DESIGN DOCUMENTATION

Enabling Learning By Teaching:
Intuitive Composing of E-Learning Modules

21st of October 2015

Author: Lene Thuseth

CONTENTS

THE IDEA

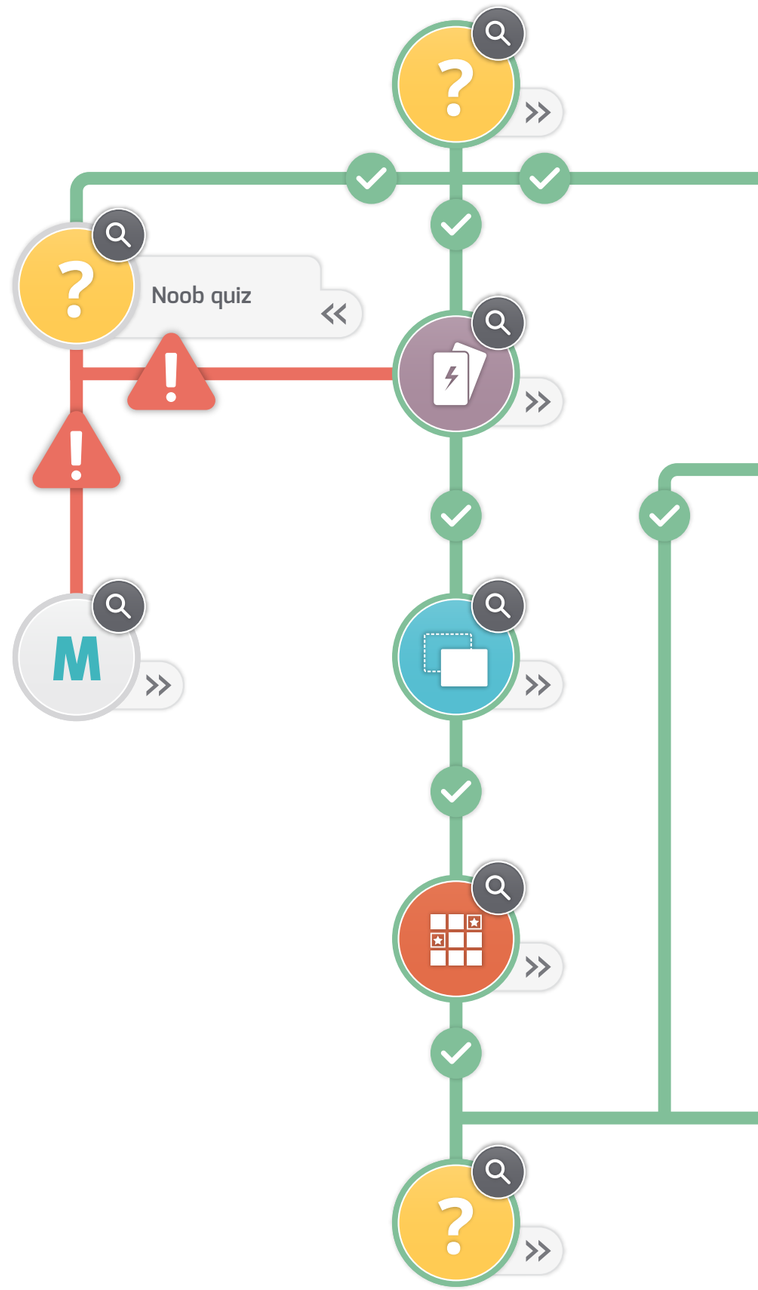
The idea is to make a highly intuitive and engaging system to compose e-learning modules. The system will support authoring of both linear and non-linear e-Learning[1], and is designed with an emphasis on younger users. In the MVP (Minimum Viable Product) the system will allow users to compose H5P-modules (h5p.org) with an unmodified version of the H5P composing tools.

The design and concept is based on research done and published by plaimi (secure.plaimi.net). More information on market research, target audience and the concept itself can be found in their paper[2].

> ***"In an effort to foster learning by teaching, we propose the development of a canvas system that makes composing e-learning modules intuitive. We try to empower and liberate non-technical module users by lowering the bar for turning them into module authors, a bar previously set far too high. In turn, this stimulates learning through teaching. By making a damn fine piece of software, we furthermore make module authoring more pleasant for experienced authors as well. We propose a system that initially enables users to easily compose H5P modules. These modules are successively easy to share and modify. Through gamification we encourage authors to share their work, and to improve the works of others."***
>
> *Abstract from the Enabling Learning by Teaching: Intuitive Composing of E-Learning Modules paper*

The focus of the concept is to foster "learning by teaching", by encouraging students to make and share e-learning modules, and re-use and re-mix modules made by others. At the same time we make it much easier for professional e-learning authors to make engaging and adaptive e-learning.

The design itself will have a heavy focus on all aspects of usability, including universal design (accessibility), motivation by gamification, and how intuitive and easy it is to use and navigate the system.



THE DESIGN

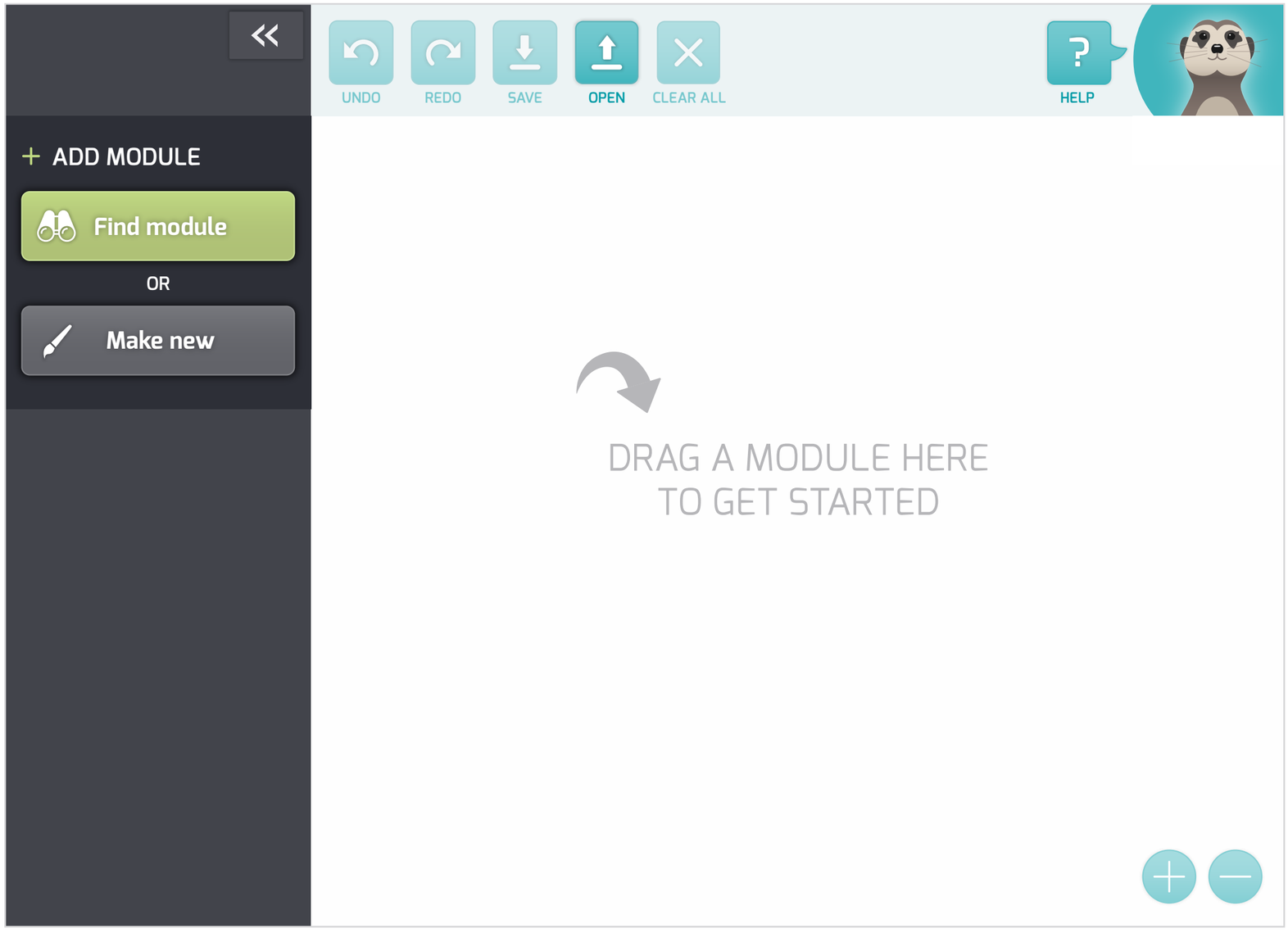
The look

Because children are the primary target audience, the design makes use of bright colours, strong contrasts, big buttons, playful avatars, and a friendly language. At the same time kids aren't the *only* target audience, so the design also needs to appeal to a more mature audience. By using bright but slightly muted colours we achieve a colour palette that looks fun and child-friendly but not too childish. This also gives it a modern feel.

Round shapes such as circles and rounded corners feel friendly and playful, enticing the user to play with the system, and that is exactly the message we want the design to send; **"come play with me, come have fun!"**

Big buttons and strong contrasts also help the user navigate the system, by making it obvious how to interact and where to click. By having a tidy interface with only the necessary functionality available, we avoid confusion and frustrating the user. The less time the user spends trying to figure out how to use the system, the more time they can spend on the actual task.

Buttons that can be clicked use visual effects such as gradients and shadows to make them appear elevated from the surface. People intuitively know that they can click these buttons.



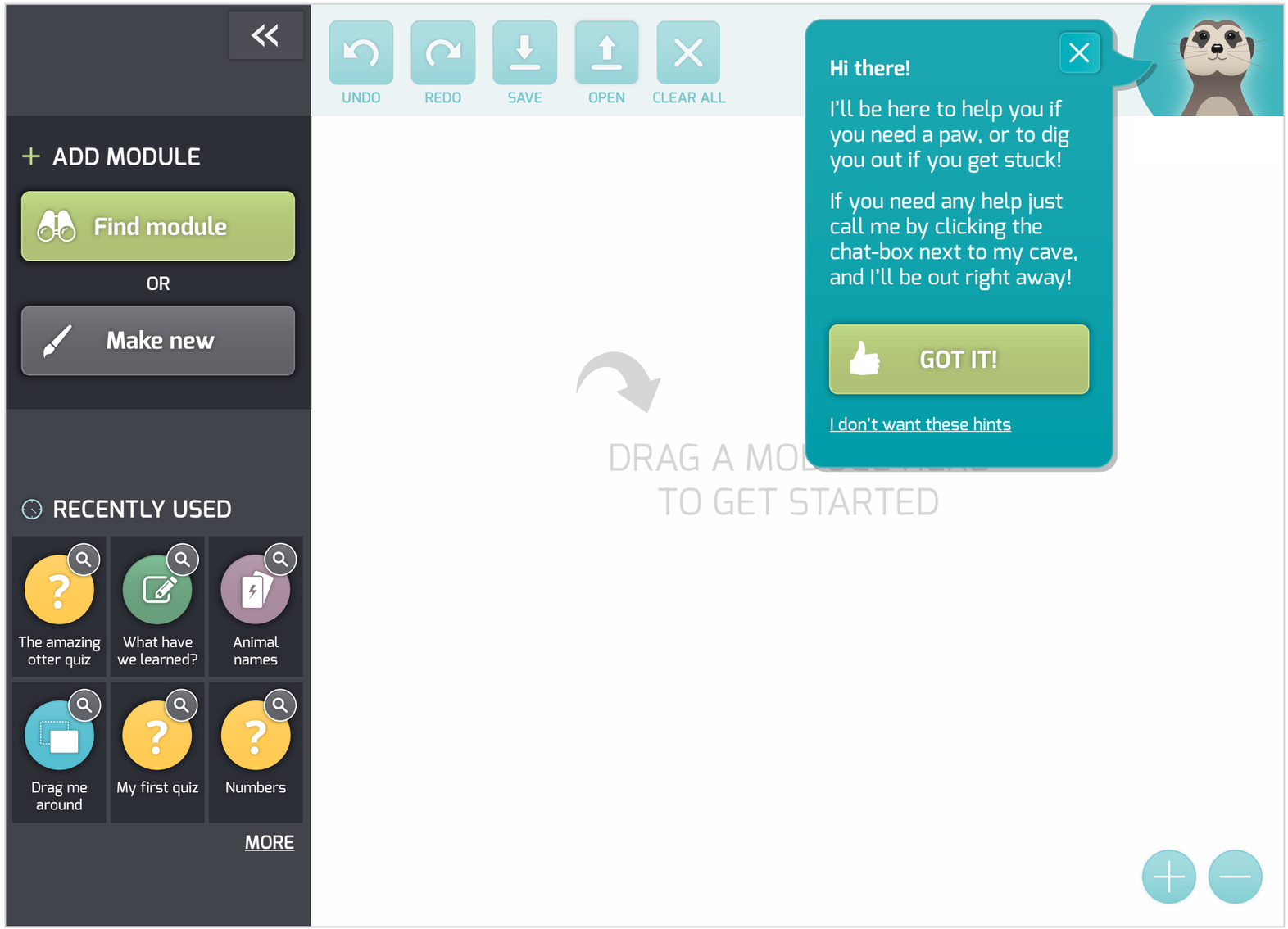
The language

By using a casual and friendly language we try to introduce a sense of humanity and human touch to the system. It feels like your friend rather than a cold and robotic system. We make use of an avatar that will pop up with hints to help the user along, or to let them know when they've done something wrong. We don't want to give kids a feeling they've done something wrong, so even when there's an error message it is phrased in a positive and friendly manner.

Instructions on buttons and actions are clear and concise. We use the terminology *find* rather than *search*, and well-established terms such as *undo* and *save*.

The language aims to be fun and easy to understand for kids, but clear and comprehensible for adults.

We use metaphors to make something a little bit complex, such as flows and forks, more understandable. Since the avatar is an otter, we use the metaphor of rivers and dams when we talk about the flow through the modules, and possible broken logic that impedes the flow. This can help make a complex concept seem much more understandable for kids as well as adults, and help them author complex e-learning modules/courses.

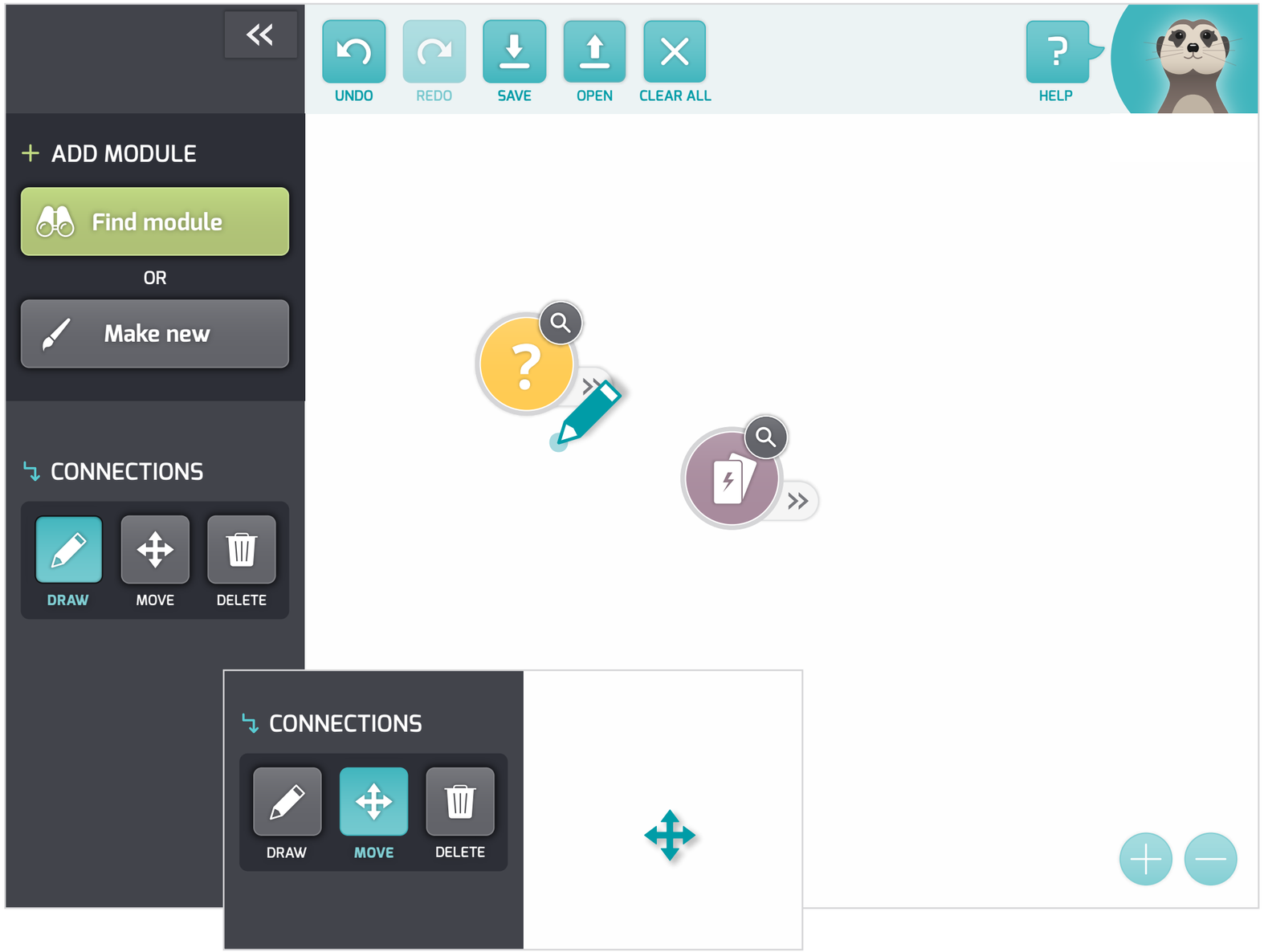


Functionality and responsiveness

The most important thing of a system of any kind is its functionality. It doesn't matter how beautiful it looks if it doesn't feel responsive and logical. It needs to give prompt feedback when actions are performed, so the system feels alive and responsive.

We have no splash screens or loading bars. Actions lead to an instant response, e.g. if a user searches for something, the search results populate as they type giving instant feedback to their actions. Kids, especially, hate waiting, and will not have the patience for fancy splash screens. Therefore the user may jump straight into the task with no delay as soon as they enter the system.

Our focus on responsiveness is further emphasised by things like hover effects for mouse-users and on-tap effects for tablet users. When you hover a button, its colour will change, and it will animate (e.g wiggle) in order to convey a possible interaction. Furthermore, when the button is clicked, its appearance changes as a visual feedback. As an example the 'CONNECTIONS' actions are in a toggle bar where either one or fewer of the actions can be active. When one action is activated the mouse cursor changes to indicate the action that can be performed, e.g. the 'DRAW' function transforms the cursor into a pencil.



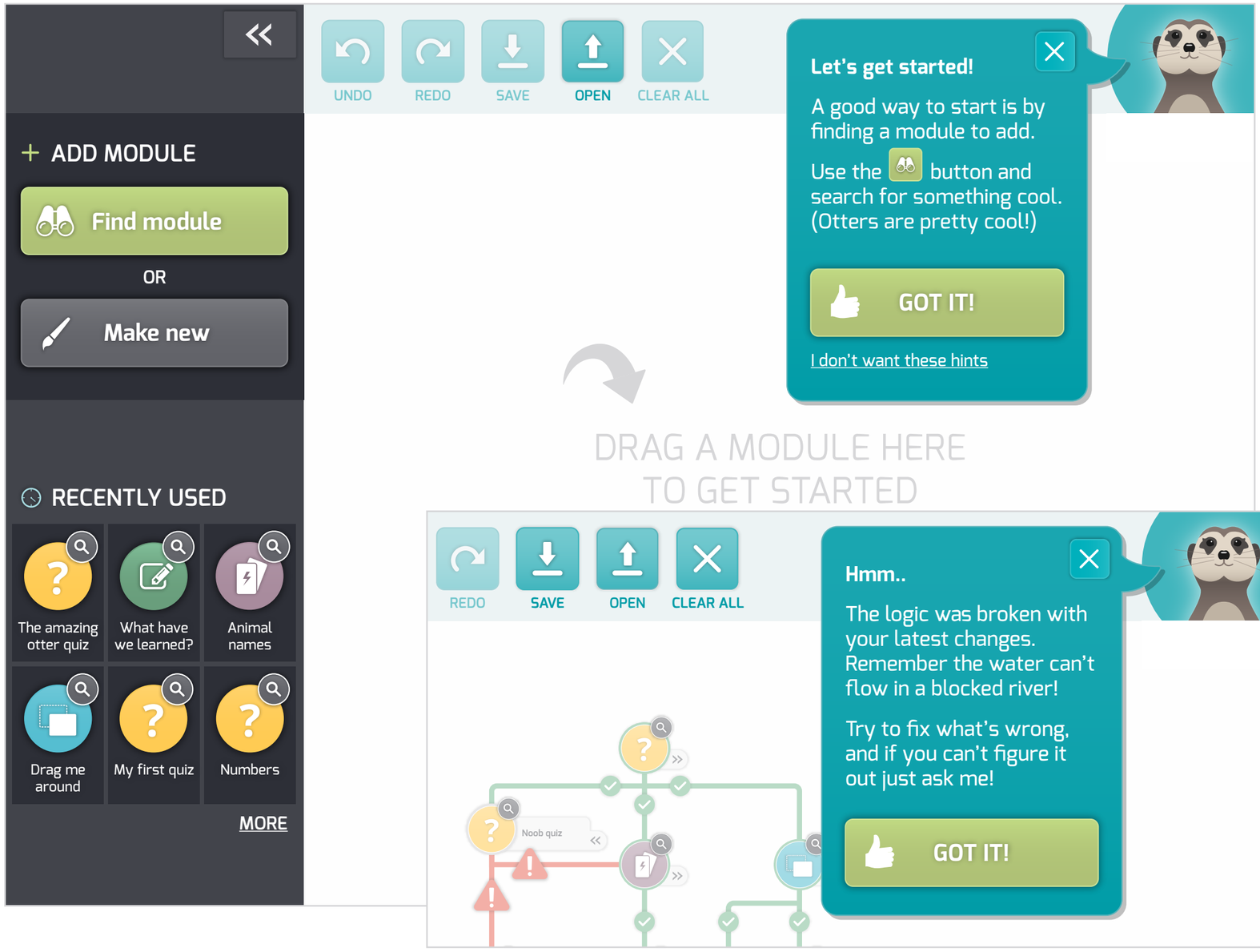
Functionality and responsiveness (cont.)

The more complex ways the system is responsive and feel alive, is by understanding a user's behaviour. If they fail at a task or take too long to do something, the system will assume the user might need help with what to do. One example is if the user starts the system and doesn't do anything within a certain time limit, the avatar will pop up with a hint on how to get started. If a user starts performing actions right away, this hint will not appear. The same hints will not appear more than once.

Another example is if the user breaks the logic of a connection. The hint will pop up telling them that their latest changes has caused errors in their flow, and how to fix it.

All 'big' actions such as 'CLEAR ALL' will have an extra layer of confirmation before the action is completed in case of enabled by accident. All actions can be reversed with the undo function or re-done with the redo function.

We never use just icons for important functionality, but include text labels as well. The only exception is when the narrow side bar is active by choice of the user.

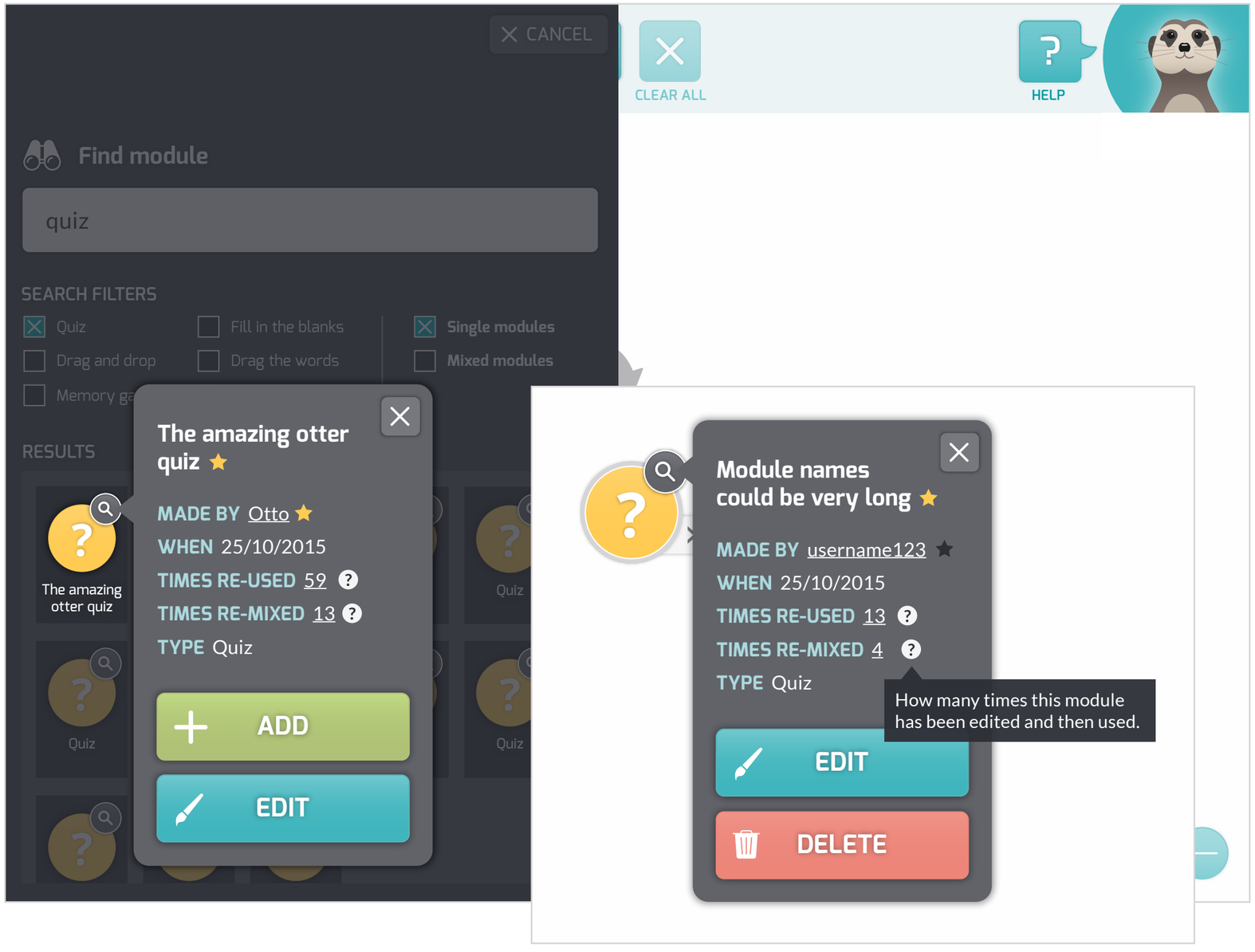


Consistency

To improve a system's usability, the navigation needs to be consistent. If a button does something different in one place in a system than it does in other places, the user may get confused and frustrated. Consistency improves predictability, which in turn increases learnability[3] (how quickly a user learns how to use the system).

This system has very consistent navigation. An example of this is how the module icon can be interacted with. If the module icon or magnifying glass icon is selected, the info box will pop up. This action is available no matter where the icon appears. The magnifying glass icon indicates that the module isn't empty, unlike when the icon is on its own when adding a *new* module. As soon as the module is authored, the magnifying glass is added to show that the module is no longer empty. From this info box, the module can be 'EDITED' (re-mixed), 'ADDED' (re-used) if it is not on the canvas already or 'DELETED' (from the canvas). So the actions are contextual, but the navigation is consistent and can therefore be predicted and expected by the user. The actions appear in order of importance, 'ADD' is more important than 'EDIT' (as it allows for quicker use), and 'EDIT' is more important than 'DELETE'.

There are also colour consistencies in buttons across the system with similar actions.



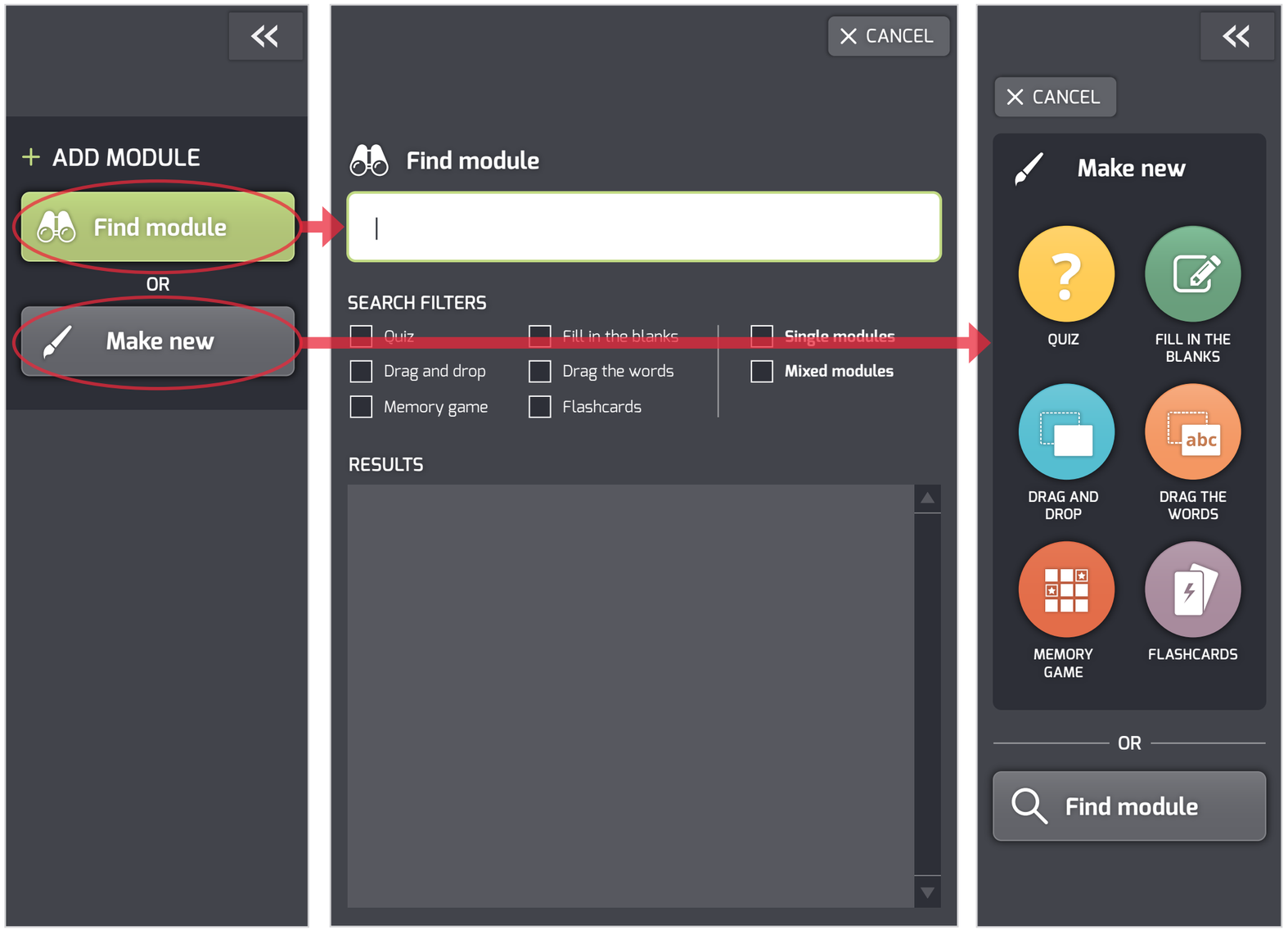
Animations and transitions

Another way to enhance the feeling of consistency is by applying animations when transitioning to a new page by clicking a button. A context shift can cause unease if it's not visually expected. By using animations, such as morphing, we can ease the transition for the user.

As an example we have the two buttons in the toolbox default view; 'Find module' and 'Make new'. When clicked these will transition into respectively the search page and the make new page. We ease context-switching with animations. E.g. by morphing the search button into its new state, and leaving it active for the user to type into straight away, we create a much less confusing context-switch.

Similarly, the two buttons will move and morph into their current appearance on the "Make new" page rather than just appearing in their new states.

By selecting the cancel buttons on each of the new screens, it will morph back into its previous state.



Gamification

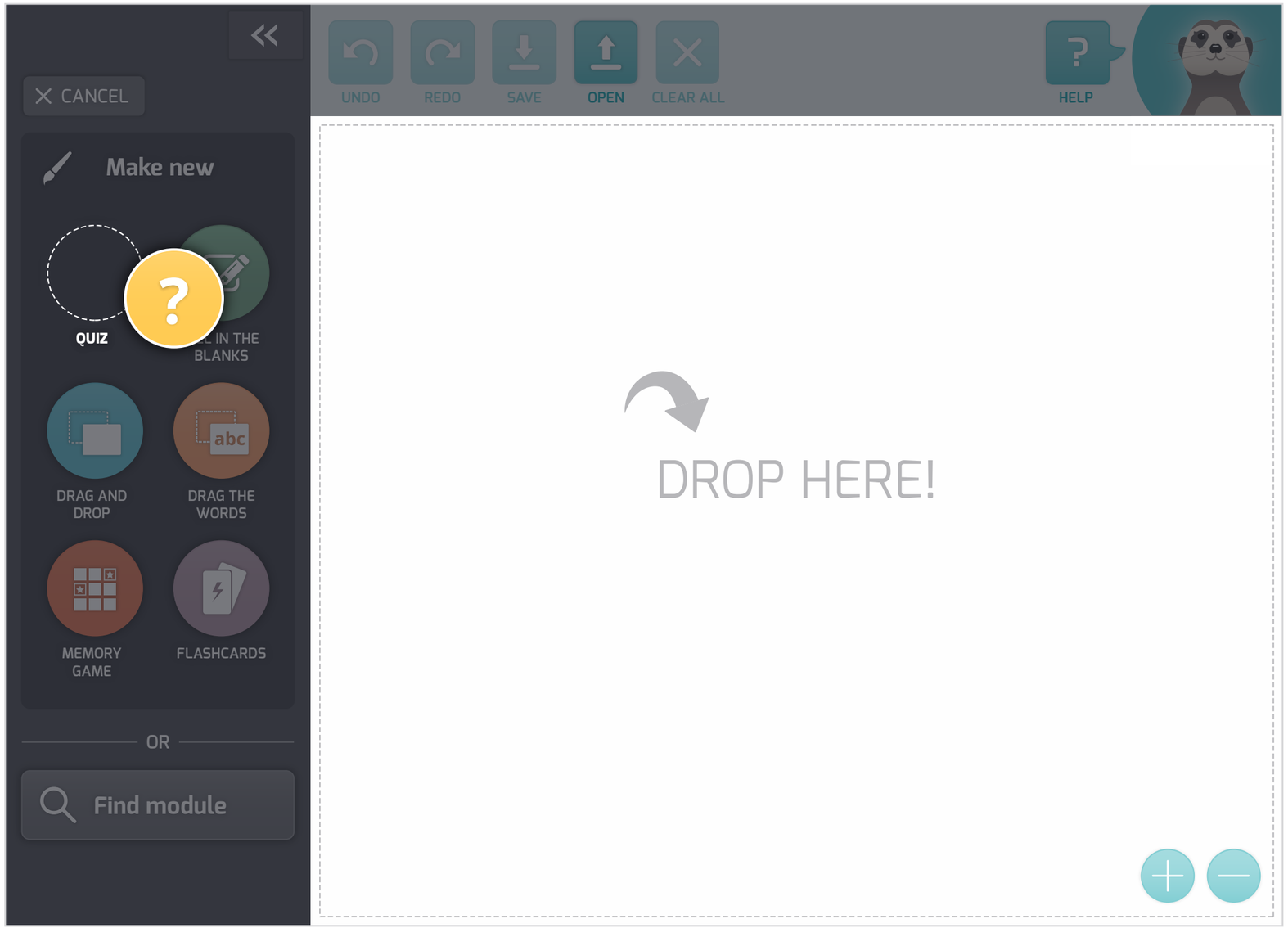
Gamification is when you apply game mechanics and metaphors to non-game scenarios in order to motivate and engage people to reach their goals.

Our system will be gamified, with the use of customisable avatars and achievements, and with the look and feel of a game, by allowing the user to drag items around freely to accomplish their goals.

We use gamification, especially to motivate learners to share and re-use other people's modules. We do this by awarding achievements for re-using modules, and for having modules that are frequently re-used. To prevent meta-gaming* we can add an 'influence' system, wherein a module gets a higher score if used by more unique users. The influence system influences module popularity.

Users may customise their avatar, which gives the feeling of having their own personal helper. This creates more attachment and loyalty to the system, and makes it feel more like a friend than a computer system. Gamification elements like these make the user feel more involved, and in turn more engaged in the task that they are doing.

*Exploiting the game mechanics in order to achieve a high score.



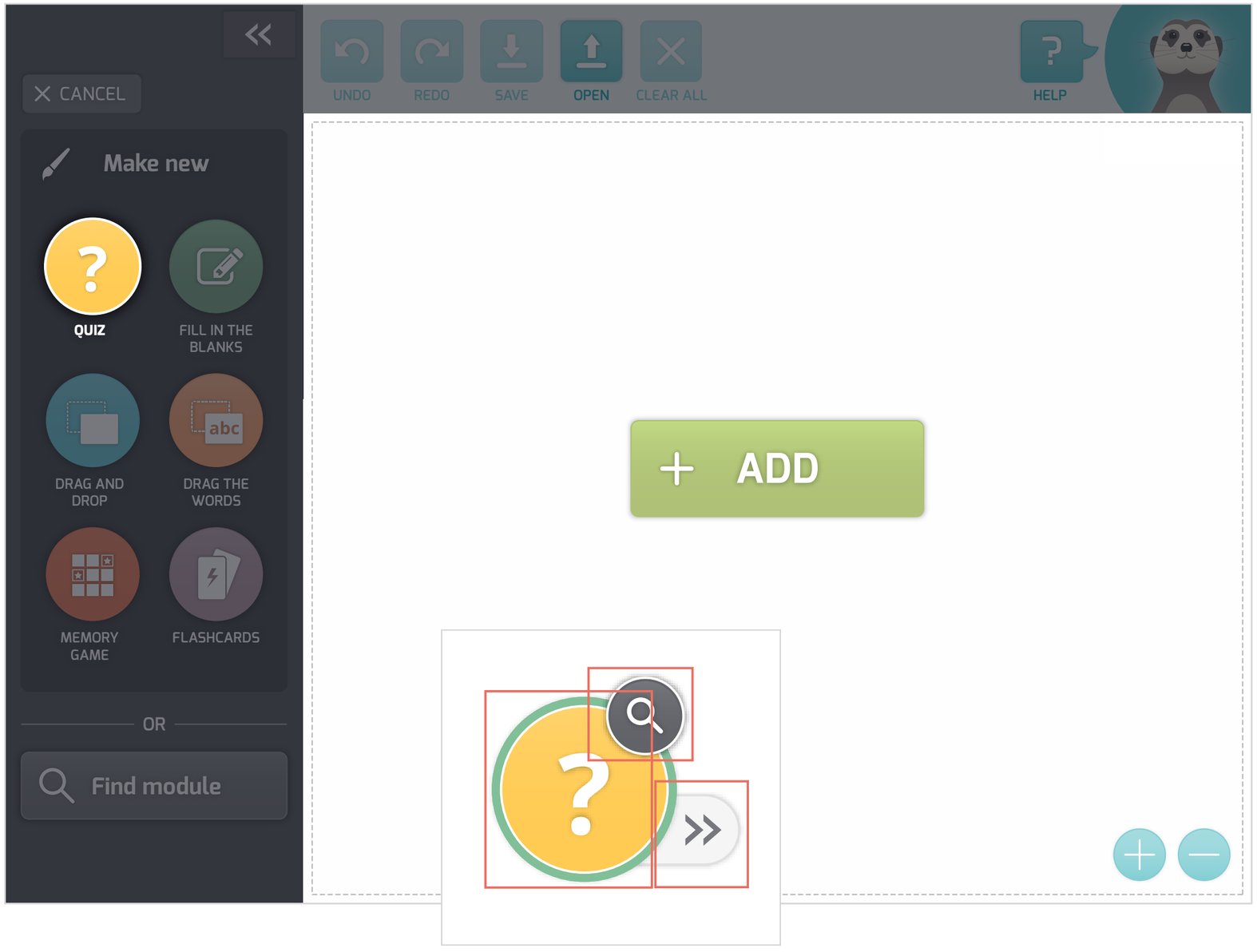
Universal design (accessibility)

Our system will be a website. One of the main motivating factors for this decision is that we want our system to be available to as many people as possible. This means we need to support the most popular operating systems and devices. This also means we need to take into account accessibility software such as screen-readers. Our system aims to follow the WCAG2.0 requirements as outlined by Difi. (uu.difi.no)

We do this in several ways. One way is making sure all colours have a high enough contrast ratio, and that we don't use colour by itself to indicate things. An example is the connections, which appear red or green, depending on whether the logic is correct or not, that also have an icon in a different shape to indicate the same thing.

The primary way of navigating the system relies on drag-and-drop, but we also have alternative methods of navigating. E.g. the icons may be dragged to the canvas, or they may be selected, at which point an 'ADD' button will appear. Selecting this button is functionally equivalent to the drag-and-drop action.

We use large buttons with larger hit-boxes because children are in the target demographic. E.g. the module-button: activating the module icon does the same thing as activating the magnifying glass. The magnifying glass serves primarily as a visual indication.



THE CONCEPT

The essence

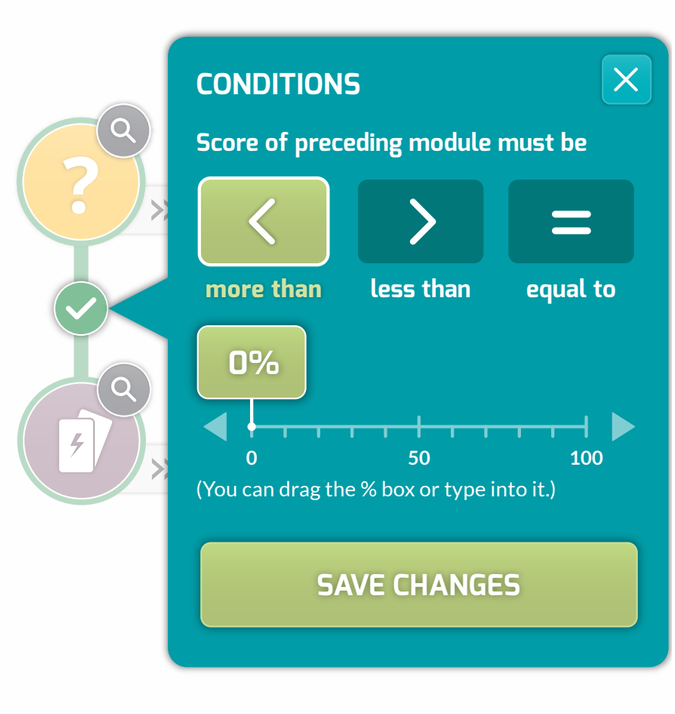
What makes this product so unique, is that it makes it easy for anyone, regardless of age, background, or technical ability, to author sophisticated non-linear module structures. This is something that previously has only been available to professionals working in complex systems. Our system is so intuitive and easy to use that the user can focus almost entirely on the task at hand.

The biggest drawback at this stage is the complicated and not very user-friendly H5P authoring system. However, this can be seen as an opportunity to work on the user experience of the H5P-system. This will in turn benefit H5P itself and every user of it, including our system. Additionally, the authoring capabilities of future e-learning modules might be designed to support our system from the ground up.

Research done in the earlier stages of this project indicate that there are several operators in the market that have some of the same features that this project has, but none that do exactly what we aim to do. There are many module authoring systems, and many e-learning systems for kids, as well as several canvas systems where users can design their own games or experiences. But there are none focussing on visual and intuitive authoring of e-learning modules. Nor any focussing on the learning by teaching effect. By using gamification to encourage students to share, use, and improve other people's work, we allow for a growth culture for social learning. We encourage problem-solving via gamification. The user is free to move items around on their "canvas" as they please, and visually build up their course, with actions and consequences.
A student can easily author a non-linear module by using if-expressions. E.g. if the score of a module is less than 80% then go to this module, if not proceed to the next module.

The system helps the user formulate if-expressions. E.g. the user has four modules, A, B, C, and D. They connect B to A with the condition of a 50% completion score of A. They then connect C to A with a 75% completion score condition. Finally they connect D to A. D then automatically serves as a "catch all", by catching all scores less than 50%. The users are of course allowed to overrule this, but then they also have to fix the broken logic.

A good way to encourage kids to learn is to involve them in their learning, and a great way to involve kids is to let them have fun. **This is a key feature of this project; making it FUN and EASY to learn and develop e-learning.**



THE SPECIFICS

01. Colours, fonts, and styles

MAIN COLOURS

MC1 #00B0BC
MC2 #009CA7
MC3 #2D2F37
MC4 #C1DA83
MC5 #AFC176

SECONDARY COLOURS

SC1 #40C4CD (or 75% of MC1)
SC2 #73D4DA (or 55% of MC1)
SC3 #E5F7F8 (or 10% of MC1)
SC4 #42444B (or 90% of MC3)
SC5 #616369 (or 75% of MC3)
SC6 #76787D (or 65% of MC3)
SC7 #F4F5F5 (or 10% of MC3)
SC8 #D0E3A2 (or 75% of MC4)

FONTS

EXO
https://www.google.com/fonts/specimen/Exo

Used for most text

STYLES
Medium
Demi-bold
Bold

Lato
https://www.google.com/fonts/specimen/Lato

Used for better readability of numbers, used in info-popups and tool-tips.

STYLES
Regular

SHAPES AND SHADOWS

The design is based on circles and rounded shapes. Shadows are used to indicate interactivity. Bigger shadows are used on most important buttons, darker shadows on dark backgrounds and lighter shadows on lighter backgrounds.

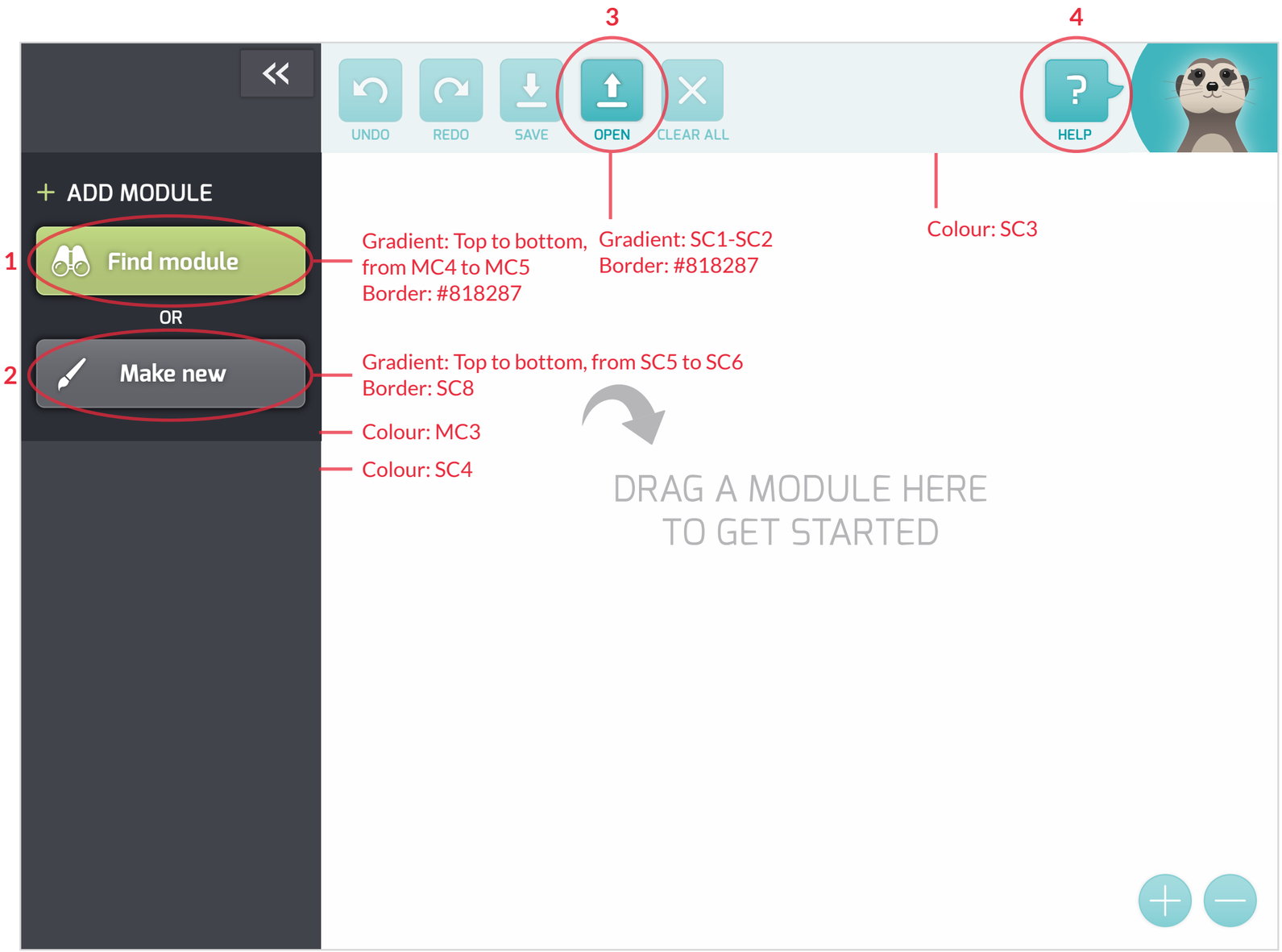
02. Initial view and interface

As already mentioned there are no splash-screens, loading or extra clicks to get into the system, you start playing right away.

To make it clear and simple, only a few actions are presented when you start off. The highlighted feature is the 'Find module' (**1**) as we want to encourage users to mainly re-use already existing modules. There is also a button to 'Make new' (**2**) module, which will use said modules authoring tool.

In addition you can 'OPEN' (**3**) a project, ask the avatar for 'HELP' (**4**), and view profile options by selecting the avatar picture. All unavailable functionality is clearly marked as such, and there is a textual hint that tells the user what they need to do to start.

If the system detects that a user spends a while without doing anything, it assumes the user is new, and a welcome message pops up. With a login-system or caching, the system would know only to do this the first time. By having a delay we avoid bothering users that actually know exactly what to do, and start off right away.



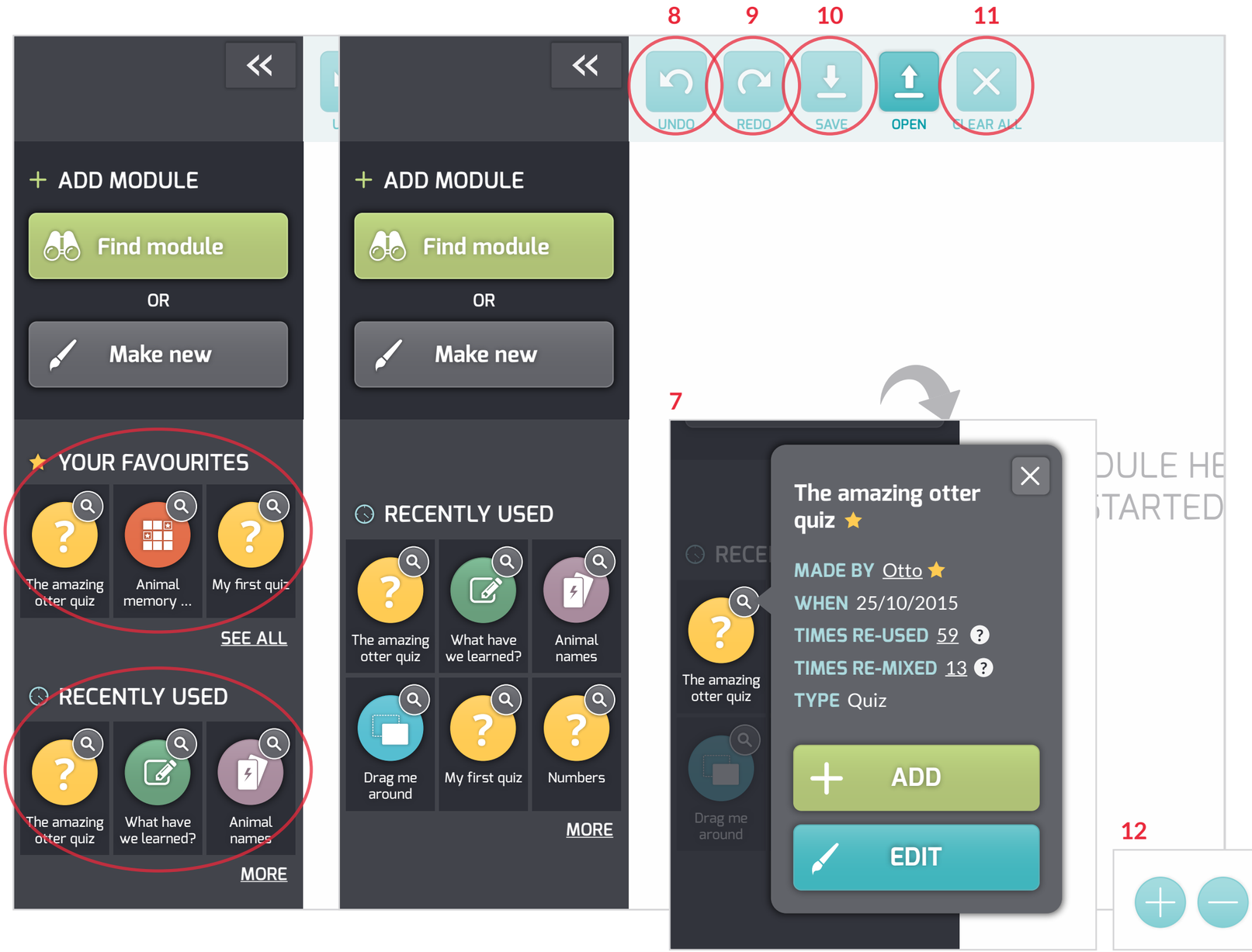
03. Additional functionality on initial view

Other functionality can be added to the initial view to motivate users, such as a list of some of their 'FAVOURITES' (5) or a list of their 'RECENTLY USED' (6) modules. Other functionality could include a list of the 'MOST POPULAR' modules, based on the influence score mentioned earlier.

Modules can be interacted with from these lists as well, indicated by the consistent design. If the user selects a module the module info box will appear (7), like it will other places in the interface. They can also drag these modules straight onto their canvas.

One problem that needs solving is how to suggest relevant modules. If a student is creating a course about Africa, then modules about algebra aren't very relevant, and serves more as a distraction than anything else. One possible solution is displaying *suggested modules* based on their search results.

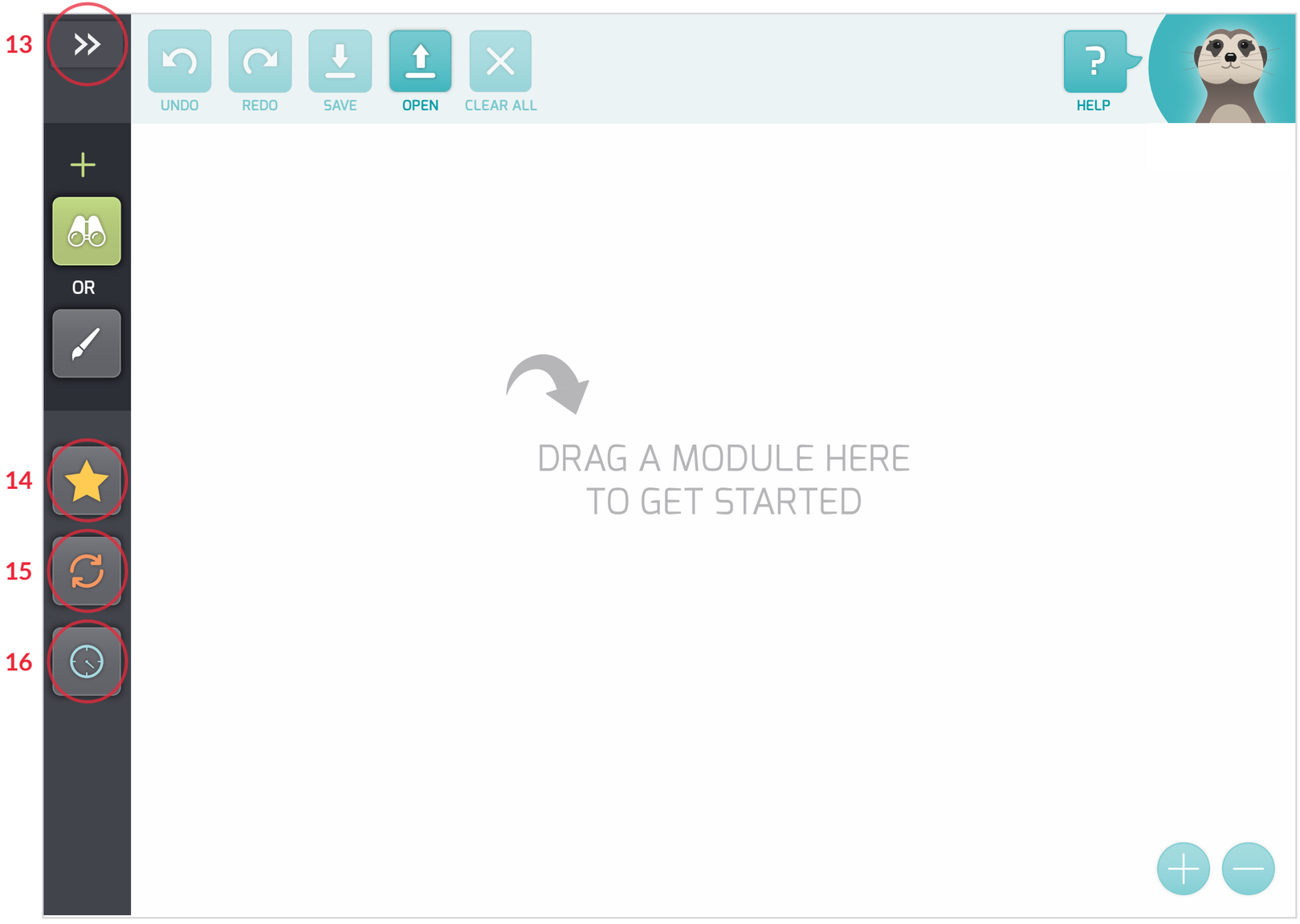
Other functionality that becomes available when appropriate is 'UNDO' (8) — that reverses the last action, 'REDO' (9) — that undoes an undo, 'SAVE' (10) — that allows the user to save their project, and 'CLEAR ALL' (11) — which clears the whole canvas. In addition the canvas may be zoomed in and out, by either scrolling or by using the '+' and '−' buttons (12) in the lower right-hand corner.



04. Expand/shrink the sidebar

To afford more room to the canvas, the left-hand side bar can be shrunk to a narrow bar with only icons. As a rule, we have text labels for all important functionality, but on the narrow bar we use only icons to save room. If the user is unsure of what any of the buttons do. they can simply expand the bar (13) to see the full buttons.

The 'FAVOURITES' (14), 'MOST POPULAR' (15), and 'RECENTLY USED' (16) buttons will result in the same action as the 'See all' (for 'FAVOURITES') or 'See more' (for 'MOST POPULAR' and 'RECENTLY USED').

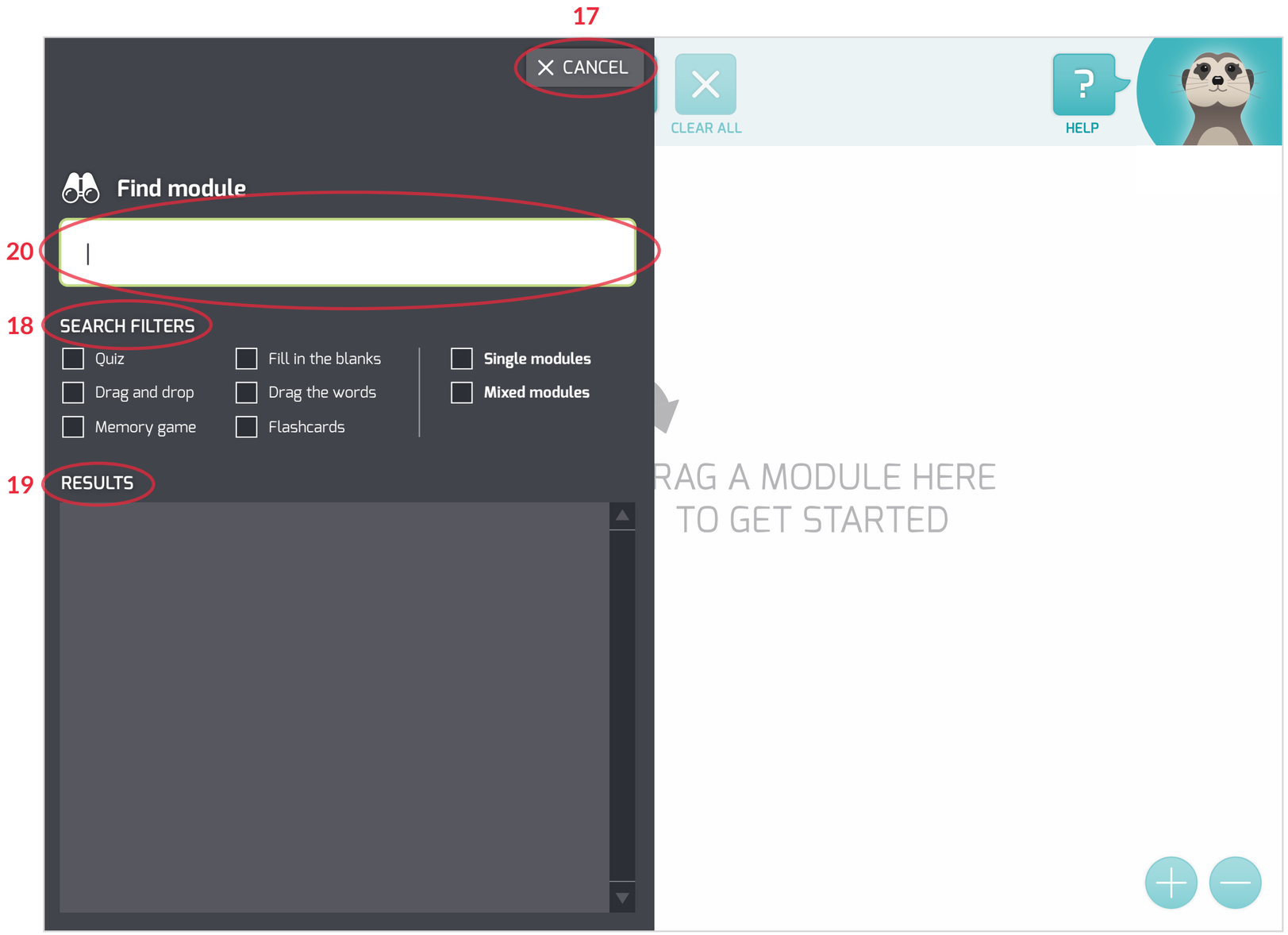


05. Find module

The ‘Find module’ panel is the only panel on the side bar that cannot be shrunk to a narrow version. Instead a ‘CANCEL’ (17) button appears. When selected the user is directed back to the previous screen. This panel takes up twice as much space as the normal side panel, to allow for extra information such as ‘SEARCH FILTERS’ (18) and ‘RESULTS’ (19).

The search field itself appears active and ready to be typed into when the user clicks the ‘Find module’ button, to avoid having them click a second time. (20)

The user can apply search filters before or after searching (21 - next page). Search results will be listed as module icons with the same appearance as elsewhere in the system (22 - next page), and can be dragged and dropped onto the canvas from there, or added by using the ‘ADD’ button (23 - next page).



21

CANCEL

Find module

quiz

SEARCH FILTERS

- [x] Quiz
- [ ] Drag and drop
- [ ] Memory game
- [ ] Fill in the blanks
- [ ] Drag the words
- [ ] Flashcards
- [x] **Single modules**
- [ ] **Mixed modules**

RESULTS

The amazing otter quiz | My first quiz | My first quiz | Quiz | Quiz

Quiz | Quiz | Quiz | Quiz | Quiz

22

CANCEL

Find module

quiz

SEARCH FILTERS

- [ ] Quiz
- [ ] Drag and drop
- [ ] Memory game
- [ ] Fill in the blanks
- [ ] Drag the words
- [ ] Flashcards
- [ ] **Single modules**
- [ ] **Mixed modules**

RESULTS

The amazing otter quiz | Animal memory ... | My first quiz | Drag me around ... | My first quiz

Quiz | Quiz | Quiz | Quiz | Quiz

23

CANCEL

Find module

quiz

SEARCH FILTERS

- [x] Quiz
- [ ] Drag and drop
- [ ] Fill in the blanks
- [ ] Drag the words
- [x] **Single modules**
- [ ] **Mixed modules**

RESULTS

The amazing otter quiz

Quiz | Quiz | Quiz

The amazing otter quiz ★

MADE BY Otto ★
WHEN 25/10/2015
TIMES RE-USED 59 ?
TIMES RE-MIXED 13 ?
TYPE Quiz

ADD

EDIT

06. Make new module

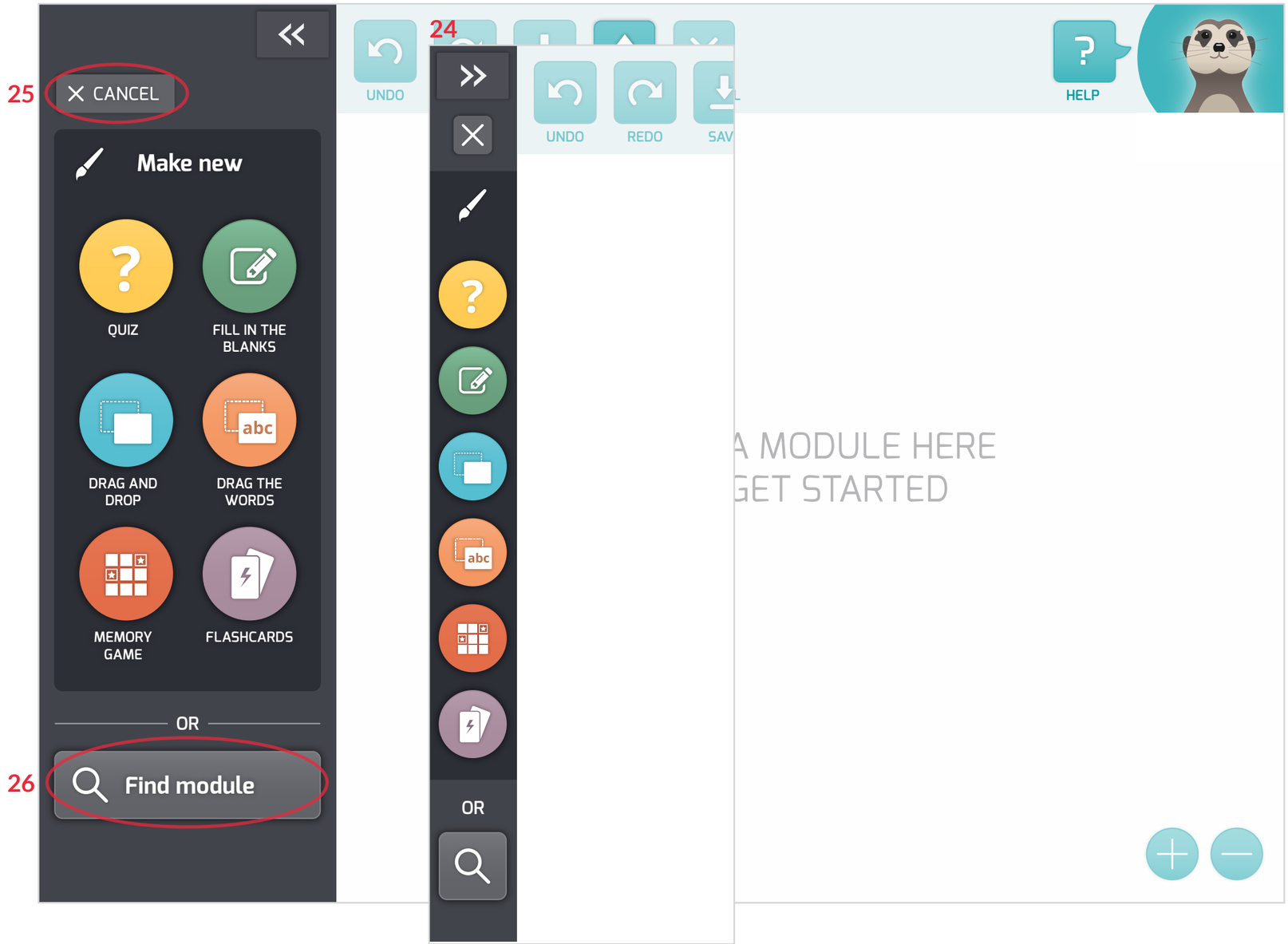
The current interface for making new modules has 6 basic H5P-types. Quiz is the *question set* H5P-type, whilst the others retain their H5P default name and icon.

The 'Make'-panel can be shrunk into a smaller side-panel (24) with the same functionality.

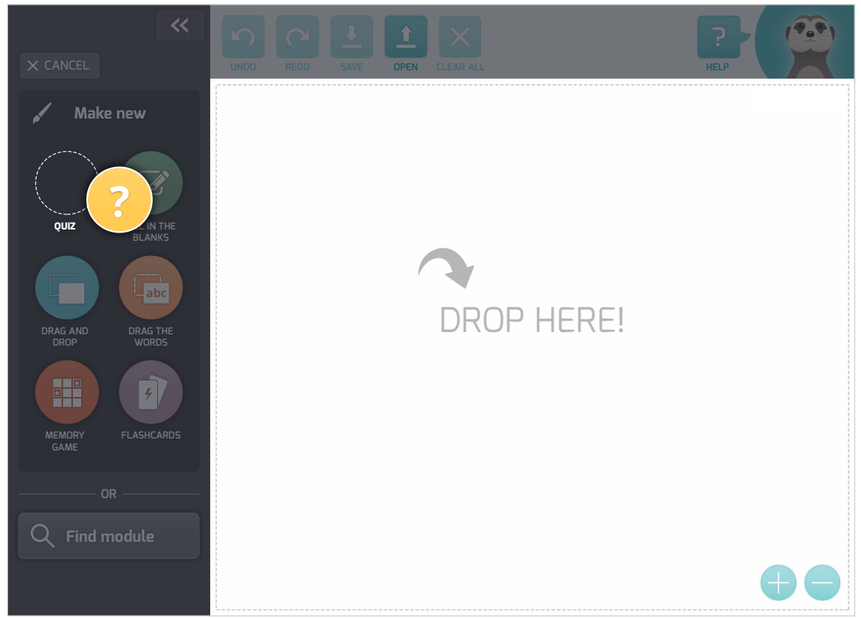
Hitting 'CANCEL' (25) will take the user back to the previous screen. The 'Find module' button (26) is preserved in case the user changes their mind, and want to re-use an existing module instead of making a new one from scratch. It is less prominent and in a different location from the default view, but with the same look and size. This, along with morphing transitions, will indicate that the button is functionally equivalent in both places.

When the user chooses to add a module, they can either drag-and-drop the icon from the side bar onto the canvas (27 - next page), or they can select it and select the 'ADD' button that appears (28 - next page). The whole interface is darkened in response, so that the drop-zone is highlighted.

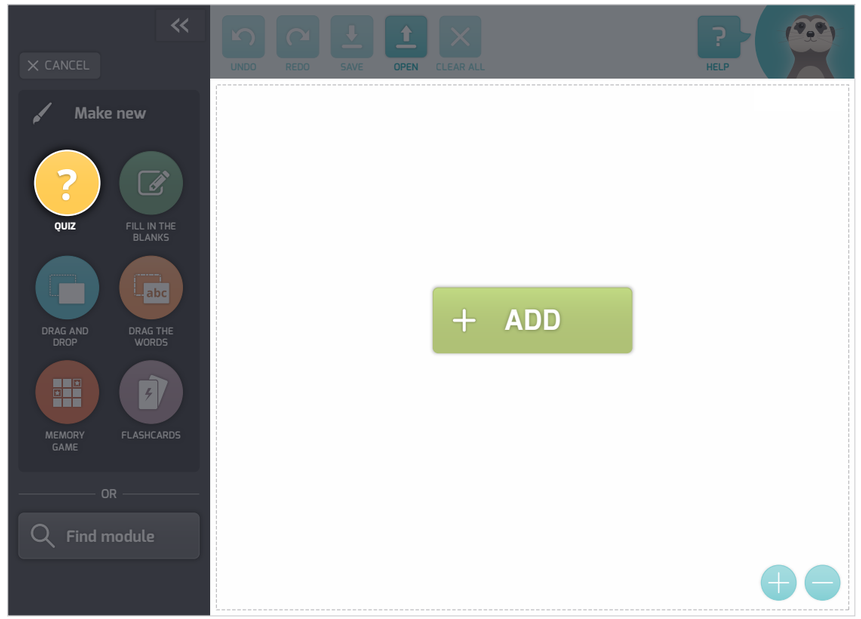
When dropped onto the canvas, the relevant module editor opens (29 - next page). Once the module is authored, it will appear on the canvas (30 - next page).



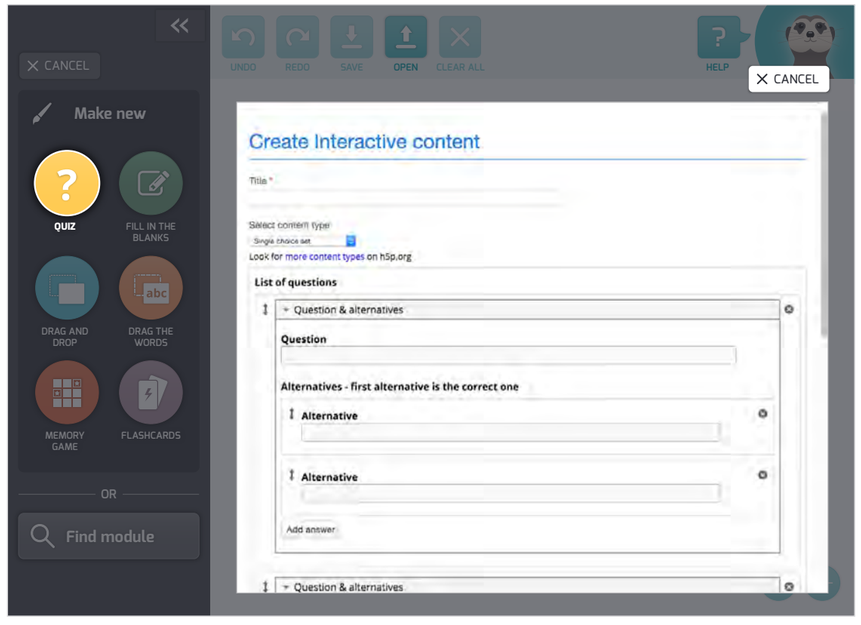
27


CANCEL
Make new
QUIZ
FILL IN THE BLANKS
DRAG AND DROP
DRAG THE WORDS
MEMORY GAME
FLASHCARDS
OR
Find module
UNDO
REDO
SAVE
OPEN
CLEAR ALL
HELP
DROP HERE!

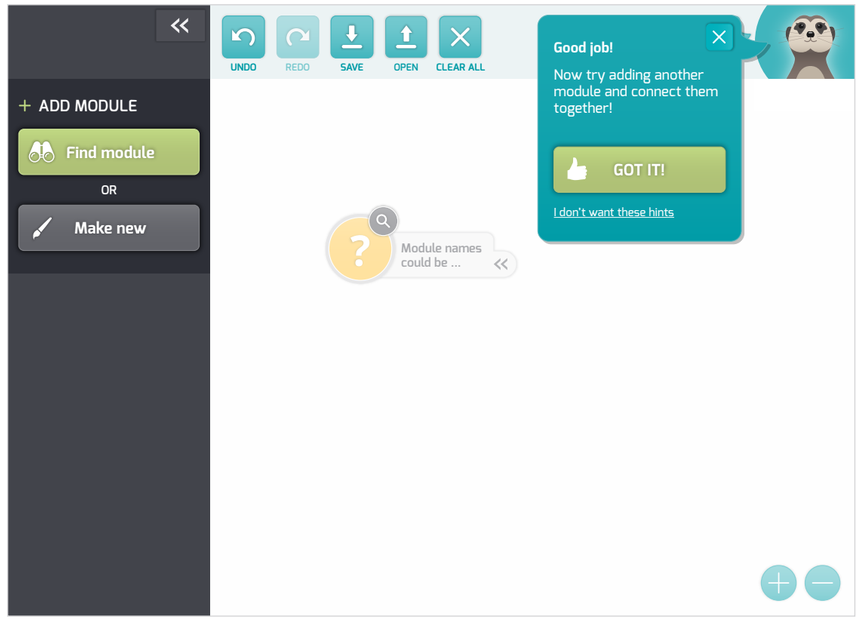
28


CANCEL
Make new
QUIZ
FILL IN THE BLANKS
DRAG AND DROP
DRAG THE WORDS
MEMORY GAME
FLASHCARDS
OR
Find module
UNDO
REDO
SAVE
OPEN
CLEAR ALL
HELP
+ ADD

29


CANCEL
Make new
QUIZ
FILL IN THE BLANKS
DRAG AND DROP
DRAG THE WORDS
MEMORY GAME
FLASHCARDS
OR
Find module
UNDO
REDO
SAVE
OPEN
CLEAR ALL
HELP
CANCEL
Create Interactive content
Title *
Select content type
Single choice set
Look for more content types on h5p.org
List of questions
Question & alternatives
Question
Alternatives - first alternative is the correct one
Alternative
Alternative
Add answer
Question & alternatives

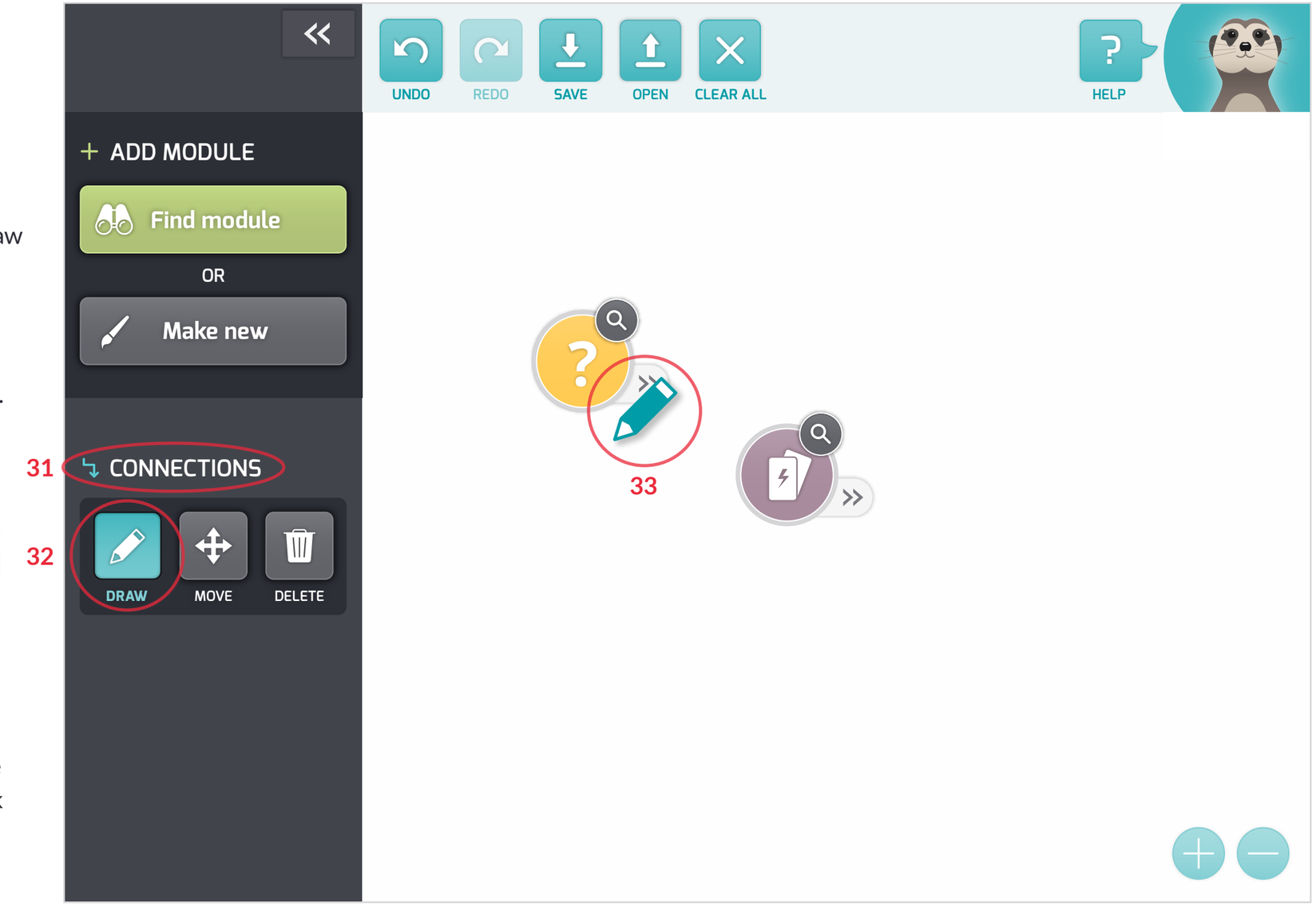
30


+ ADD MODULE
Find module
OR
Make new
UNDO
REDO
SAVE
OPEN
CLEAR ALL
Good job!
Now try adding another module and connect them together!
GOT IT!
I don't want these hints
Module names could be ...

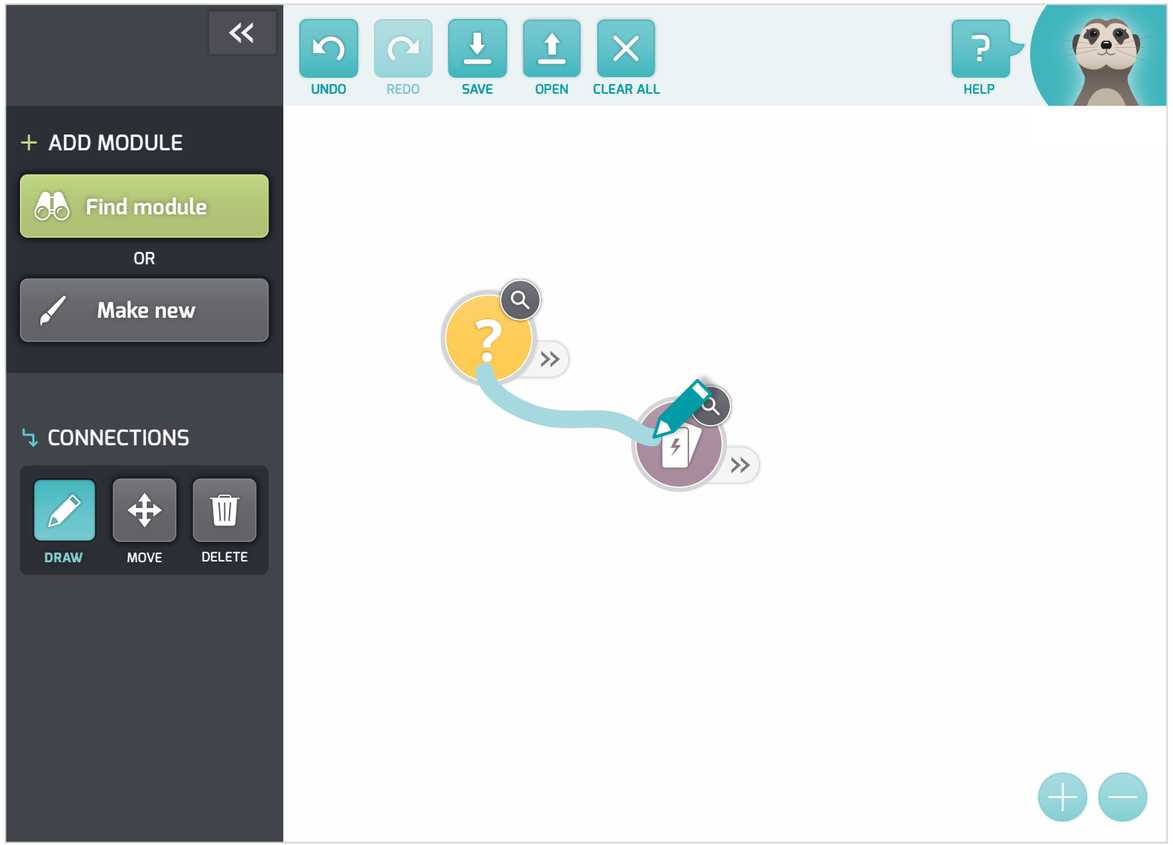
07. Make connections

When two or more modules are added to the canvas, the side-menu will change to include a new action: the 'CONNECTIONS' (31). This is a toggle-menu, where the cursor will change depending on which action is active. E.g. when the 'DRAW' functionality (32) is active, the cursor turns into a pencil (33) that allows the user to draw connections between modules.

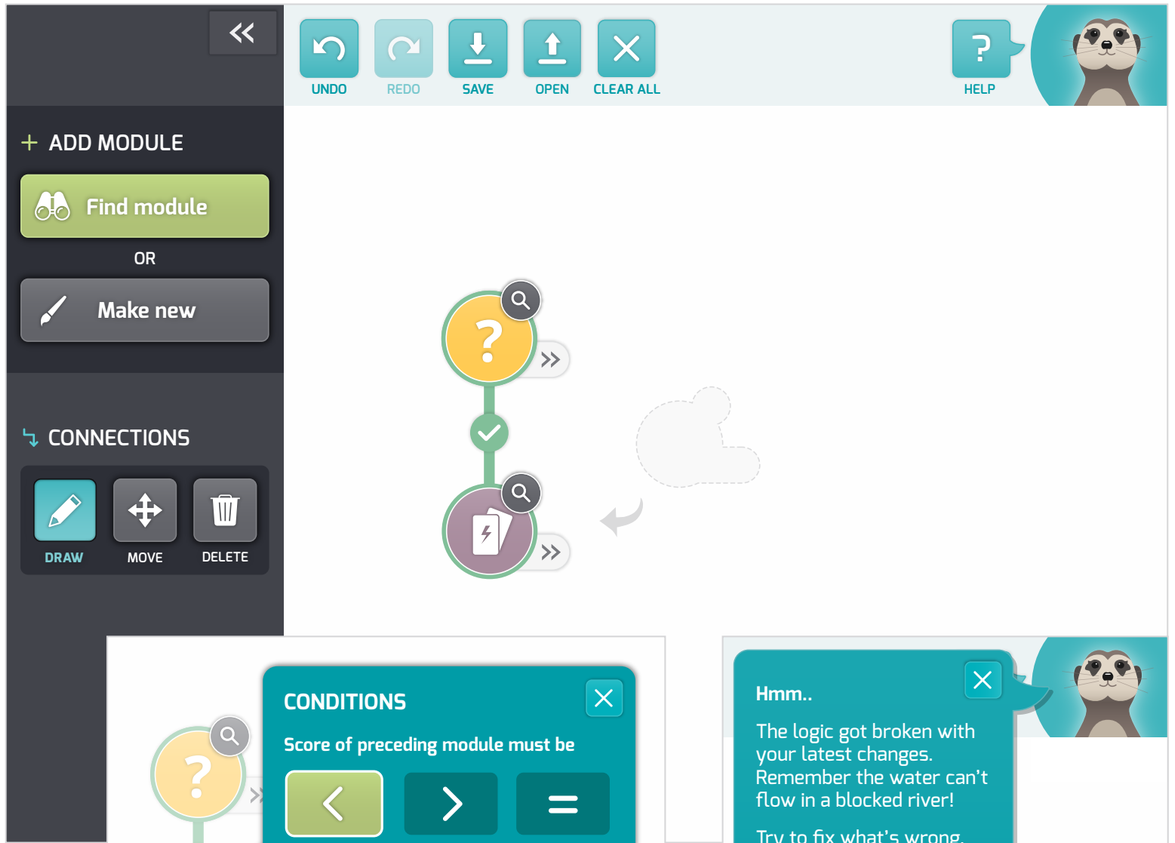
Once the connection is successfully drawn (34 - next page), the system will align the connected modules appropriately to create a more neatly organised system. (35 - next page). With only one module connected to the preceding module, the default condition on this connection will be that the second module follows the first, or that the score of the preceding module needs to be more than 0% (36 - next page). This can be overruled by the user, to allow for complete freedom, but the system will then indicate that the logic is broken for that particular connection, and that they need to fix it. The avatar will show up with a friendly warning the first time this happens (37 - next page), as well as the visual indicators on the module structure itself, with the connection turning red, and a triangle exclamation mark appearing in place of the green tick icon.



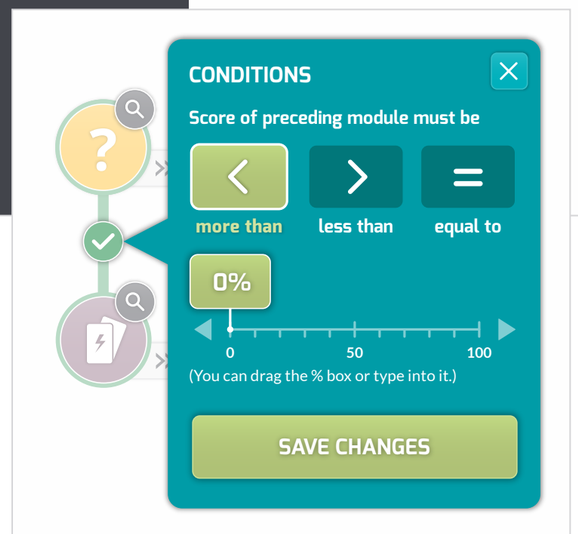
34


UNDO
REDO
SAVE
OPEN
CLEAR ALL
HELP
ADD MODULE
Find module
OR
Make new
CONNECTIONS
DRAW
MOVE
DELETE

35


UNDO
REDO
SAVE
OPEN
CLEAR ALL
HELP
ADD MODULE
Find module
OR
Make new
CONNECTIONS
DRAW
MOVE
DELETE

36


CONDITIONS
Score of preceding module must be
more than
less than
equal to
0%
0
50
100
(You can drag the % box or type into it.)
SAVE CHANGES

37

Hmm..
The logic got broken with your latest changes. Remember the water can't flow in a blocked river!
Try to fix what's wrong, and if you can't figure it out just ask me!
GOT IT!

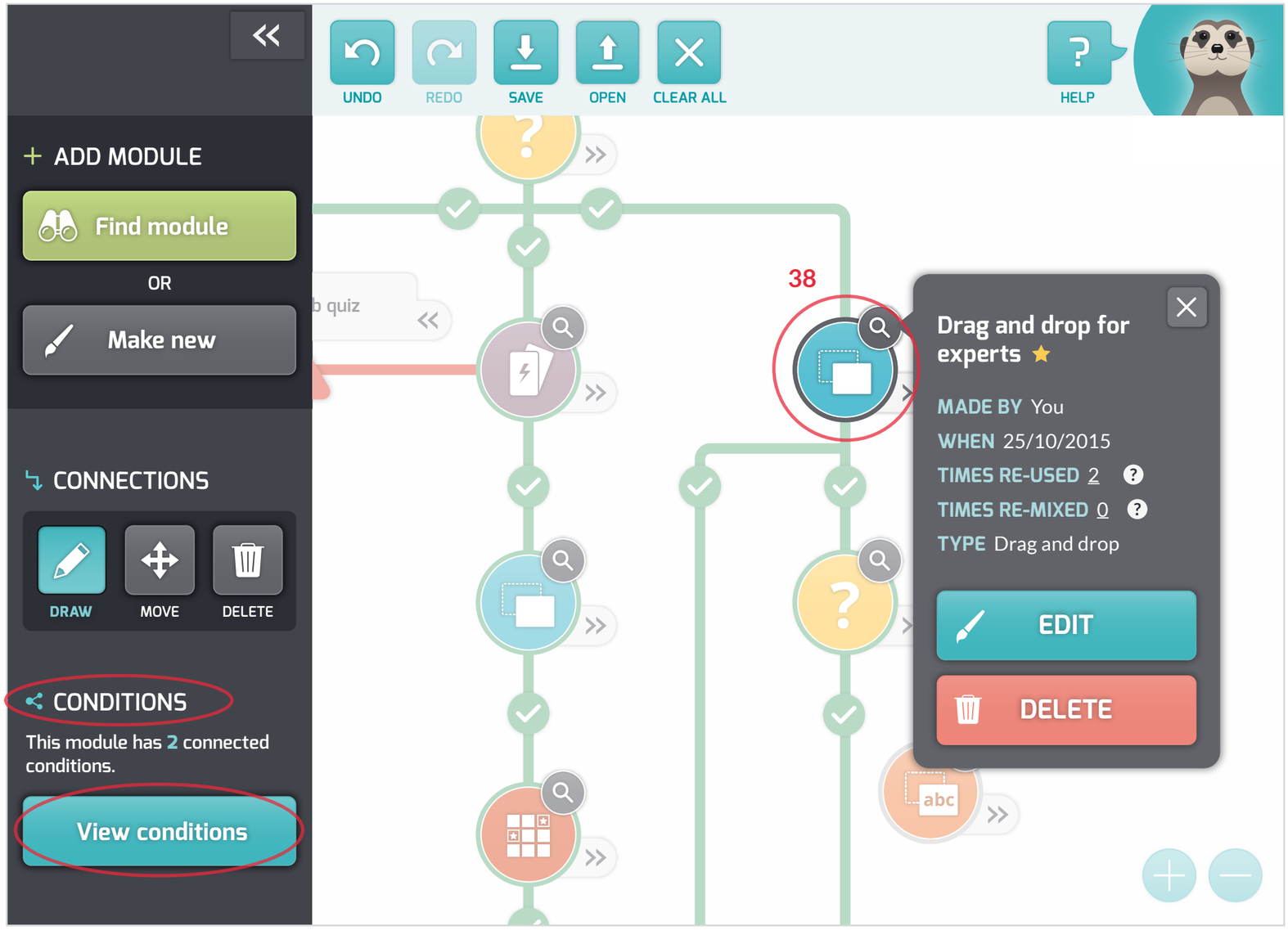
08. Demo

When a module is selected (38), the module info-panel will appear. This allows the user to see details about the current module, such as who it was made by, when it was made, how many times it has been re-used and re-mixed, and what type of module it is. From here, they may also edit or delete the module.

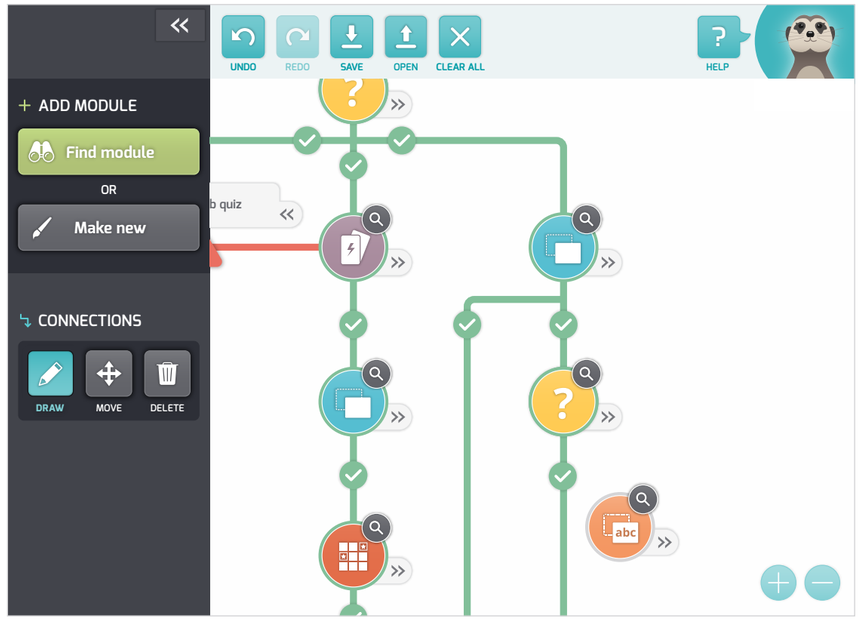
Additionally, when a module that has connections going from it is selected, a new menu item called 'CONDITIONS' (39) appears. This describes how many conditioned connections this specific module has. A button that allows the user to view these conditions (40) also appear. This will highlight the conditions that belongs to this module.

When a lot of modules are added the canvas may look cluttered on a small screen (41 - next page). One way to see more of the canvas, is by shrinking the side panel, (42 - next page) or by zooming out (43 - next page) the canvas. The heads-up display notifications retains their dimensions in order to preserve readability.

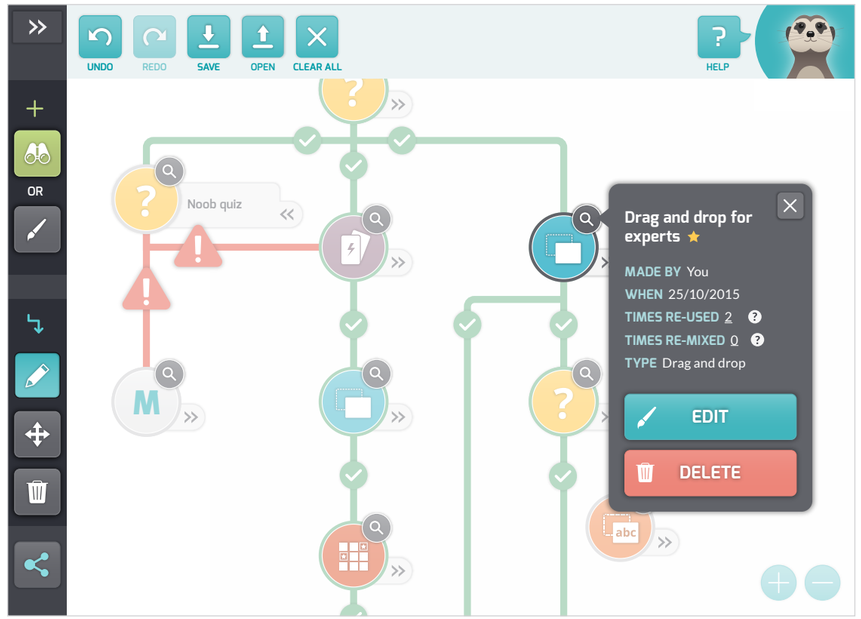
If a user's device has a large screen, the interface will adapt by giving them a larger area to operate in (44 - next page).



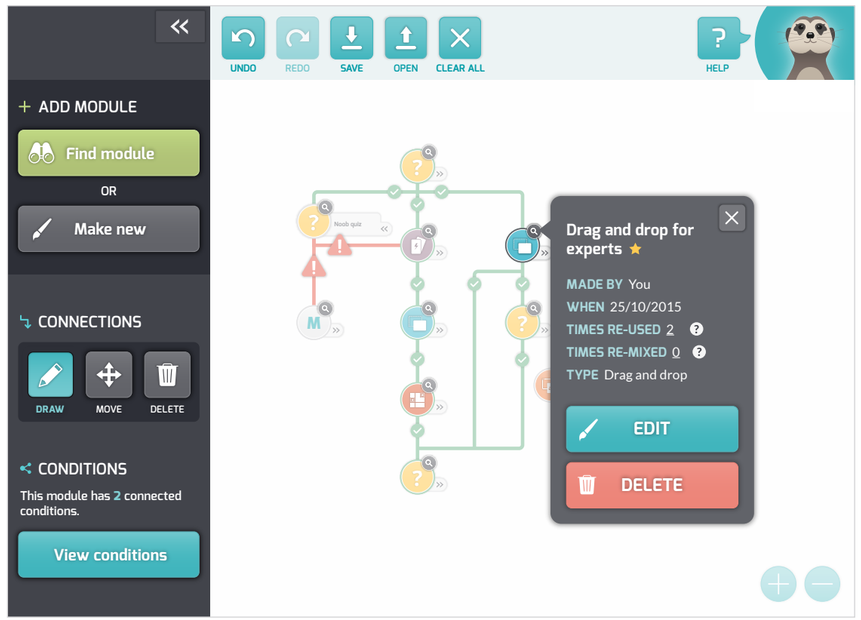
41


UNDO
REDO
SAVE
OPEN
CLEAR ALL
HELP
ADD MODULE
Find module
OR
Make new
CONNECTIONS
DRAW
MOVE
DELETE

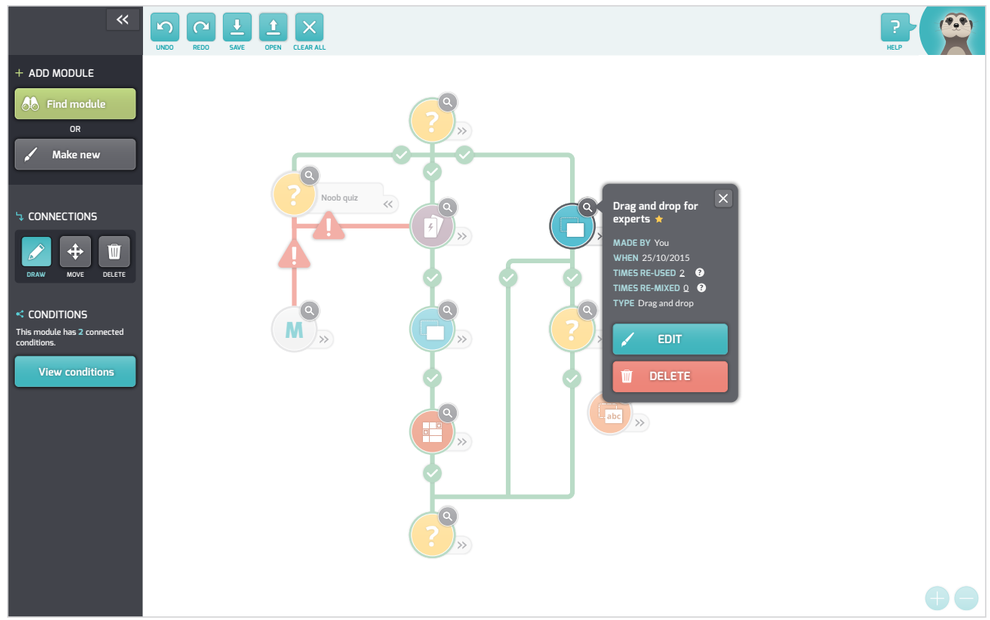
42


UNDO
REDO
SAVE
OPEN
CLEAR ALL
HELP
OR
Noob quiz
Drag and drop for experts
MADE BY You
WHEN 25/10/2015
TIMES RE-USED 2
TIMES RE-MIXED 0
TYPE Drag and drop
EDIT
DELETE

43


UNDO
REDO
SAVE
OPEN
CLEAR ALL
HELP
ADD MODULE
Find module
OR
Make new
CONNECTIONS
DRAW
MOVE
DELETE
CONDITIONS
This module has 2 connected conditions.
View conditions
Noob quiz
Drag and drop for experts
MADE BY You
WHEN 25/10/2015
TIMES RE-USED 2
TIMES RE-MIXED 0
TYPE Drag and drop
EDIT
DELETE

44


UNDO
REDO
SAVE
OPEN
CLEAR ALL
HELP
ADD MODULE
Find module
OR
Make new
CONNECTIONS
DRAW
MOVE
DELETE
CONDITIONS
This module has 2 connected conditions.
View conditions
Noob quiz
Drag and drop for experts
MADE BY You
WHEN 25/10/2015
TIMES RE-USED 2
TIMES RE-MIXED 0
TYPE Drag and drop
EDIT
DELETE

REFERENCES

1. James Ignacio. Linear vs. Non-linear. https://ionelearning.wordpress.com/2013/07/12/linear-vs-non-linear/, 2013. Visited 2015-10-09.
2. Alexander Berntsen, Stian Ellingsen & Emil H Flakk. Enabling Learning By Teaching: Intuitive Composing of E-Learning Modules. 2015.
3. Jerry Cao, Kamil Zieba & Matt Ellis. UXPin: Interaction Design Best Practices. Mastering Words, Visuals, Space. E-book