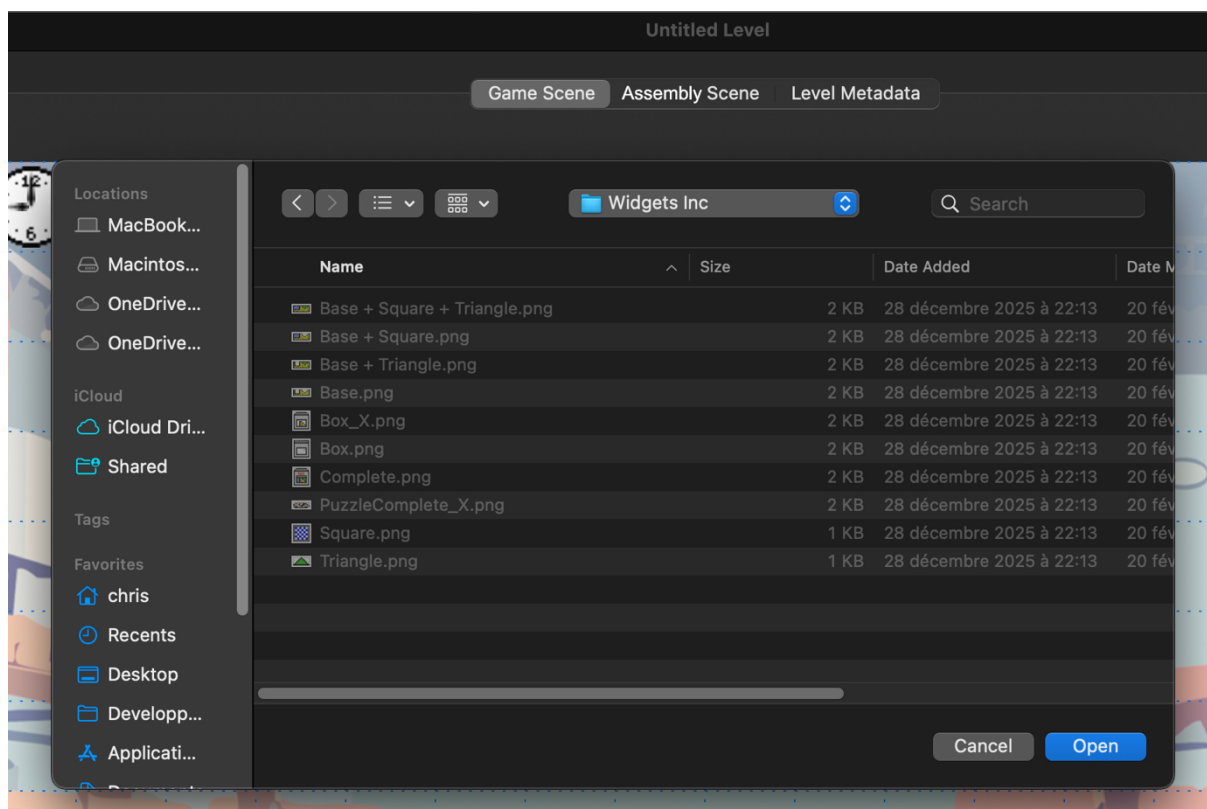




The component set:

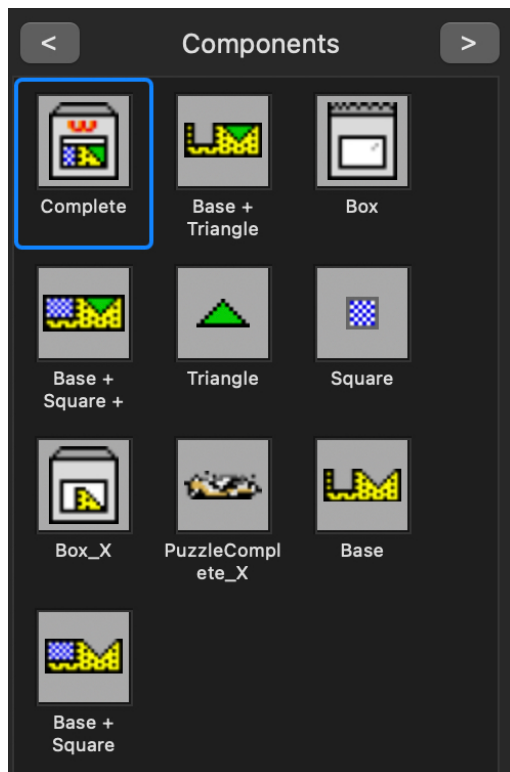
It starts as an empty zone for each new level you design. There, you can choose the Directory where the component images are stored. You may also reset everything if you plan to change the component set.

Clicking on the “Import components from folder...” button launches an Open dialog like this.

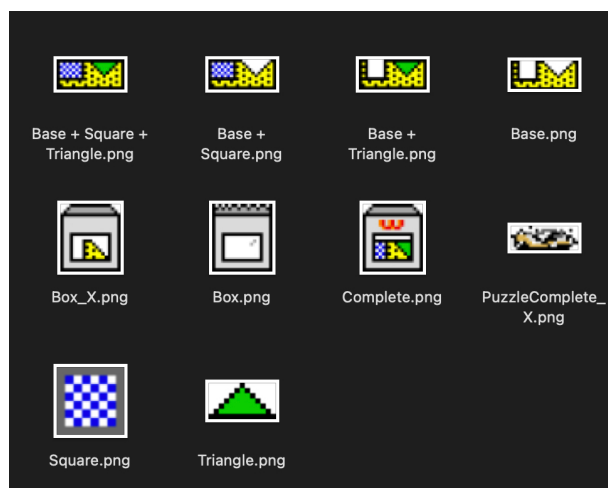


BEWARE: there is a naming convention for components (see the Components design section below)

The complete product displayed here is composed of a Base assembled with a Square and a Triangle dropped into a box. Your folder should also contain .png images for wrongly assembled intermediates. Depending on how the assembly line is designed, some combinations may never occur. In the example below, which is taken from the tutorial, the right image shows the content of a Finder window with full component names. Once chosen in the editor the components appear normally but their name may be truncated due to the limits of two subtitle lines. This is normal.



Below is the content of the folder used to import components. Notice that the full names of the component are displayed. *Don't mind if the full names don't appear in the collection view of the editor on the left. It's because the display name of each object is limited to two lines.*



Component design:

Each component consists of a single image drawn with your favourite bitmap editor and saved on a transparent background in .png format

Component size rules:

NEVER make the components taller than 40-pixels @ 72 dpi. Prefer 38 pixels otherwise they would contact the press in press switches. Image width is also limited to 40 pixels @ 72 dpi.

Component naming rules:

Use simple names for initial components: prefer Bottle to EmptyBottle

Intermediate assemblies should have a "+" sign between spaces in their name like "Bottle + Liquid" instead of "FullBottle".

Notice that different components may have the same image but there should still be one .png file for each of them. Of course, as they represent different steps of complete product assembly, they are expected to have different names.

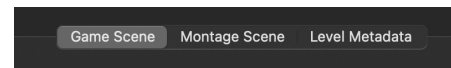
When a transformation produces a wrong assembly, add parentheses around the name if it's a combination of at least 2 components and add an "_X" suffix . For example, a broken bottle could be "Bottle_X" while a defective assembly could be "(Bottle + Plug)_X". A correct assembly would be "Bottle + Liquid + Plug", but if it corresponds to the final product, just call it "Final" or "Complete".

2- The Scene and Metadata editors

As discussed in the Flaw Forge Player Manual, a level design consists in creating scenes that are displayed sequentially to the player

- 1- The Introduction text
- 2- The Assembly Guide
- 3- The Factory with the assembly line
- 4- The Score scene

All the parameters that control these scenes are set up within 3 tab-views found at the top of the zone:



• Scenes

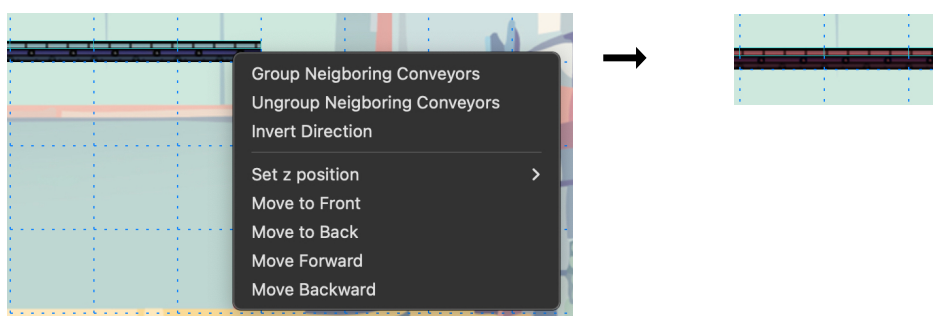
The scenes consist of a **16x10 grid of 64x64 pixels cells @ 72dpi** with a grid displayed. The modules or the components are snapped to fit into the cells. The origin of coordinates (x: 0, y: 0) is at the bottom-left of each scene.

Game Scene:

• Module Edition

<i>Operation</i>	<i>Prior action</i>	<i>Command</i>	<i>Undoable?</i>
Place a single module	Select the Module in zone 1	Click in the destination cell	Yes
Place a line of modules	Select the Module in zone 1	Shift-Click and drag	Yes (module by module)

Remove a single module	None	Option-Click	Yes
Displace a single module		Click and drag	Yes
Operation	Prior action	Command	Undoable?
Displace a line of conveyors (Belts and Pipes)	Group them first (Local menu) → the line becomes red	Click and drag Then <i>Ungroup the modules</i> (Local Menu)	Yes (All the line together)
Change the drawing layer		Chose the z position submenu for precise positioning of the top sprite or use the Move to Front ... Move Backward items in the local menu. For modules behind, use the Sprite Inspector.	No



Grouping a line of belts to prepare collective displacement

• zPositions

As stated above, the zPosition of a given module can be changed manually, provided that the whole assembly sub-line would display consistent behaviour. For example, changing the default value of a vertical pipe module should be repeated on each module of the column along the path mobile components may follow. Every module can be inspected i using the **sprite inspector** (see chapter 3). To leave room for future modules, all the zPositions are multiples of 2.

- The **zPosition** for a mobile component is **0 (zero)**.
- The default zPositions follow these rules:

Module	zPosition	Child	Inserted Component
Mobile Component	0	-	-
Conveyor Pipes	+4	-	-
Boxes	+4	-	-
Decor Pipes	+2	-	-
I/O Sprites	+4		+6 for exit module's associated components
Elevator belt	-2	-	-
Belts	-2	-	-
Single Storage	-2	+2 for the door	-
Accumulator	+2	+4 for the gauge	+6 for the associated component
Filler	+2	-2 for the liquid	-
Oven	+2	+4 for flames	-
Washer	+2	+4 for frame and content	-
Wrapper	+2	-	-
Liquid Reservoir	+2	+4 for the gauge	+6 for the associated component
Trigger Switch	+2	+2 for the Press	-
Decor modules	-6	-	-
Gear modules	-8	-	-
Wall background	-10	-	-
Emergency Pause	+10	-	-

- *Building rules for the Assembly line (Game Scene)*

Minimum requirements

There should always be a single Clock (put at the top left corner by default), a single Entry Pipe and a single Exit Module (either a full Truck or a Truck Pipe) in a Game Scene.

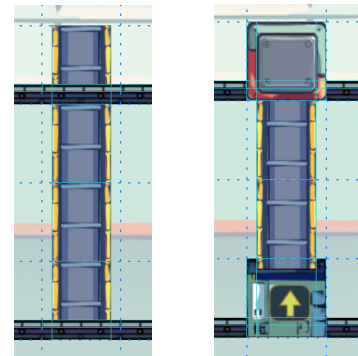
Module variations

Some modules are available in different flavours e.g. bottom boxes with horizontal or vertical arrows. The player can act on the exit direction during the game, but when you build a level, choose the flavour you want to start the level.

Elevators

Elevators should extend vertically as much as possible to link two belt lines. It means that a module will touch the bottom line, and another one will be positioned above the top line. You should add a Bottom Box and a top Exit Box to complete the setup.

Don't care about the drawing layer. It is all applied automatically. This means that the boxes can be placed before the elevator too, or that the belt line can be placed at the end of the operation with the same result. This is handy for adding an elevator somewhere.



⚠ A consequence of automatic z-positioning is that when you move modules displacement can affect an unexpected component (for example, moving a bottom box will first move the rubber belt behind it, then only you can repeat the operation to move the box)

Elevators can be crossed by belts along their body, and the components carried by the two types of conveyors are not expected to interact. When elevators cross horizontal belts, they can have their zPosition in front or behind the belts. Change the **WHOLE** elevator line to have consistent component displacement using MoveToFront or MoveToBack contextual menu items. *Notice that components will move on the zPosition layer of value 0, but if a module is also set at zero, the components will randomly move on front or behind it.*

Elevator top boxes can be used as neutral modules along belt conveyors to mask a zone (alternative = horizontal decor pipe)

Accumulators

Both **Single Component Storage** and **Multiple Storage** modules can be activated either by an Automatic Switch or manually, by a user click. If they are just above a Belt, put a simple gantry module below them (found in the Decor Modules), adjust its z position if necessary, and don't forget to place a « *Click to release* » sticker nearby.

For **Multiple Storage Modules**, the contextual menu provides a way to set the total capacity of an Accumulator (8, 10 or the default = 12 components), and the number of components released at a time (default = 1, but you can choose 2 or 3). In the latter cases, indicate it by placing a « x 2 » or a « x 3 » sticker nearby

For **Multiple Storage** and **Liquid Storage** modules, the module should know what component to accept or empty itself in the case of an error. For **Multiple Storage**, purge will result in the release of a generic garbage component.

Exit Trucks

Like for Multiple Storage modules, inserting the final product sprite onto the truck side is required. This operation is not necessary when a Truck Pipe is used.

Automatic Switches

They should always be positioned above belts. Component going through horizontally will command action from the module that is located above.

They come preset as « on » (yellow) or « off » (brown), and are available with or without a press/metal curtain.

- **Liquid Fillers** support only switches **without press**
- **Single Storage** support automatic switches **with or without presses**
- **Multiple Storage** modules can be operated in 3 ways:
 - without a switch, by responding to the player's click (see above)
 - with an automatic switch devoid of a press
 - with an automatic switch press
 -

N.B. Depending on the assembly that will operate behind the press, a spark emitter (found in Decor modules) can also be positioned behind the trigger. Z positioning is automatic.

Wrappers

There are 3 models you can choose between: A horizontal piston, a rotating device and an oil-operated machine. They function like jokers in your game and can do whatever you want.

Wrappers can be placed directly on a conveyor belt. In this case, they let components get through them to perform transformation. Alternatively, they can be put on the top of an Automatic Press Switch and operate from above.

The Oven and The Washer should be placed directly on a conveyor belt so that components go through them. They can be activated/deactivated by user click and come in the two On/Off flavours too.

Conveyor crossings

Conveyors can be crossed in your design: a belt can cross an elevator (with or without a BottomBox at the crossing) and a belt can cross a vertical tube. If there is no box to change the component direction, it will simply follow the conveyor it is in/on. Beware, components falling behind vertical decor tubes (except those falling behind enlarged tubes) are prevented from contact, and thus from unwanted assembly with components that move in perpendicular direction.

• Building rules for the Assembly guide (The tab entitled “Assemble Scene”)

Here you describe the assembly rules for each step of your level. Each assembly step is described by

- A right arrow that points to the product of the assembly (P)
- Two components C1 and C2/M2 that are one above the other with at least an arrow in between, and that get combined into P.

C2/M2 means that the entry from the top can be directly a component (C2) or a module (M2) like a Filler, an Accumulator, a Wrapper, an Oven or a Washer

Therefore, in the general form you should obtain:

C2/M2
↓
C1 → P

- If C2 is stored in a Single Storage, just put C2 in the graph with a single arrow below it
- Also add a single arrow if M2 is a Filler without an accumulator above it, Wrapper, Oven or Washer

Note that the link between C2/M2 and C1 can comprise more or less steps

C2
↓
Accumulator that accepts C2 (if Multiple Accumulator)

↓
C1 → P

C1
↓
Liquid accumulator
Filler
↓
C2 → P

To build the guide, just place the components, transformer and the appropriate arrows from the module & component catalogue with the same tools and edition possibilities as in the design of the assembly line.

C2/M2 C4/M4 C6/M6
↓ ↓ ↓
C1 → P1 = C3 → P2 = C5 → Final product

• Setting the Level Metadata

This step is quite straightforward. There you set

- 1- The level's title
- 2- The introduction text (to be displayed in the Introduction Scene)
- 3- The order components of component delivery by the feeding pipe. You can send up to 10 components consecutively. This means you can send a few components repeatedly too. The 10-cell row is just like a scene, so you can add, remove, displace and undo placement just like in the two previous scenes.
- 4- You can add the best musical ambiance that you think is suited to play your level. You can choose between 7 ambiances you can test using the button “play”. The default musical theme in the startup scene is “progressive”, then it will change as soon as a new level is introduced.
- 5- You should set the Truck capacity, Quota and level of Chaos (how much the Player will randomize your initial feeding order). The Difficulty field is just indicative, it has no impact on the game, but if filled by the author, it can help rating the level to know how to place it in a magazine issue.
- 6- The author, location and contact fields are the ones you filled when you first started the Editor application. Here you may change or complete some level-specific information like the contributors of a collaborative work for example. The default is a blank field. When several authors are listed, separate them with a comma. The location(s) should be entered as **Town1 – Country1, Town2 – Country2 ...** list. Contact(s) are displayed in the same way as authors.

3- The inspector

The Inspector will display all the parameters that Flaw Forge Editor uses to characterize a Sprite. To display a Sprite in it, you should select it first by a long click in the assembly scene, the guide scene or the feeding order scene. Such selection will result in the occurrence of a local menu of the different sprites that appear under the mouse click. When you have chosen a sprite from it, a purple rounded square will occur around the cell, and the sprite definition properties will be listed.

This inspection mostly has a debugging purpose.

You can see the list of node properties for each of the Flaw Forge sprite.

What could be important for you:

moduleType	the category of modules (or components) the object belongs to.
id	the name of the object
displayName	the same as above
imageName	the name of the image built in the FlawForge application
position	the actual position of the object. Remember the origin of the scene is at the bottom-left. [See also the zone 5 chapter]
positionOffset	It is (0,0) for objects centred in cells. Else, it records the positioning of the object inside the cell.
Size	the object size in pixels
bodySize	When objects interact with each other thanks to SpriteKit's physics simulator, a physicsBody of 'bodySize' is added. This size may be different from that of the object's texture itself. The internal categories that define the type of physical interaction are also listed below but they give no valuable information to design and debug levels.
bodyOffset	the same as the positionOffset but for the physicsBody
hHotSpot	if the FlawForge engine needs to compute proximity between objects
vHotSpot	it may use these parameters.
isDynamic	defines if the objects can move or not in the scene
gravitySensitive	defines if objects obey or not to the gravity field
zPosition	the drawing layer for each module.

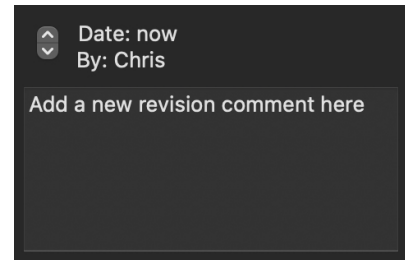
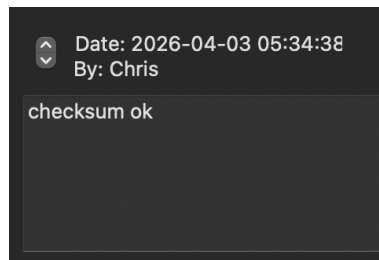
The other properties correspond to other internal sprite parameters defined at the development level that have no key influence of your level's behaviour.

4- The Revision zone

If you want to trace level revisions, the zone at the bottom-right of the windows is for you. You can enter a new revision sentence to comment what you changed. It will be saved automatically with the present date and time inside the level and integrated in the internal checksum. You can loop through previous revisions using the two-way button.

Shown are a previous revision example (left) and the prompt for new revision information.

Notice that if you don't add anything into the text field to replace "Add a new revision comment here", no revision data will be saved into the level's file.



5- The Mouse location zone

The real-time mouse coordinates (unsnapped mouse x, y) and the snapped location (at the center of each cell) are displayed in this Zone.

Mouse location: x = 648.1 , y = 202.8
Snapped: x = 672.0 , y = 224.0

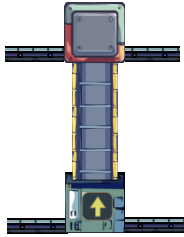
C- Saving and testing a level

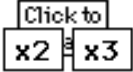
Saving operates via usual commands Save as... and Save and writes a .fflevel file (actually a .json file under the hood) to the location you chose.

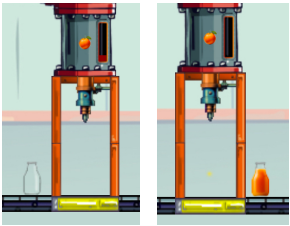
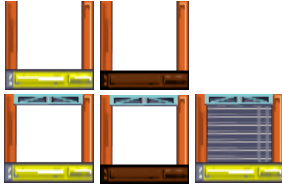
D- Level Printing

Printing a level produces a 2-page report starting with the level's title in red. Then, the 3 Tab views (Game Scene, Assembly Scene, Metadata Scene) are printed below. Note that in the Game and Assembly scenes, the module's physics bodies are not displayed in the printed version.

E- Module reference

The Clock		Runs at 1 min per second of real time at Slow speed
The Emergency Pause Button		Equivalent to Pause (⌘ P, macOS) or to the swipe left gesture up to 3 times depending on the speed you start from (iOS)
The Belts		Belts go in a fixed direction. They can span all the factory and wrap the left and right sides (they have no visible ends in this case). Otherwise, the objects that are conveyed and that reach a Belt End will fall down after a brief cartoon-like stop in the air. <i>Falling components</i> follow gravity physics and will stop or not depending on the configuration: they can stop on a belt below, in a storage box or in an accumulator for example, but they don't if they go through a vertical pipe that crosses a belt or through a falling box with a vertical exit set.
The Elevator, Bottom Box and Exit Box		Elevators never appear alone. They start climbing behind a bottom box that you can chose and continue until they reach an exit box that will deliver climbing components to the right or to the left, depending on the direction of the belt at that point. Note that bottom boxes can also appear in the body of elevators (not only at the bottom like this). 👉 Click or tap on the bottom box to change its output direction.
The Neutral Box		It looks like an Exit Box but it does nothing, except that it will hide the component that moves behind it. It's just like a horizontal pipe connected to the belts but it stands before the belt.

	 	component is released! Here the module can store only green triangles. It is empty. The Liquid Storage Module is absent from the original <i>Factory</i> game, so it will appear only starting at issue #2 of the <i>Flaw Forge Magazine</i>. It is fed with components that can be mixed into a liquid. This module functions as a reservoir for the filler that stands just below it. Liquid delivery is only possible if the reservoir is not empty. <i>Like for the Multiple Storage Module there is a red bar that climbs in the black window to indicate the amount of liquid available.</i> Special indications for the storage modules are displayed beside it. Like for Single Storage Modules, Multiple Storage Modules can release a component only if clicked or tapped. In some cases, the delivery of a new element (solid or liquid) requires that 2 or 3 components have entered the Storage Module. The indicator bar must be red to allow delivery, otherwise the amount stored is insufficient.
In and Out Modules	   	Each scene contains at least a Feeding Pipe and a Truck Exit, either the full truck or the pipe. Note that the Truck side (not the pipe) displays the icon of the finished product. You may also encounter two more modules: The Recycler and the Trash.

The Fillers		If liquid delivery is needed into a recipient, you will find one of the 3 Filler Modules. Liquid availability can be controlled or not by a liquid accumulator above it. The Filler Module is always under the control of an automatic switch. <i>Example: Before and after filling a bottle with orange juice.</i>
The Oven, The Washer		These modules can be switched on and off by the user. They are placed directly on the belts.
The Transformers		They can be either placed directly on the belts or controlled by automatic switches
The Automatic Switches		They come in two versions. The Naked Switch (above) can be set to 'On' (Yellow bar) or 'Off' (Brown bar). You will see all the transformation that happens in it. Typically this is the model you will encounter under the Fillers. The Press Switch has a press integrated in it. It comes down during the assembly as shown. The press will mask all that happens behind the iron curtain. Sometimes Sparks may be visible from behind the press to represent some metal cutting or soldering work.
The Decor Modules The Decor Pipes	?	I will not show them to you here. They just do nothing in the game except make it a little more fun or self-explanatory. You will encounter, Windows, Ladder, Balcony (with you on it), Smoke, Parts of a big metal Tank, Gears, Danger signs etc... Discover them as you progress through the game.

