Getting a Grip on Exploratory Testing: Exercises

Many competent testers use one or two exploratory approaches, but are not comfortable working outside that range.

In the course *Getting a Grip on Exploratory Testing*, I teach a variety of exploratory techniques. By exposing participants to a range of techniques and disciplines, I have tried to put testers in a position where they gain an appreciation of their own style, and of the range of options available.

To teach the techniques, I have developed an number of interactive machines. Each of these machines has been designed primarily to help teach a specific technique – although other trainers may find different uses.

I have decided to make the machines available through testingeducation.org. This document is part of that distribution, and contains teaching notes for each machine. Exploration is a process of learning, so if you have received this document as part of a class in exploratory testing, you need to know that you will get much more from the exercises if you put this document aside.

The notes for each machine contain a brief description of the exercise I use to teach a technique or discipline, and an example of work that a keen student might produce. Separate sections cover what you should know about the deeper structures of the machine, and what you might want to look out for while teaching to assess the progress of a class.

If you still plan to read the teaching notes without doing the exercises, I have to assume that you are the kind of person who does yesterdays crossword by looking at the answers in today's paper.

My thanks to the people – particularly Alan Richardson, James Bach and Robert Sabourin – who have encouraged me to develop these exercises, and to the colleagues and course participants who have taught me how to teach them.

Technology

The machines have been developed in Flash 5, allowing:

Compatibility across browser, OS and platform.
Use without installation - can be run from CD or network
Small size

Using the exercises

Open index.html in a browser
Use the test machine in the support section to check that you have the flash plugin.
Follow the links for the four exercise machines A-D

Role of the trainer / coach

You need to present the technique or discipline, coach participants as they test during the exercise, and facilitate discussion after the exercise.

Different exercises puzzle different people. You have to pay close attention and decide whether to intervene, or let to let a participant arrive at a solution themselves.

Contact

	James Lyndsay
Company:	Workroom Productions Ltd.
Website:	http://www.workroom-productions.com
Email:	jdl@workroom-productions.com
AIM:	workroomprds
Tel:	+44 (0) 20 7372 6986
Mobile:	+44 (0) 7904 158 752

Machine A: Input / Output / Linkage

Summary: Disciplined exploration of an abstract machine.
Takes: Between 10 and 20 minutes, including discussion.

Introduce idea of active, systematic exploration - and a framework to help build a simple model. Help re-consider concepts of input and output. Encourage imaginative extensions of diversity of input / output, types of dependency. Share different approaches to GUI discovery.

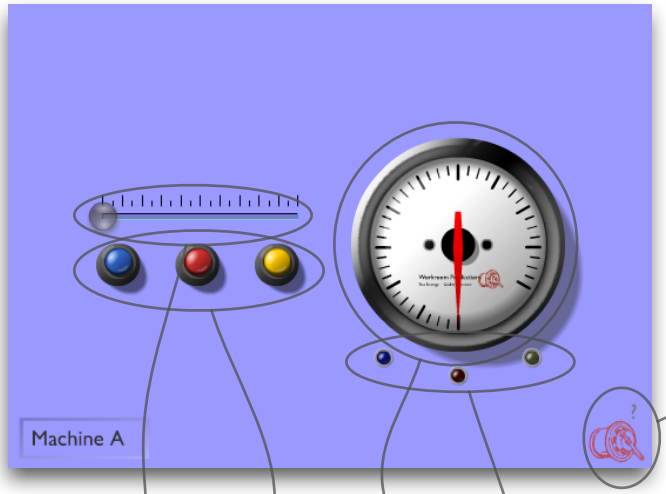
Assess and assist group progress:

Simple ideas of 'an input is a button' should give way to more complex concepts.

On analysis, 'random' clicking will crystallise into techniques. Different individuals will have different approaches – notice these and encourage participants to try novel techniques.

Participants may not believe you about the logo / '?'. Encourage them in this!

The group may need help to model the linkage. Challenge them to increase their certainty.



Teacher Awareness

The machine should be simple to explore. The buttons, lights, slider and dial are clear and act independently. There are no hidden tricks.

The blue button is a toggle, the red stays down as long as it is pressed, and the yellow is transient.

The lights correspond to the state of the picture button, not to the state of the mouse button.

The blue button is the only one that allows its state to be fixed for further action/testing. Its state (and by implication that of the blue light) has no effect on any other input or output.

Paired testing is highly effective

The '?' and red logo in the bottom right corner respond to rollover/mouseclick. They have no direct effect on the machine (and are a common feature). *I like to let people find these before I tell them – it helps them think of different surprises.*

There are no known bugs

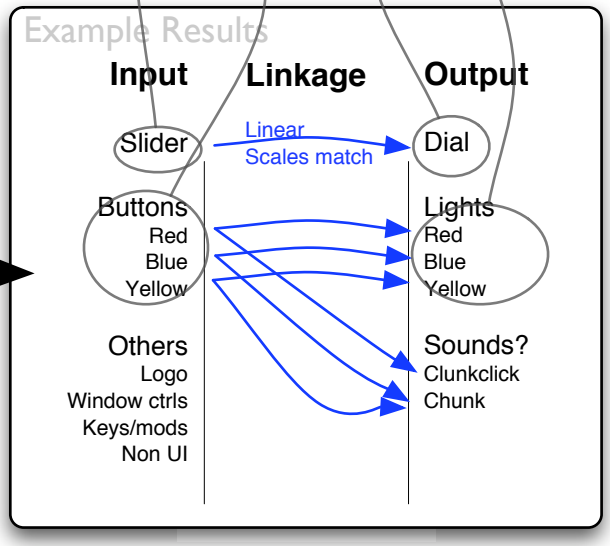
Suggested exercise:

Identify 'inputs' and 'outputs' (3 minutes).

Discuss what makes an input, what makes an output. What others might there be?

Identify links and dependencies between identified input and output. (3 minutes)

Discuss types of linkage - one-to-one, multiple dependencies, linearity



Extending:

Discuss the differences between input if seen as information / stimulus, and input if seen as something that can be stimulated

Introduce resizing, ctrl-click, keypresses, platform etc.

Discuss APIs and automation.

Discuss discovery of input/output during exercise, and effectiveness (or not) of systematic approaches.

Machine B: Event / Behaviour / Information

Summary: Disciplined exploration of an abstract machine.
Takes: Between 20 and 30 minutes, including discussion.

Introduce a second exploratory framework. Highlight that testers/users are not always the direct cause of an observed effect. Different behaviour / response indicates different state. Imagining underlying system – making a model modelling and testing cause / effect. Use state model to assist exploratory testing.

Assess and assist group progress:

Once the machine stops for the first time, some delegates may think they have broken the machine. Some, perhaps feeling they have proved their prowess, will go no further. Gently ask them to reproduce the bug, and to describe the actions that they took – further investigation may lead them to question their initial judgement.

Can the group tell you about their theories about what the machine might be doing, and why it stops?

Ask the group about the lights – how is the green one lit?

Teacher Awareness

The machine starts as soon as it is opened, and the central dial spins until the machine stops. While the machine is runnning, either the left or right dial spins – the blue button toggles between them. The machine stops when either of the outer dials reaches its clockwise maximum.

There is no 'reset' button: Once the machine stops, the user needs to take external action (i.e. reload in browser) to start the machine again

The machine has been designed to be a poor subject for input/output/linkage – and to also introduce testers to the idea that they must observe events that are not triggered by their own actions.

There is a (noisy) bug that can be observed if the blue button is pressed as one of the dials reaches the end of its travel. This bug is hard to reproduce – modelling the state transitions can help focus attention and allow it to be observed more reliably.

Note: there are two ways of leaving a state, and they have non-exclusive triggers. The bug appears when the two exits happen close together – the state model shows this potential.

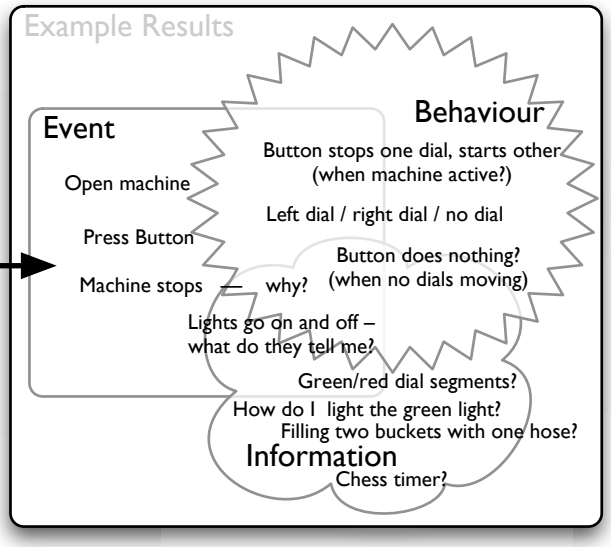
Suggested exercise:

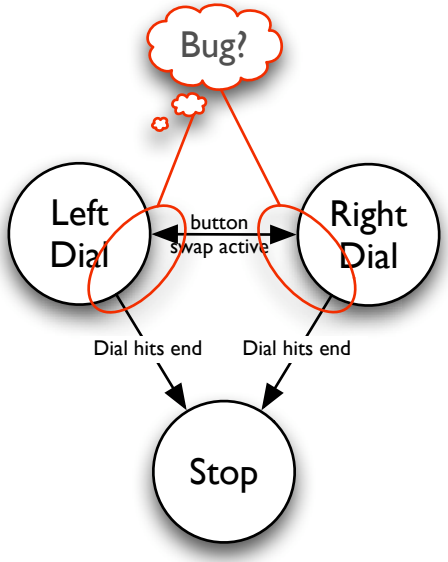
Identify the events that affect the machine, the behaviours it displays, and any information that gives you clues / seems important (6 minutes).

Discuss events that are not triggered by testing / testers

Use information (unanswered questions, models etc.) to imagine links between events and behaviours. (3 minutes)

Discuss similarities between behaviour and state. Draw state diagram and hunt bugs. Discuss different state diagrams that could be used.





Machine C: Testing against external expectations

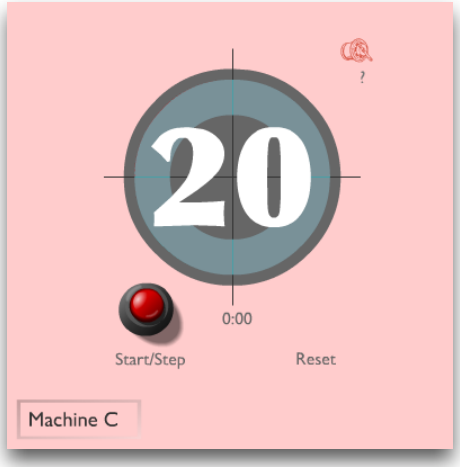
Summary: Disciplined exploration of something with a known function.
Takes: Between 15 and 30 minutes, including discussion.

Finding bugs in something that has an accepted way of working. Ways that planning and focussed error-guessing can help find bugs, and hinder the discovery of others. Experience of observing unexpected problems and following leads. Judging faults.

Assess and assist group progress:

Some people may spend the entire period attacking the input. Others may not change the timer to an appropriate value for the short time available for testing. Encourage delegates to take diverse approaches, and to design tests fo fit the situation.

Judgement of a bug is key to exploration – without judgement, it is hard to consider which of many paths to follow. Testers that concentrate entirely on ambiguous characteristics may need to be challenged to find more valuable faults.



Teacher Awareness

The logo and ? have useful information – you may want to reveal this to the class.

At this stage, delegates should be thinking of test design as well as exploration. You may want them to consider risks, likely faults, or particularly significant errors. They should also consider the constraints of the test parameters; a few minutes will not be enough to expose some issues, what can they say about coverage?

As individuals find bugs, others in the group may be distracted from their own paths (particularly if you're using pair testing with ebullient testers). Encouraging competition can motivate discovers to keep their discovery secret until the period of discovery is over.

Alternatively, you may wish to split the group into two parts – the away group thinks about risk and methods without testing, while the discovery group think about the principles that might help guide the others to discover bugs more quickly.

Changing the time on the PC Is a interesting attack...

Suggested exercise:

Find bugs – and justifications about why the characteristics you have identified are indeed bugs (5 minutes)

Discuss the methods used to discover the bugs – were bugs found because of carefully-aimed tests, or perceptive observation?

Discuss the discovery of the bugs; what bugs were found first? Did different people find different bugs?

Discuss the bugs themselves – are they all bugs?

Known bugs

The circular timer runs 10% slow [this can be seen by comparison with your watch – or by looking at the numerical timer. If you set the timer to a minute, you can see this problem in the first 15s]

'Tick/Tock' is layered in front of the logo / ? mark text . Note – 'paused' is behind.

Start/Step – should be Start/Stop [is this a typo? If 'Step' is intentional, is it still a bug]

No ambient/aural indication when timer reaches 0. [also missing from production version]

Potential usability issues

Reset works as a button – but without the graphic. No explanation of elements – particularly input text box. Cursor appears in text box. Inconsistent response to tab. Nasty pink.

By design

Can't reset the timer while it's going [to avoid accidental reset]. Pausing the timer doesn't stop the numerical timer [measures elapsed time]. New timer appears after timer counts to 0 [requirement to show time since timer reached 0]. Can't stop count-up timer [ask why this might be necessary].

Machine D: Discover and judge inconsistencies

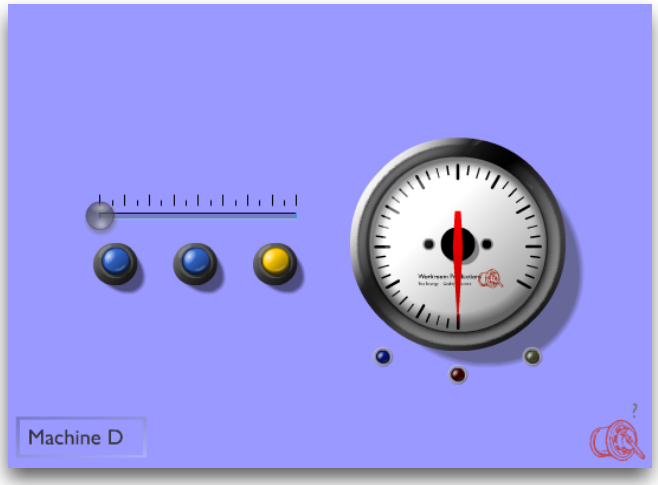
Summary: Compare two similar machines
Takes: Between 15 and 30 minutes, including discussion.

Re-use and extension of existing method. Modelling failure / difference, designing tests to verify model. Judgement of differences / bugs. Ways that repeat testing is influenced by what has gone before – use of prior test results to guide test design.

Assess and assist group progress:

What methods are in use to explore the machine? Are people making another map of input/output/linkage? Comparing machines, or maps? Are they working from memory, or do they have their subjects open side-by-side?

In judging the differences, people will have to build models of the internal logic/connections. How are they building these models? Can they be drawn/articulated? What tests can be devised to expose the differences?



Teacher Awareness

The class needs to know that Machine A and Machine D are different versions of the same machine. However, there is no information about which is the earlier version. You might want to discuss the influence that this information would have on judgement of bugs etc.

Non-linear response is not visible at the boundaries of slider travel. However, it's easy to see the difference in the middle, or while moving. How does this relate to BVA / ECP?

The machine is limited by possible interactions. If it had a physical interface, or if its code/ circuitry/mechanics were exposed, different tests would be available. You might want to encourage the class to think of these tests.

Having two machines open at the same time has no designed effect – but are people considering it?

Actions that didn't work on Machine A may not even be tried in Machine D – has anyone tried hitting keys, resizing the window etc.? Perhaps there are more differences? Try tab...

Suggested exercise:

Identify differences between the machines. For each of the differences, make a judgement as to whether the difference is a bug. Justify that decision. (6 minutes)

Discuss the methods used to discover differences.

Discuss whether new tests were needed to justify assessment of differences as bugs. What influenced the test design?

Discuss the judgements made.

Differences

Slider scales differ	May be a bug – dial scales are the same in A and D. In A, there is a correspondence between the dial and slider - in D, that correspondence is broken.
D Dial has a non-linear response to slider	Unknown – there is no evidence to indicate whether this is required or consistent behaviour.
Middle button is red in A, blue in D	Bug. Button works like a red button, but is blue. The two blue buttons work differently in D. Is it a red button that is the wrong colour, or a blue button that works wrongly? Given the button/light colour correspondence, which remains the same for blue and yellow, it is likely to be a red button that is the wrong colour.
Yellow button affects blue light in D, but not in A	Unknown – not enough information to judge. Simple models of the interaction might be: a) Only one light can be on at a time; b) the yellow button inverts the blue light; c) the yellow button disconnects the blue light; d) the yellow and blue lights can't be on at the same time. (a) and (b) can be disproved – but it is not possible to manipulate the machine directly to examine (c) or (d).