Black Box Software Testing
Spring 2005
SCRIPTING:
AN INDUSTRY WORST PRACTICE?

Cem Kaner, J.D., Ph.D.
Professor of Software Engineering
Florida Institute of Technology
and
James Bach
Principal, Satisfice Inc.

These notes are partially based on research that was supported by NSF Grant EIA-0113539 ITR/SY+PE: "Improving the Education of Software Testers." Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Scripting

- COMPLETE SCRIPTING is favored by people who believe repeatability is everything and that with repeatable scripts, we can delegate to cheap labor.
- Here's the kind of detail you often find in scripts

1 __ Pull down the **`File`** menu
2 __ Select **`Print`**
3 __ The program displays the **`Print`** dialog
4 __ Enter 2 in **`Number of copies`**
5 __ Enter a check in **`Collate`**

- The idea (the fantasy) is that the script specifies the test so precisely that the person following it can be a virtual robot, doing what a computer would do if the test were automated.

Scripting: Another format

Step #	Check ?	What to do	What to see	Design notes	Observation notes
1.	____	Pull down File menu	File menu down	This starts the blah blah test, with the blah blah goal	

This is the format I used at WordStar, long ago (back when I thought scripting was a good idea).

Scripting: Perceived benefits

Proponents believe scripting offers several benefits:

- **Repeatability**: Different testers, following the same script, will do the same things and achieve the same results. This provides a level of process control for testing.
- **Scalability of expertise:** One expert tester writes the scripts and junior testers execute them. We leverage our best people.
- **Ability to add inexpensive staff**: The juniors cost less than the senior.
- **Supports outsourced testing:** Cheap labor far away can follow scripts.
- **Ability to add staff late in the project**: scripts provide structure for the tester who is unfamiliar with the product. If the project schedule falls behind, we can add testers and keep them busy without too much training cost.
- **Training**: The script acts as "training wheels" for the new tester. After several months of following a wide range of scripts, the new tester will have learned by example a lot about the program and how to test it.

So, what's wrong with scripting?

Let's try an example: A simple arithmetic exercise

Problem	We have a vat of marbles and three jars, with the following capacities			Obtain this many marbles
1	21	127	3	100

Arithmetic exercise

Problem	We have a vat of marbles and three jars, with the following capacities			Obtain this many marbles
1	21	127	3	100
2	14	163	25	99

Arithmetic exercise

Problem	We have a vat of marbles and three jars, with the following capacities			Obtain this many marbles
1	21	127	3	100
2	14	163	25	99
3	9	142	6	121

Arithmetic exercise

Problem	We have a vat of marbles and three jars, with the following capacities			Obtain this many marbles
1	21	127	3	100
2	14	163	25	99
3	9	142	6	121
4	18	43	10	5

Arithmetic exercise

Problem	We have a vat of marbles and three jars, with the following capacities			Obtain this many marbles
1	21	127	3	100
2	14	163	25	99
3	9	142	6	121
4	18	43	10	5
5	23	49	3	20

Arithmetic Exercise

- What we see is that people adapt to procedures.
- Once in the mindset of a set procedure
  - people are likely to miss or ignore blatantly obvious information that tells them that:
    - a different procedure would be more efficient (in this case)
    - this procedure is not applicable to this situation
    - something else happened while they were following a procedure, that isn't within the scope of consideration of this procedure

This contrasts with a common assumption

- The assumption
  - When someone is following a standard procedure, they'll keep their eyes open and notice unusual things, notice variation, et cetera

This contrasts with a common assumption

- The assumption
  - When someone is following a standard procedure, they'll keep their eyes open and notice unusual things, notice variation, et cetera
- **Have you ever noticed?**
  - What happens when you drive to a new place
    - from a map, or in some other way, making your own way
    - with a navigator who says, "turn left next light" etc
  - If you try to go there again, do you remember the route?

This contrasts with a common assumption

- The assumption
  - When someone is following a standard procedure, they'll keep their eyes open and notice unusual things, notice variation, et cetera
- **Demonstration**
  - We'll do this together in the live class
  - If you're taking this class remotely, please skip to the *Scripting Part B* slide set

This contrasts with a common assumption

- The assumption
  - When someone is following a standard procedure, they'll keep their eyes open and notice unusual things, notice variation, et cetera
- **The contrast**
  - The examples suggests the reverse, that people get locked into procedures, focus more on following the instructions than on the system being worked through those instructions.
- **My assertion**
  - When people embrace procedures, they often become automatons.

Another example

How do Fornix-Fimbria Lesions Affect One-Way Active Avoidance Behavior?

Cem Kaner, Bob Osborne, Harvey Anchel, Mark Hammer & Abraham H. Black

McMaster University

86th Annual Convention of the American Psychological Association

Toronto, Canada, August 28, 1978

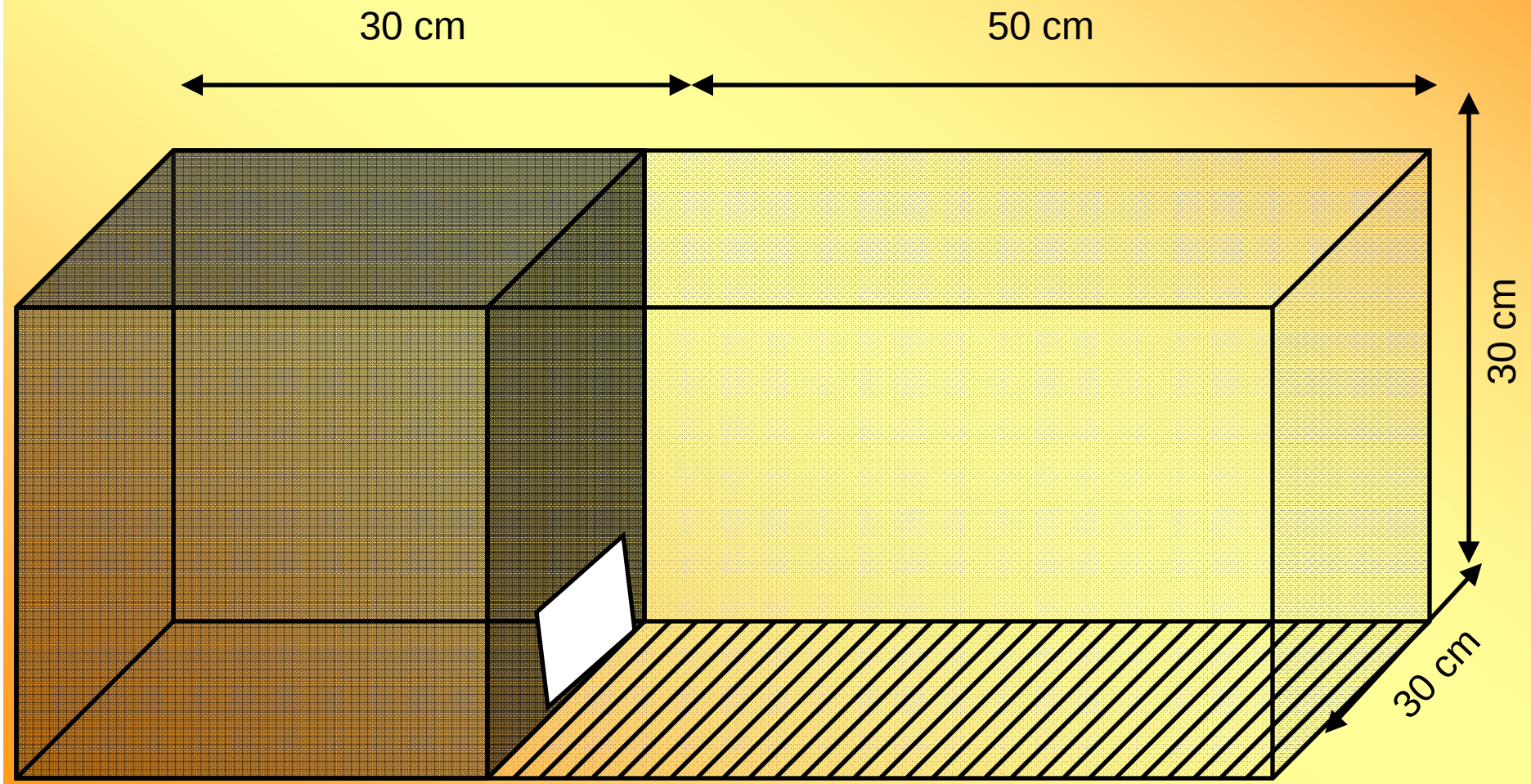
- Study compared normal rats with rats with fornix-fimbria lesions that destroyed the hippocampus.
- Through a series of pilot studies, the experiment simplified and simplified until it was finally a big white box (with an electrical grid) for a floor and a smaller black box, with a metal door between them.
- Rats were placed in the white box, given a signal, and had to move to the black box within 5 seconds or get shocked.
- Normals appeared to know where they were and where they wanted to go. The lesioned rats appeared to have learned a fallible procedure for escaping.

Normal, hooded rat (rattus norvegicus) raised in our lab

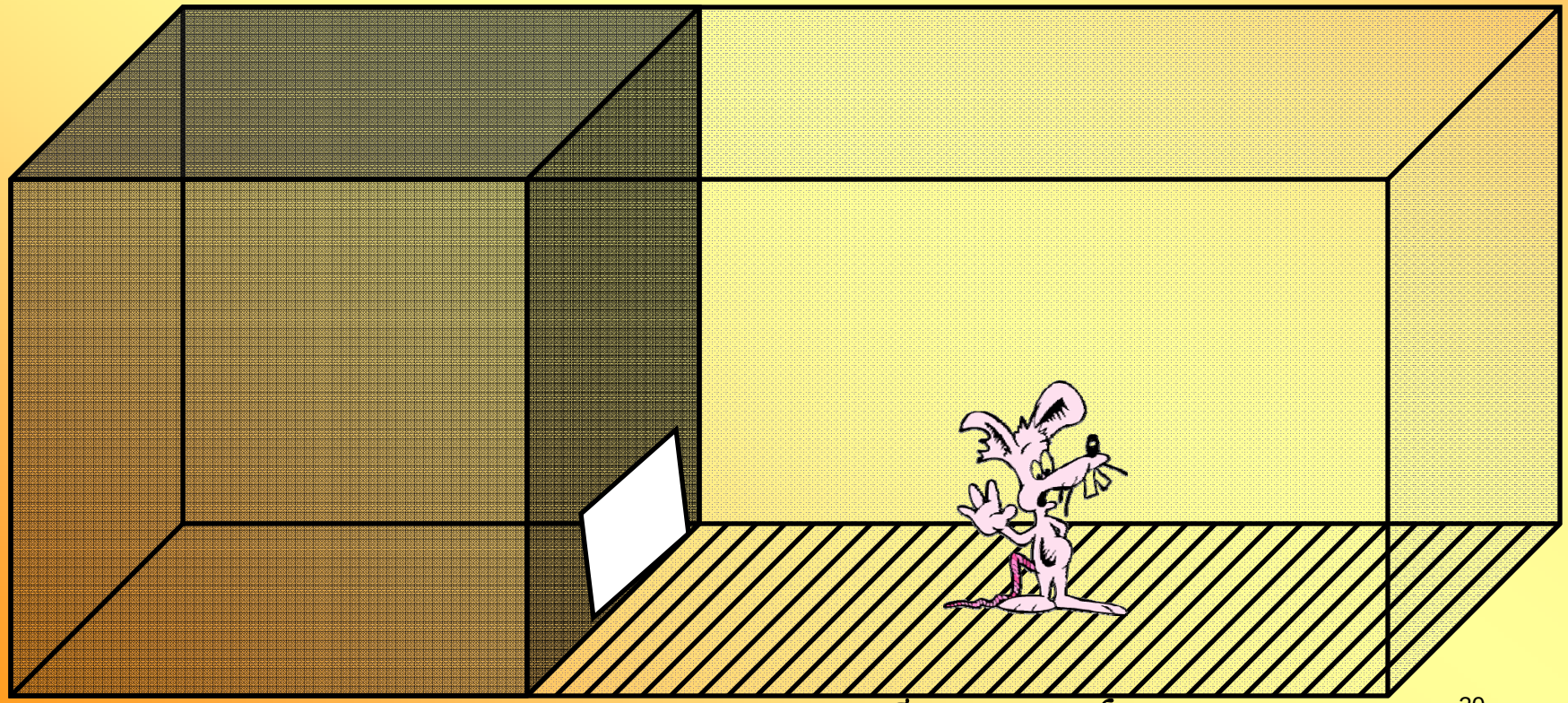
Lab rat with a lesion of the fornix, which causes damage to the rat's hippocampus.

- *affects (among other things) the rat's short term memory and its ability to interpret its location in space*

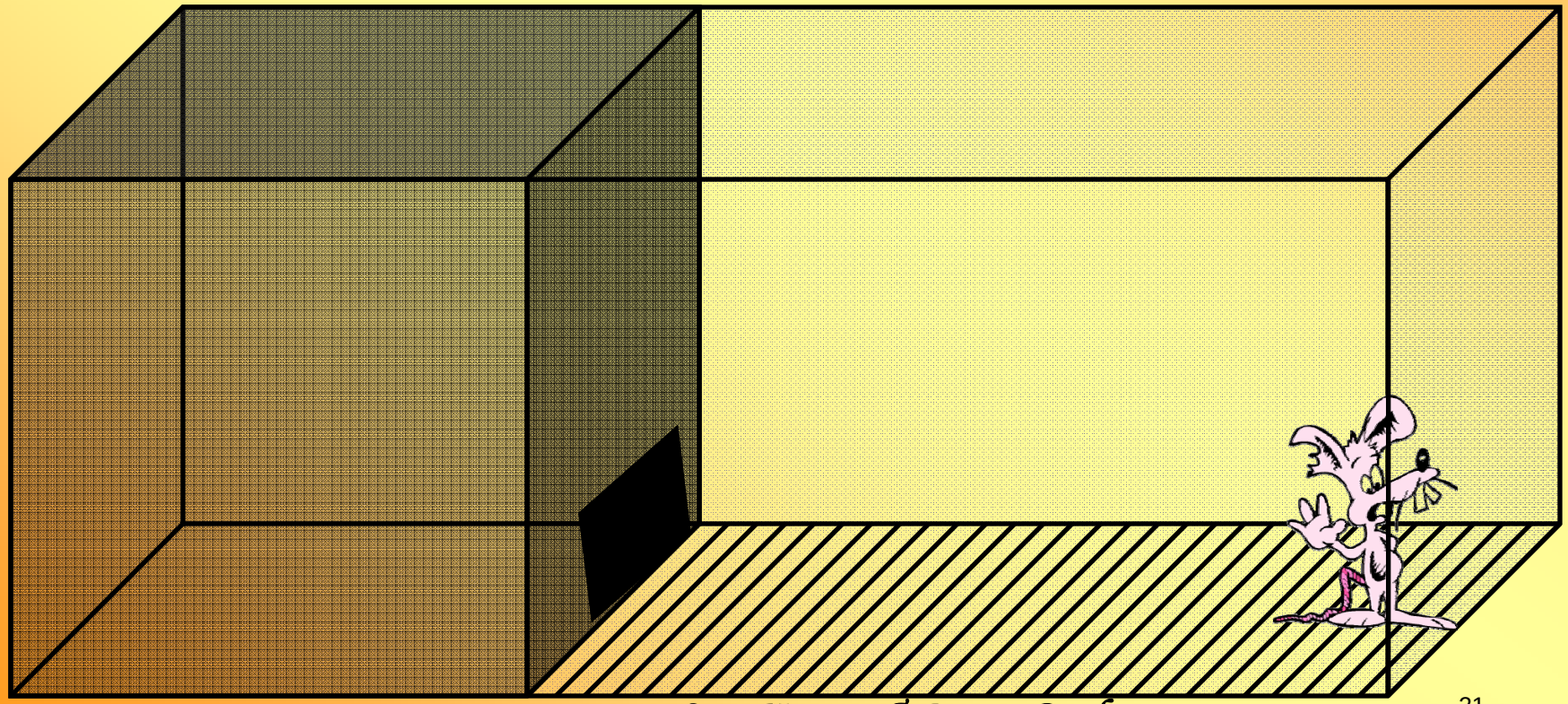
The experimental chamber



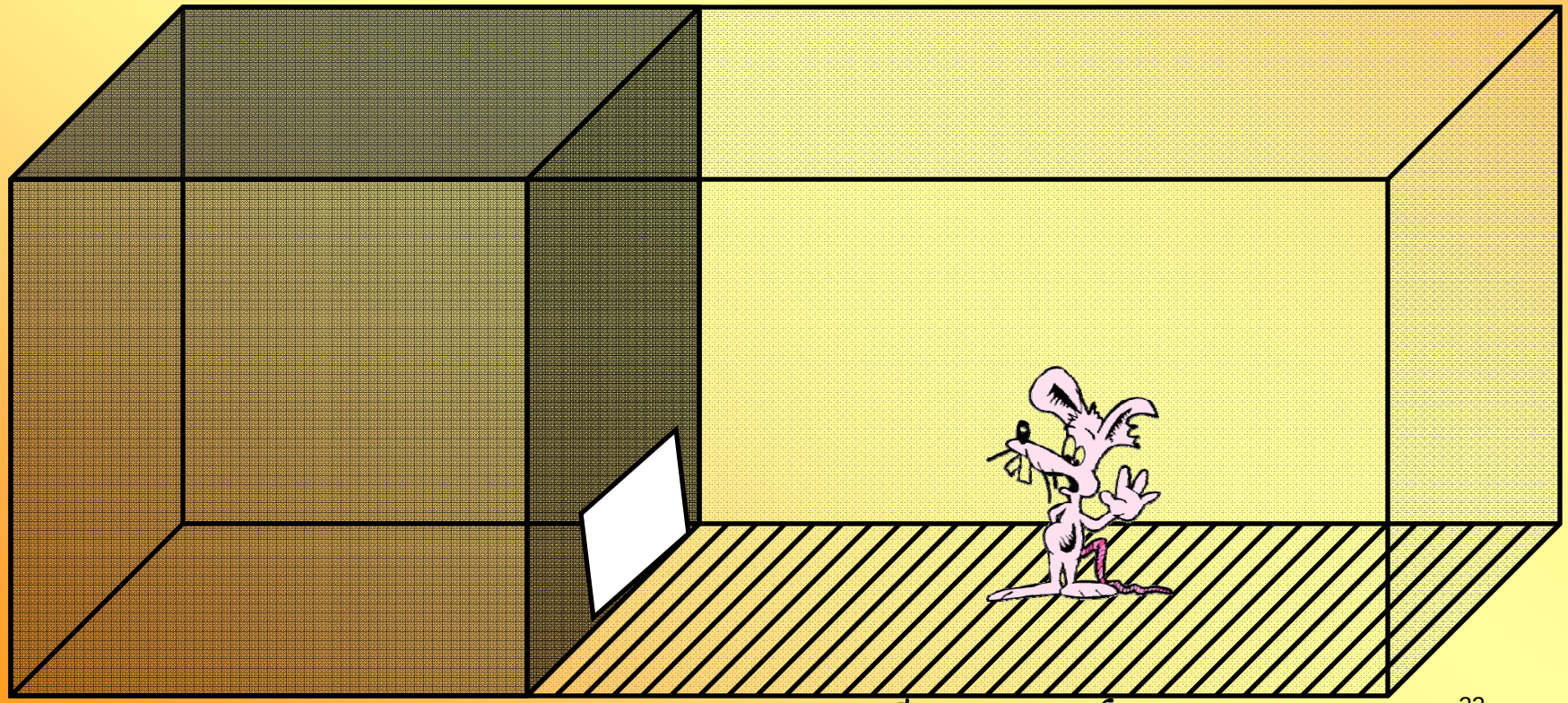
1 minute of exploration, then start after he enters the black side



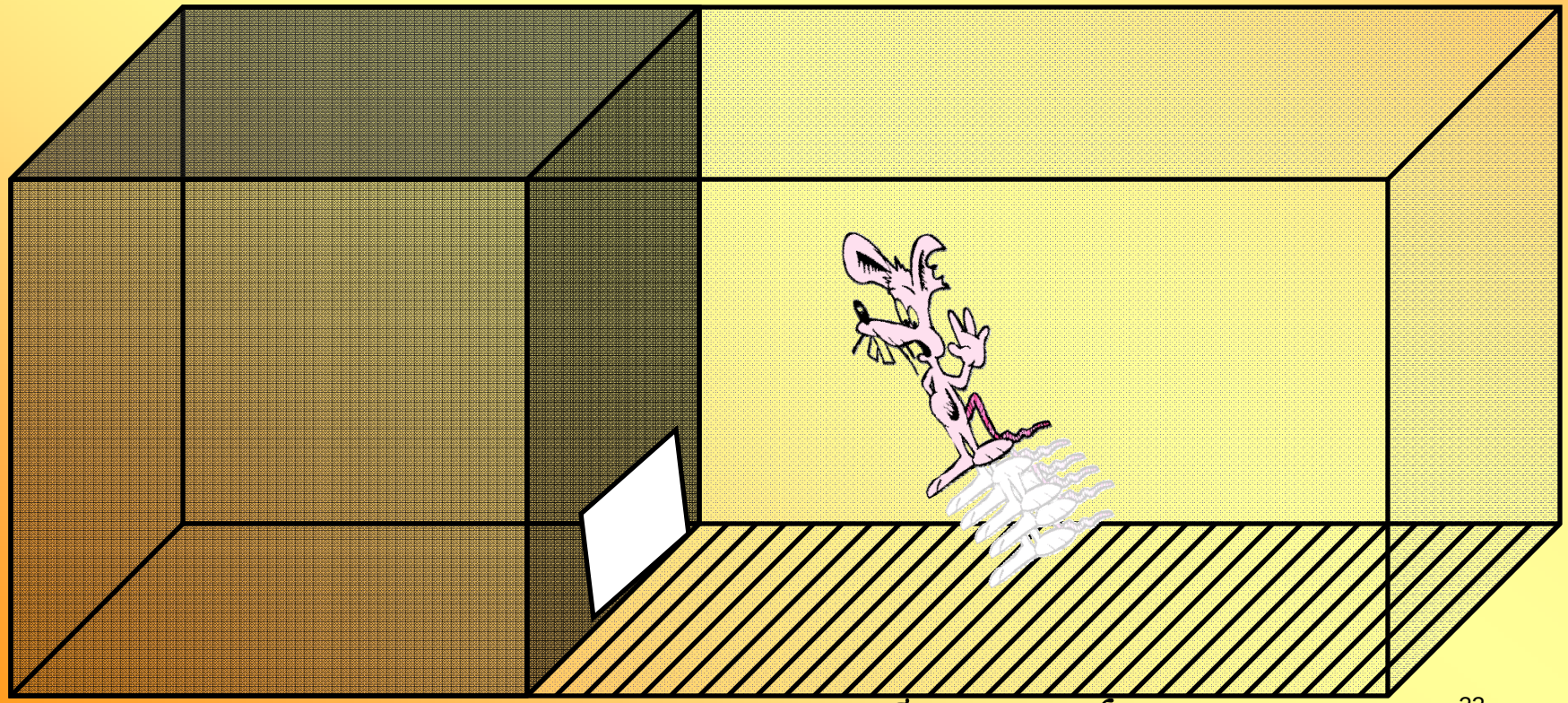
Put him in the white section, away from the door



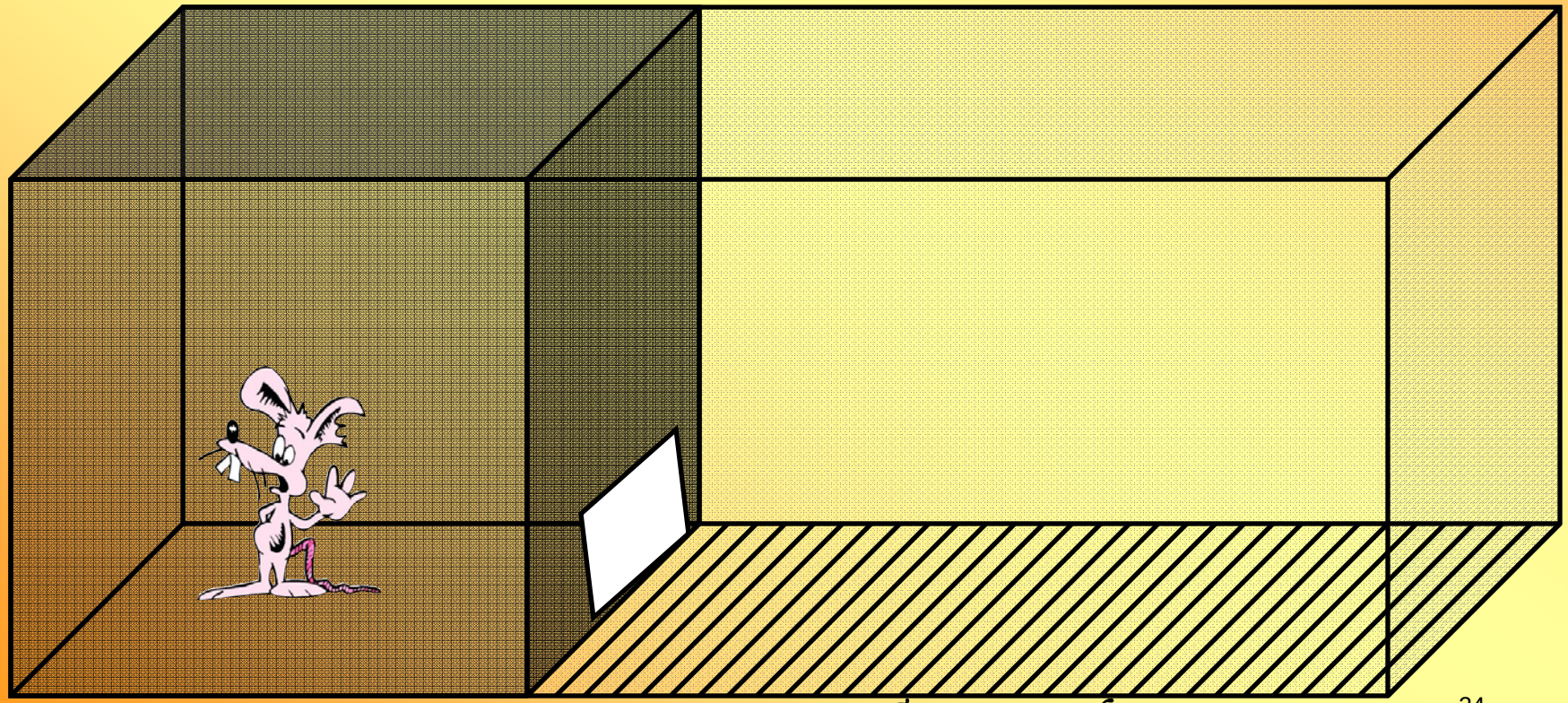
He explores. After 16 seconds, the buzzer sounds and we open the door



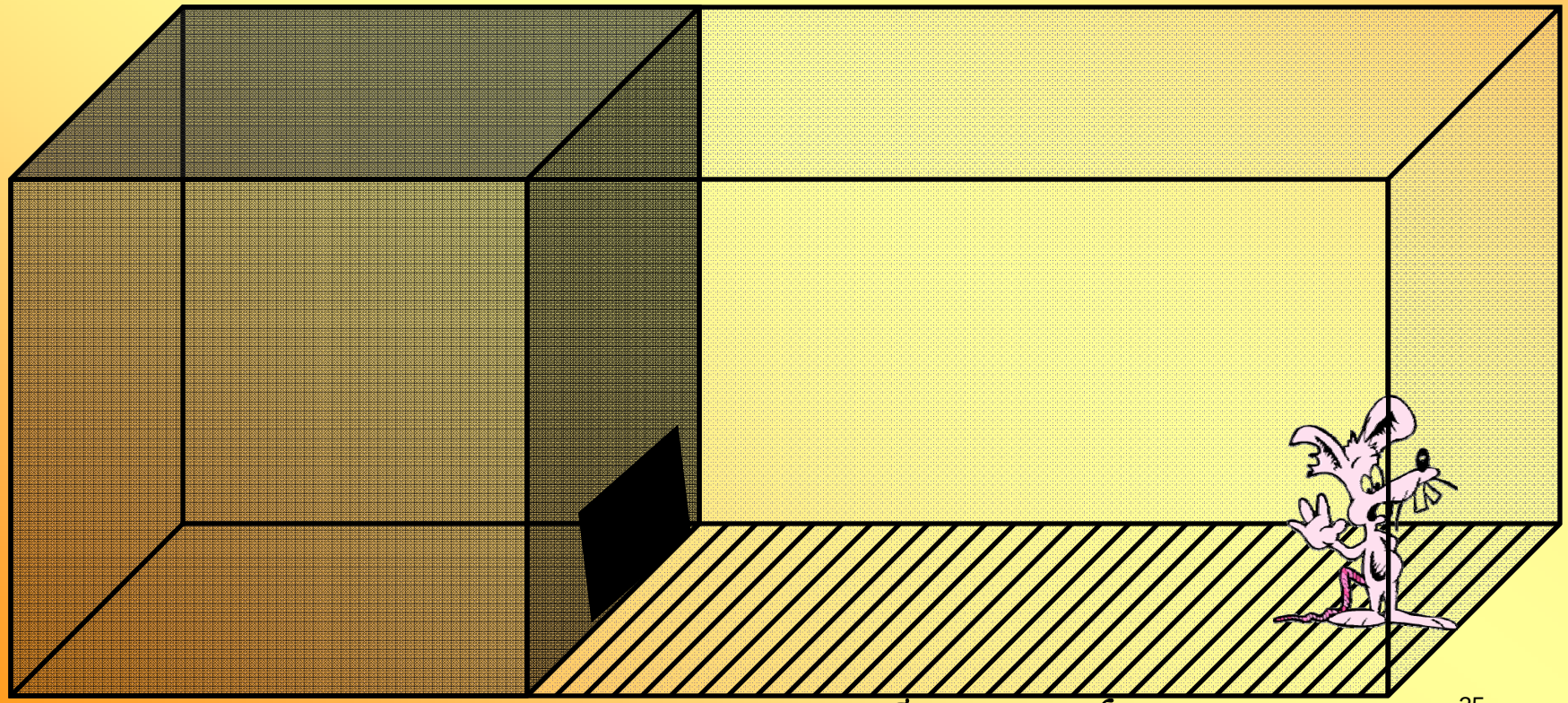
5 seconds later, Eeek! A shock.



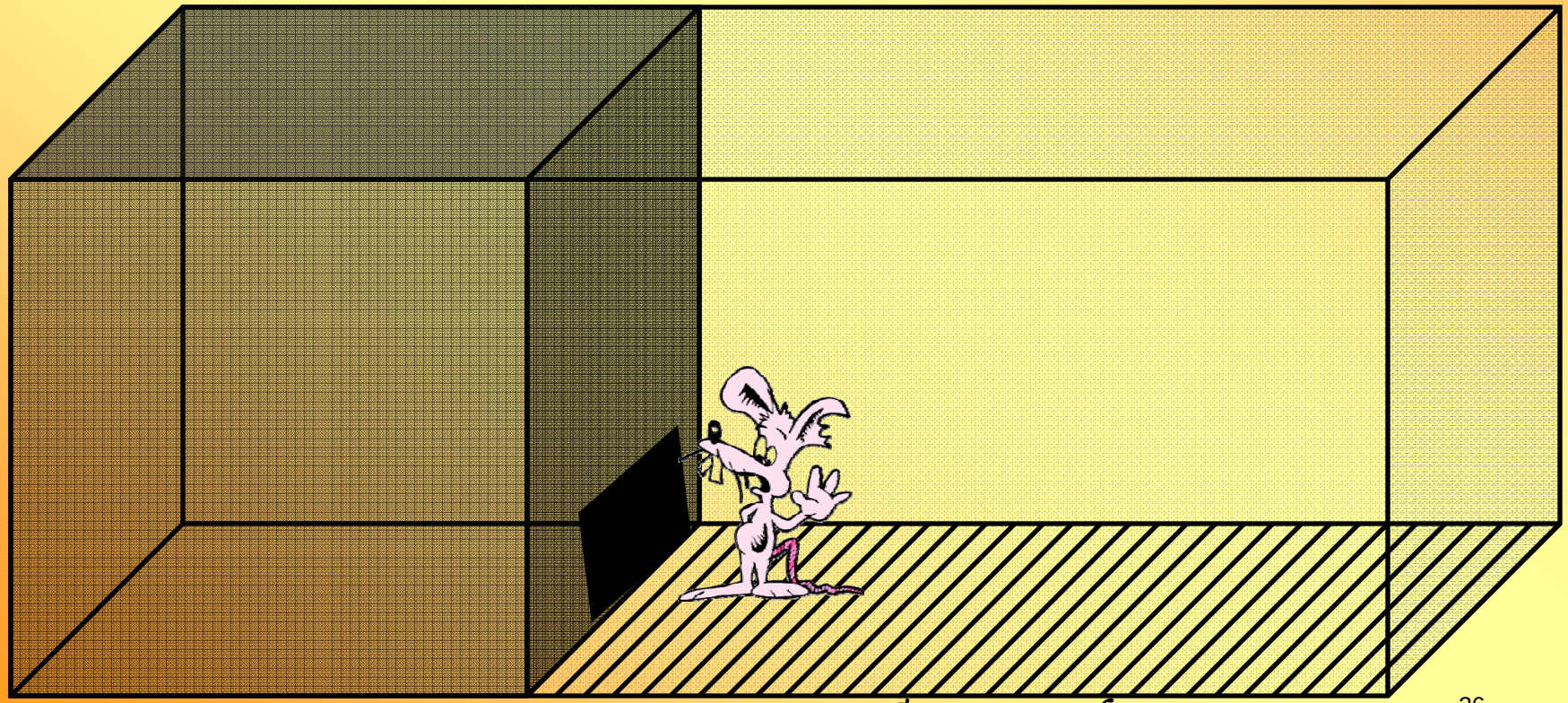
He bounces and runs until he gets to the dark side where there is no grid and no shock



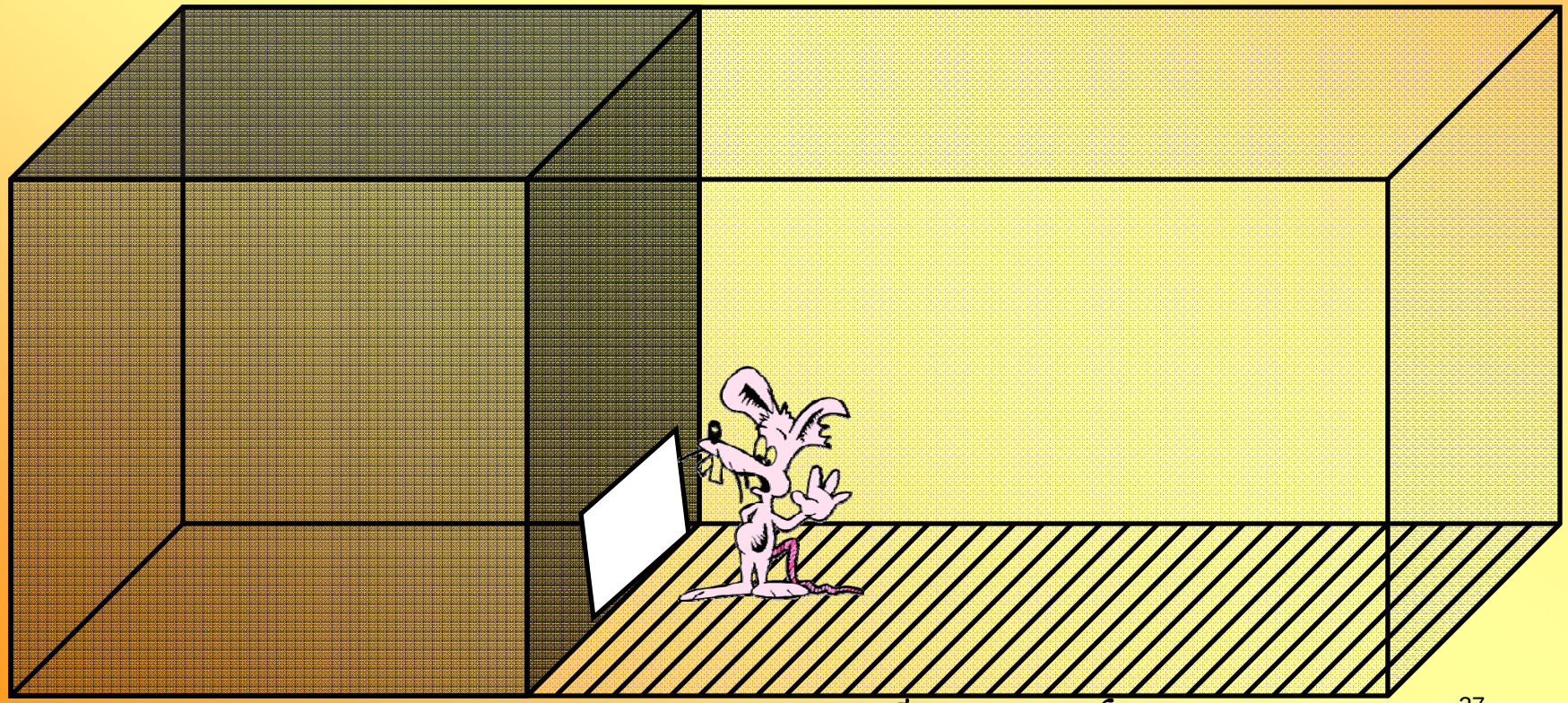
A minute later, we start again



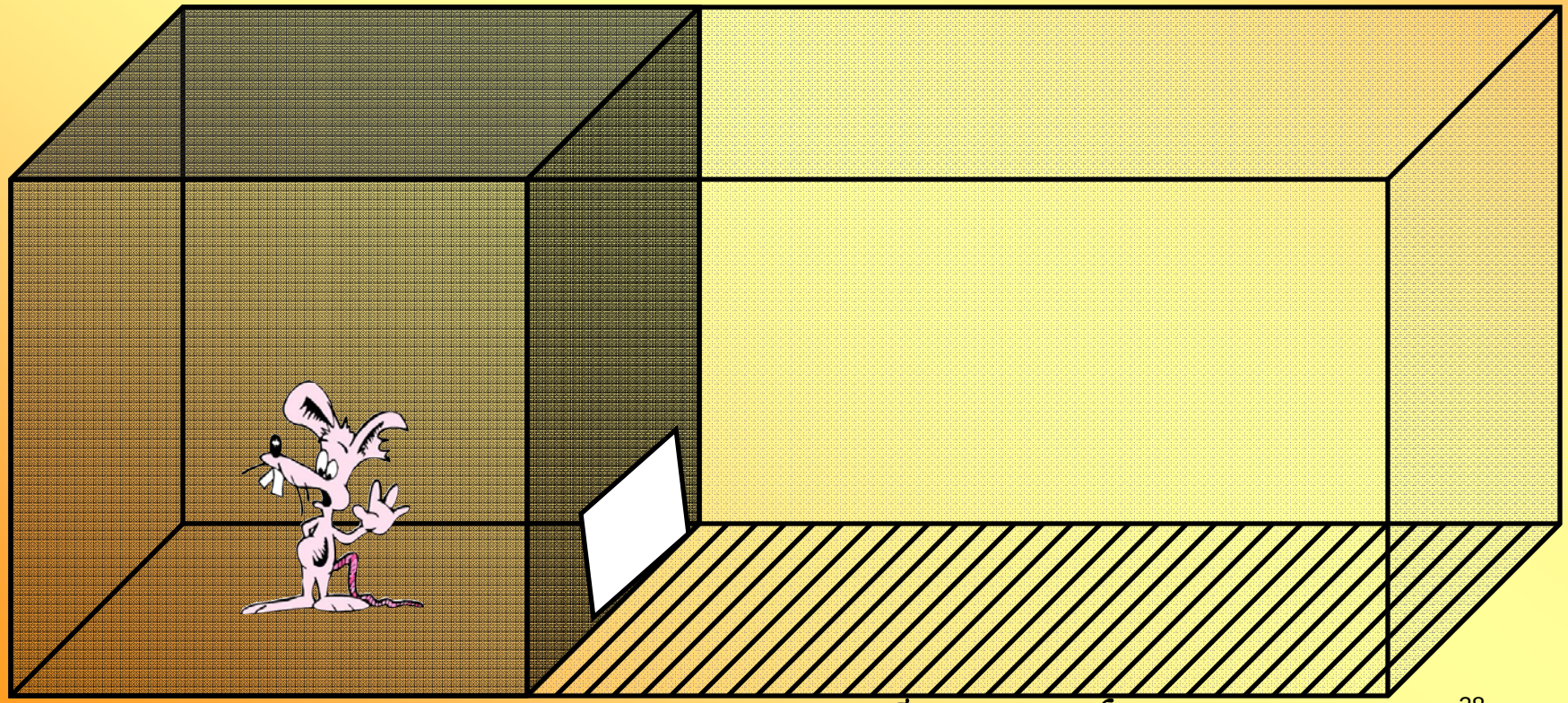
He comes to the front right away, sometimes chews on the closed door, trying to drag it open



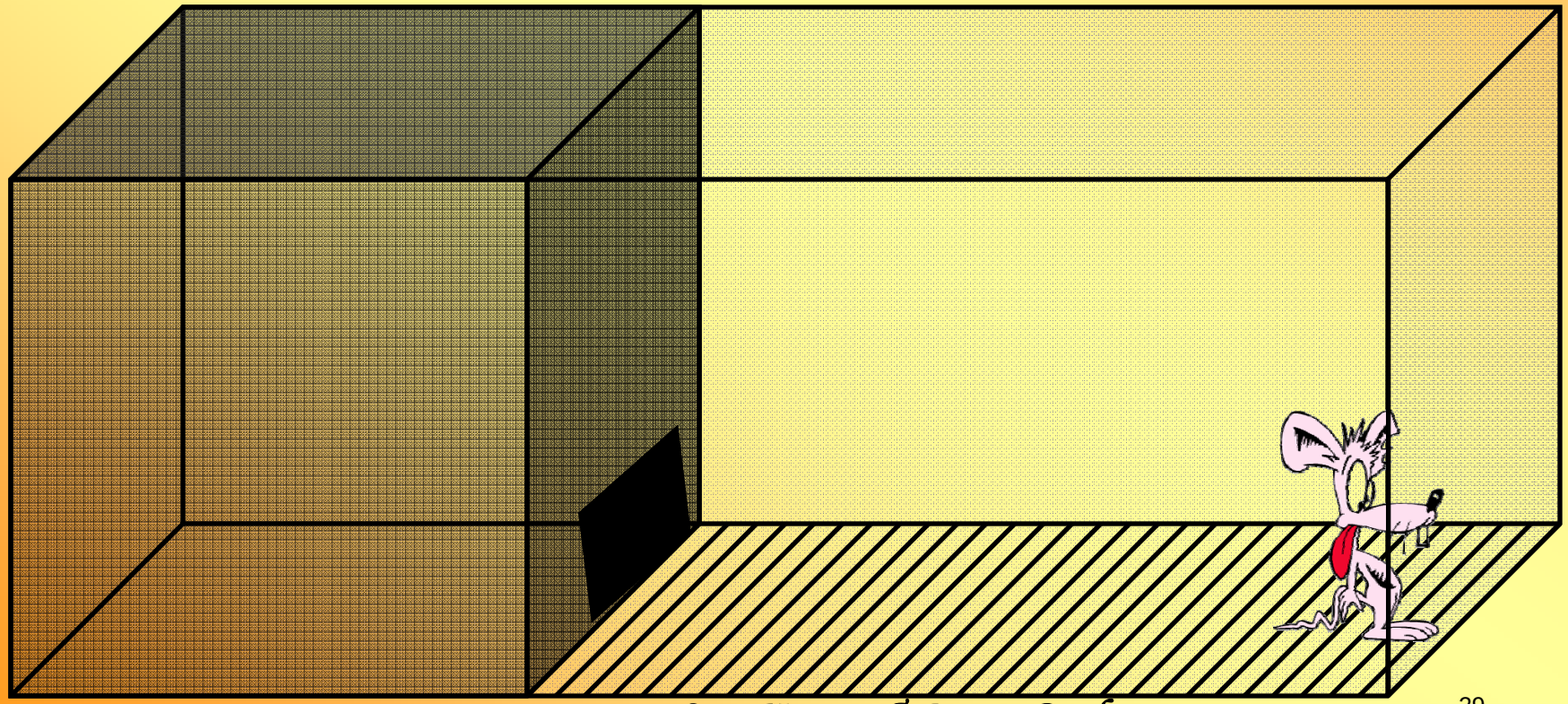
When the door opens …



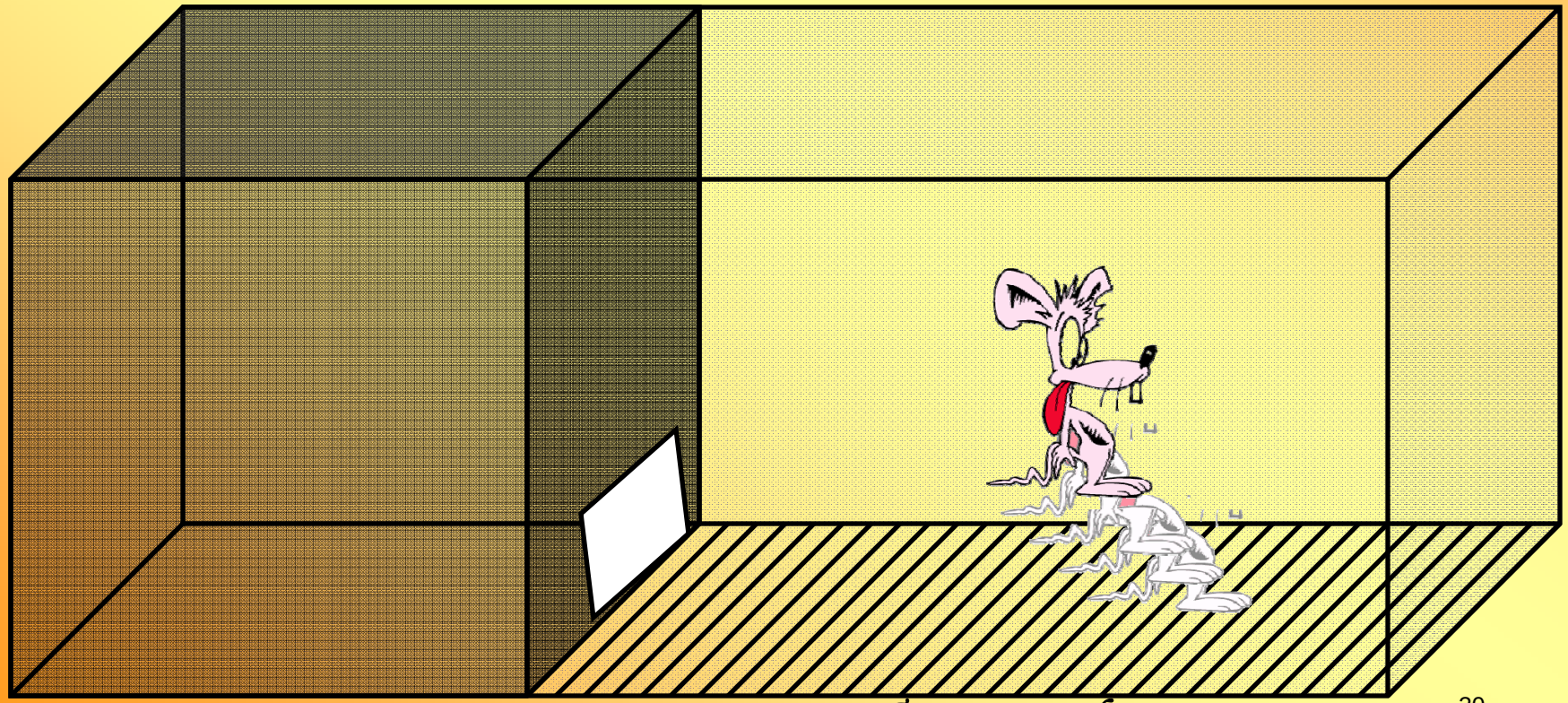
He runs to the dark side and stays there



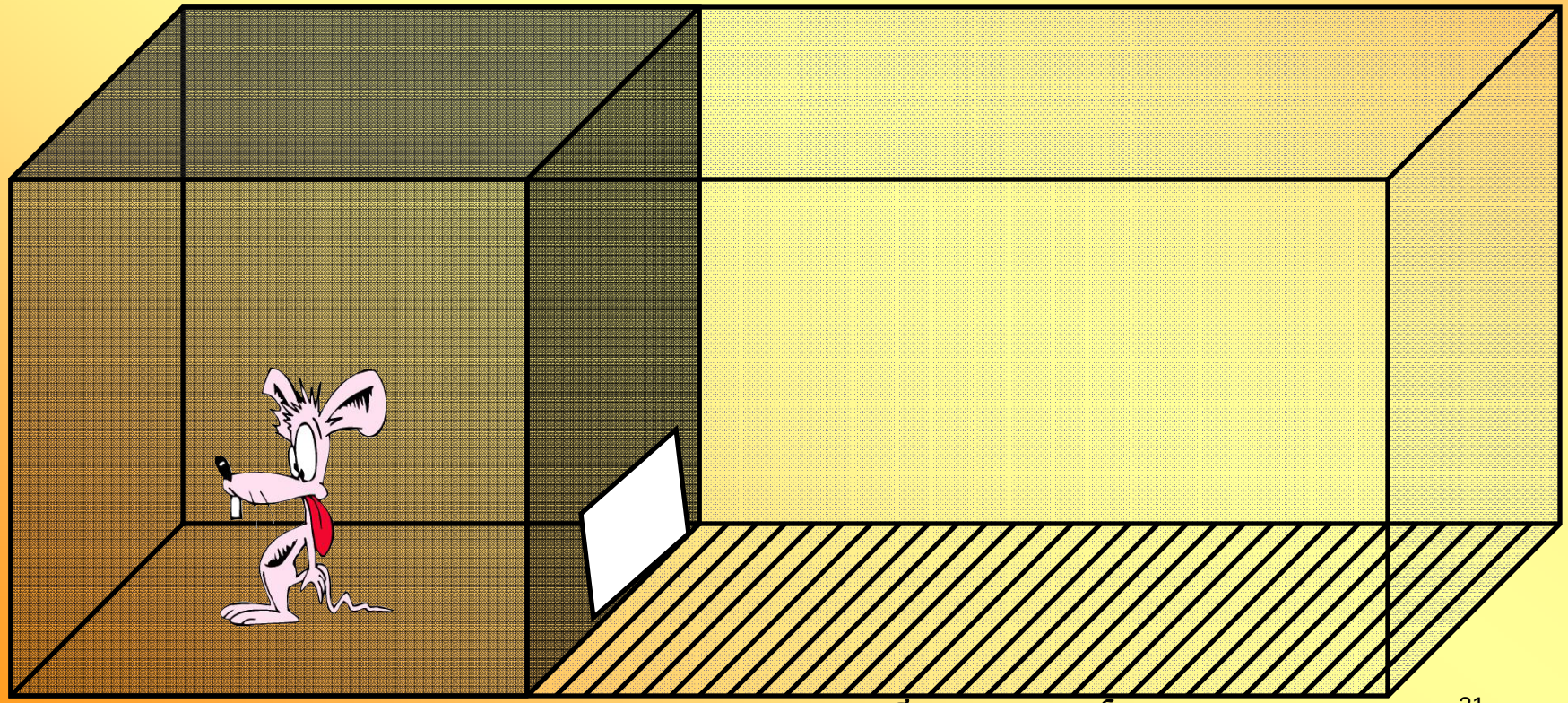
The experiment with the lesioned rat starts the same way. Explore first, then position him away from the door



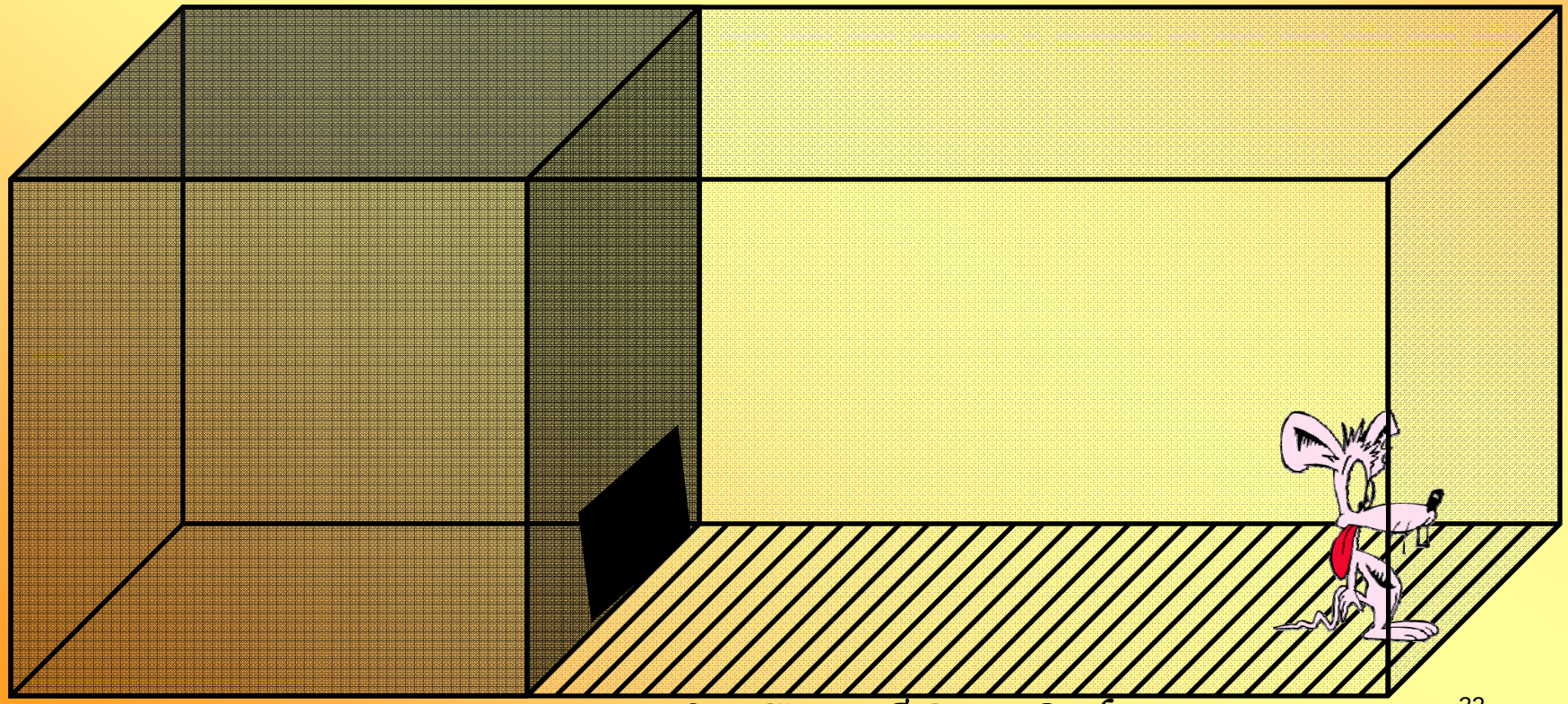
Eek! Ouch! Bounce!



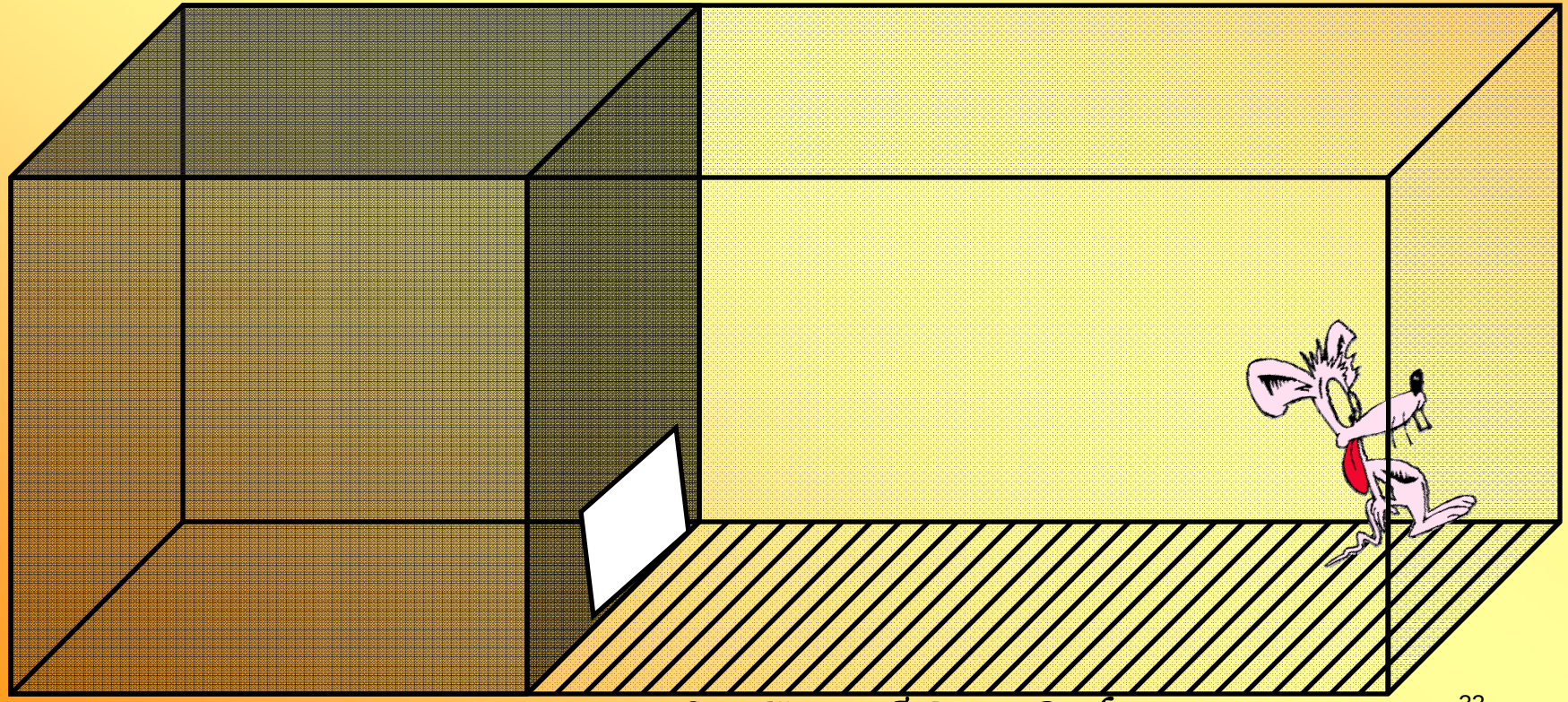
Safe, at last



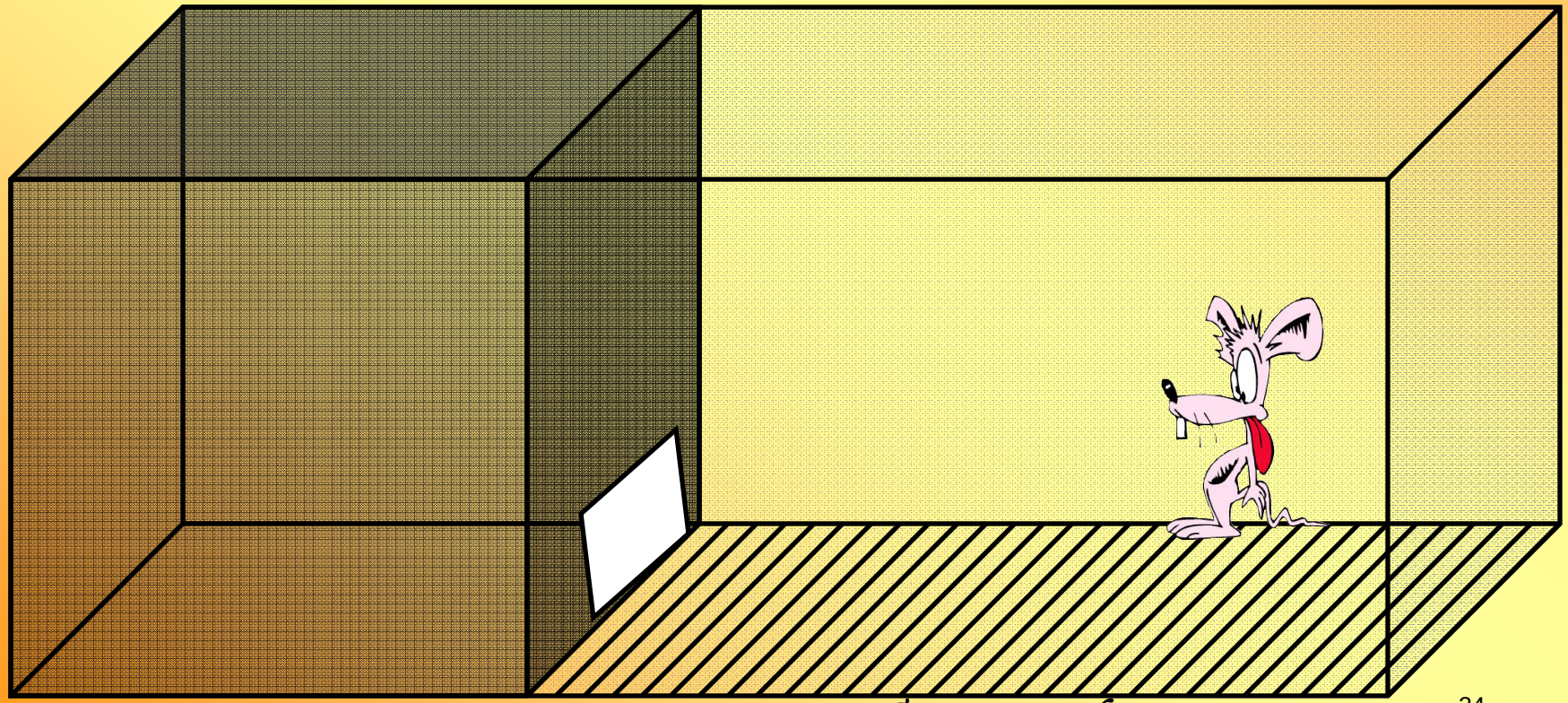
Next trials. Stays at the back for the full 16 seconds. Visibly frightened. Gets shocked a few more times.



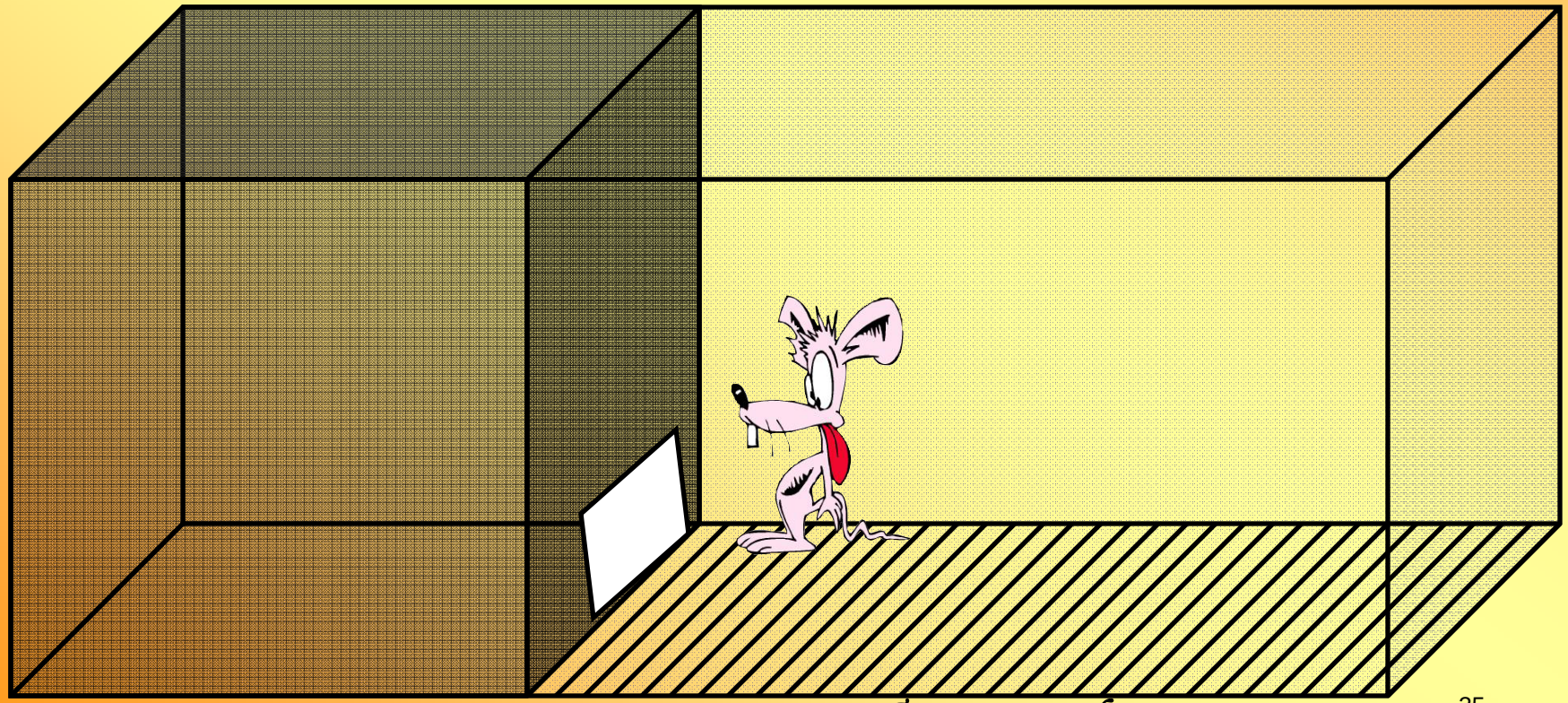
New strategy. Waits 16 seconds, then turns left. Goes to the wall.



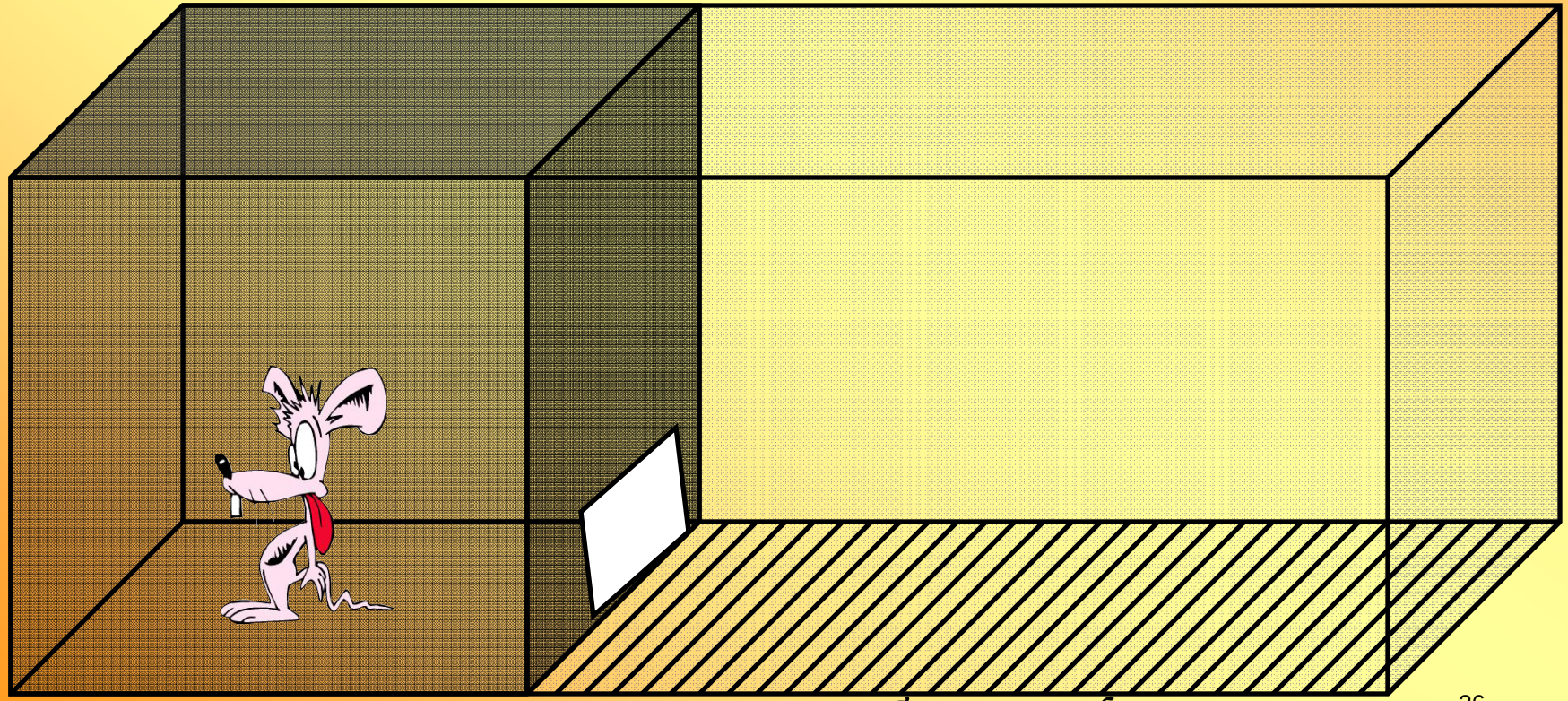
Turns left



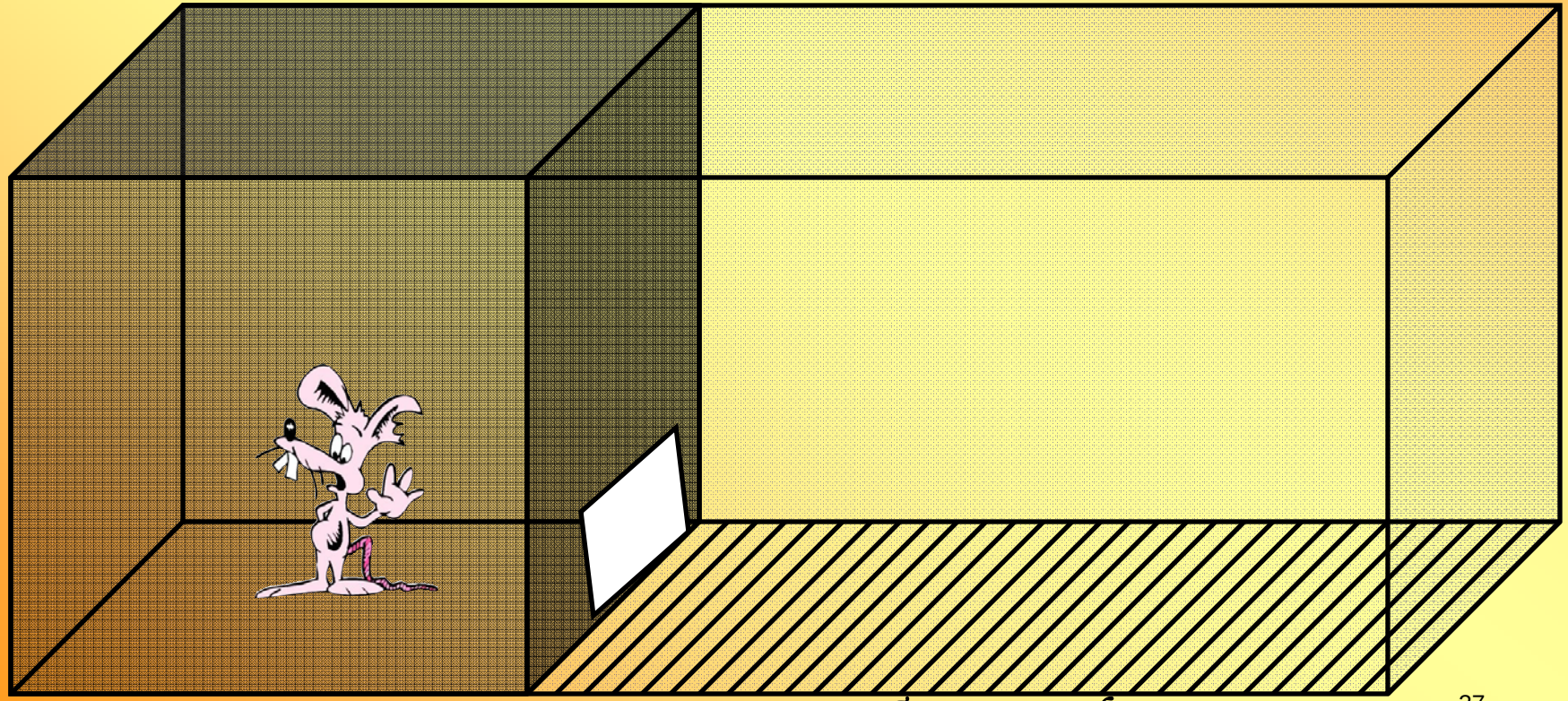
Runs to the wall



Turns left, runs to the door. And jumps through



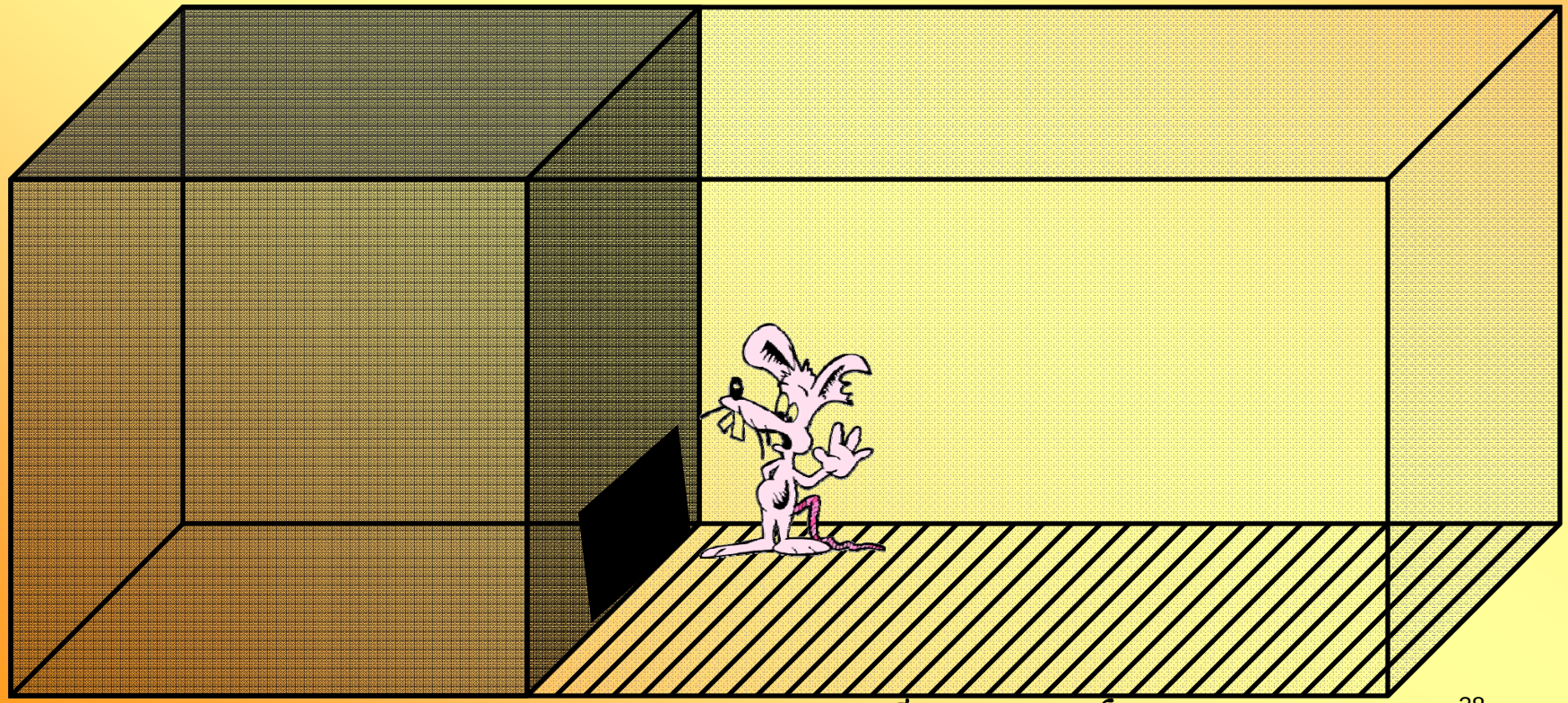
Same patterns again and again. The normals go to the front and try various things to get out right away. The fornicals wait, turn, run, turn, run, turn, run, turn into black side—follow the same procedure almost every time.



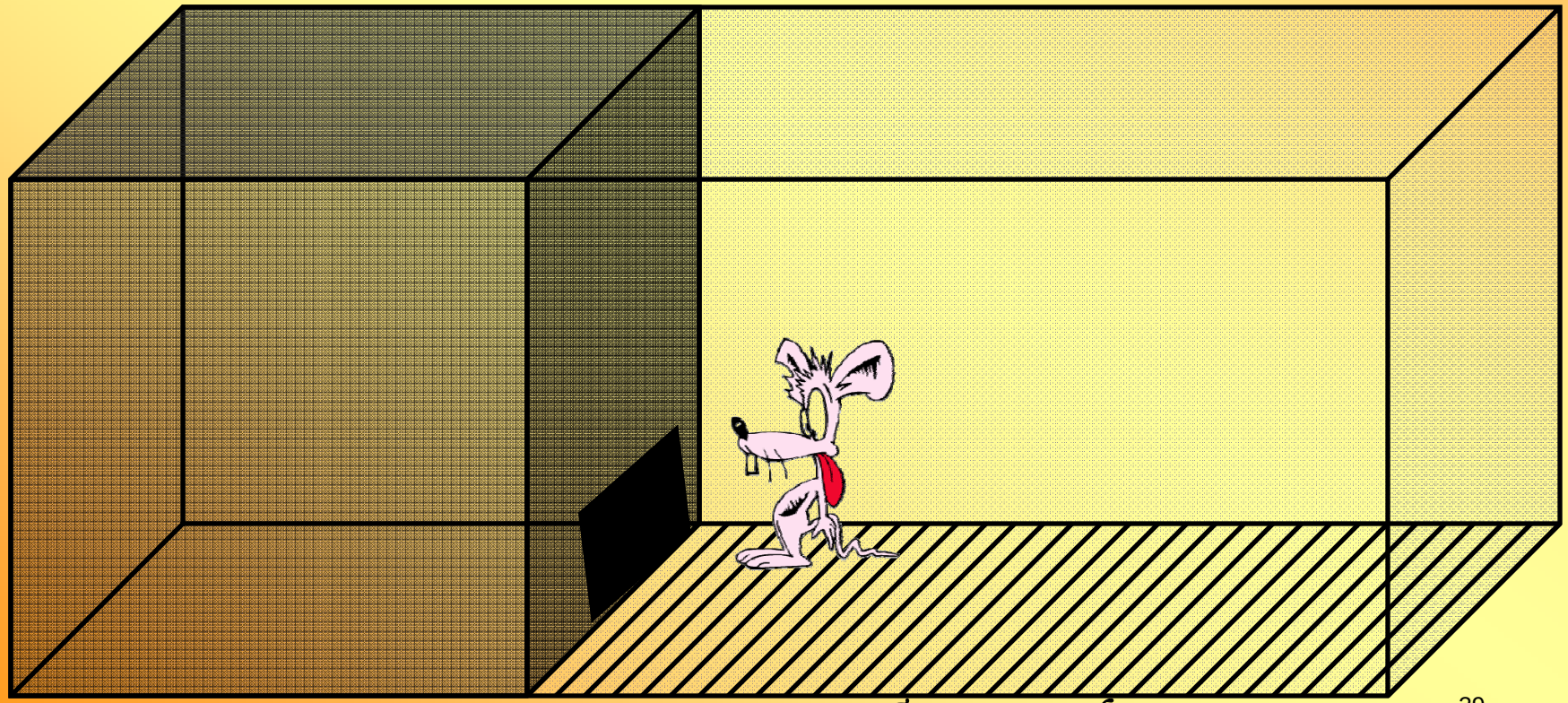
At the end, try some probes.

Example, put rat at front.

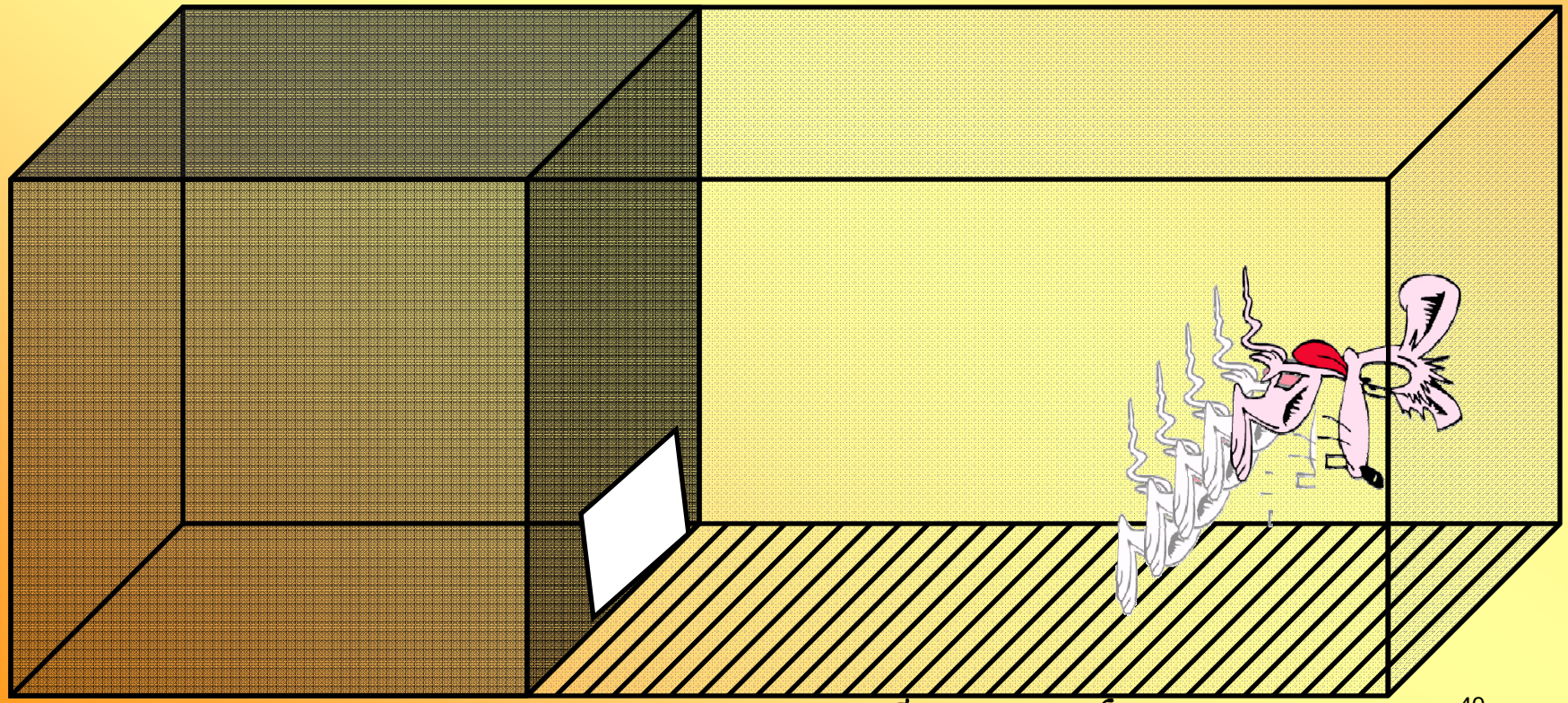
The normal stays at front and tries to get out. …



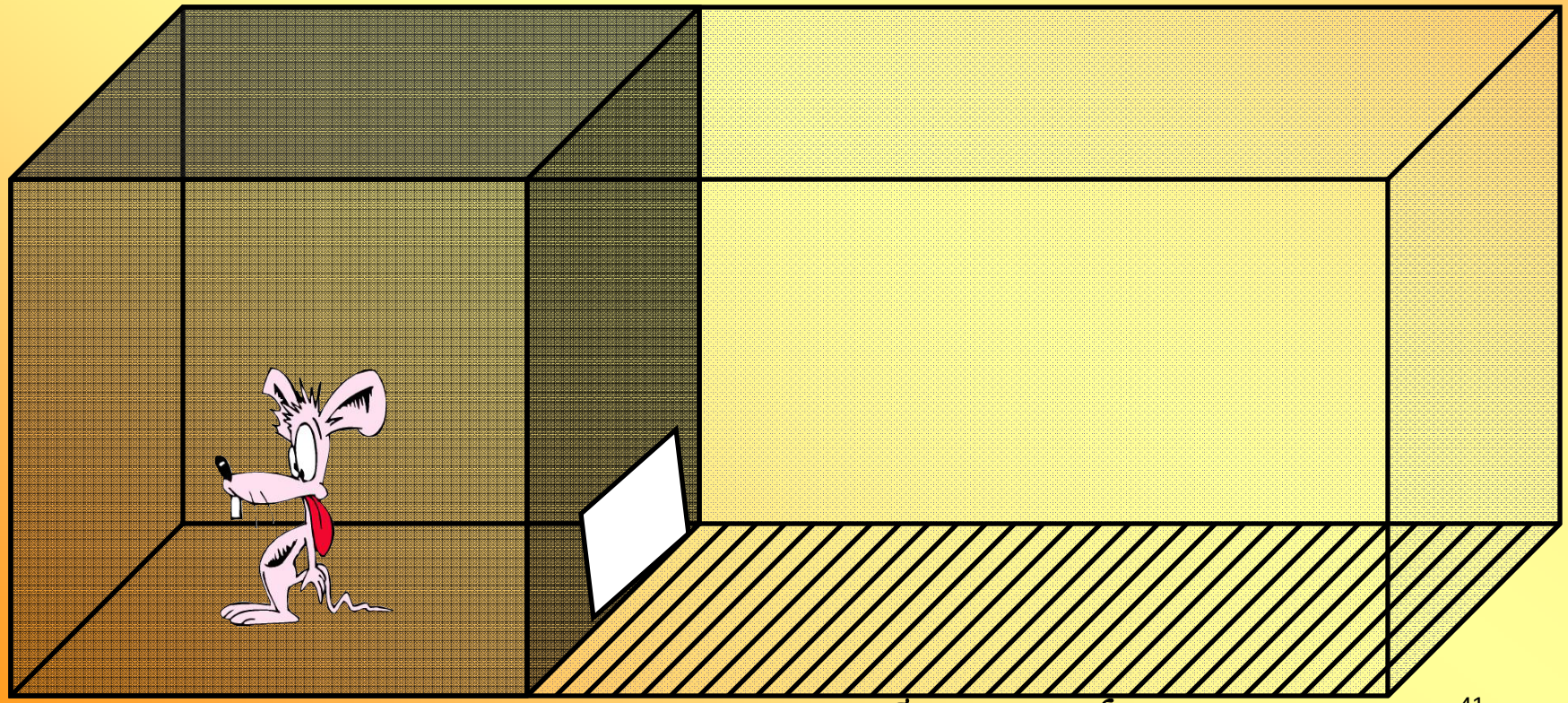
The lesioned rat stays at the front, waits, then turns and runs and turns and runs. …



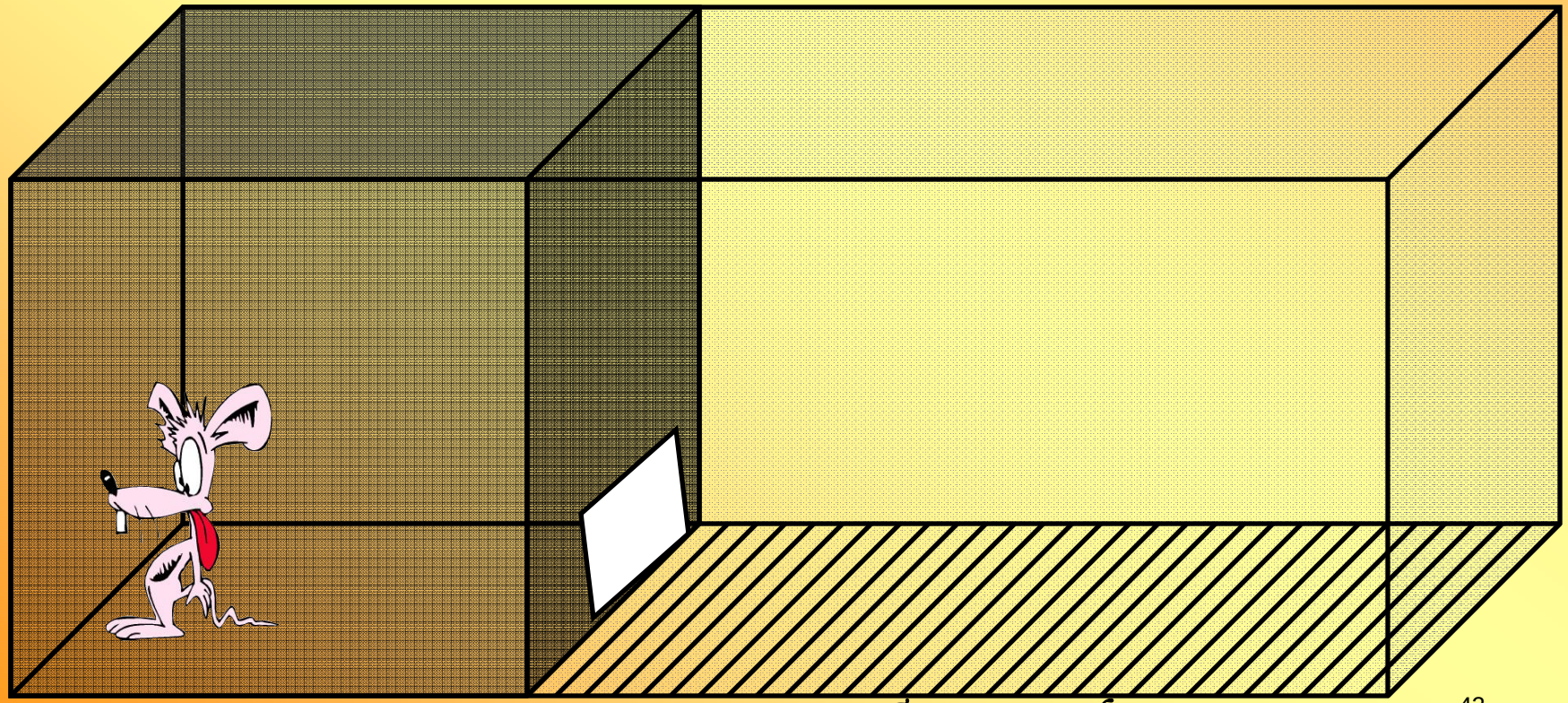
Several lesioned rats even jumped (as if they were jumping into the black side) and bounced off the wall. (Ouch!)



Then they turn and run and turn and run and turn and run and finally jump into the dark side, often taking a shock in the process.



In another probe, we put rats in the black side, waited 16 seconds and opened the door. Normals stayed in the black side. Fornicals often turned and ran and turned and jumped into shock.



In sum …

The normal rats seemed to understand where they were and where they wanted to go. They acted accordingly.

The lesioned rats seemed to use few external stimuli as cues, relying on rigid response patterns instead of their context.

Scripting and the testers' tour bus

- **The Greyhound Bus of software testing:**
  "Just relax and leave the thinking to us."
  - To the novice, the test script is the whole tour. The novice tester goes through the script, start to finish, and thinks he's seen what there is to see.
    - The novice follows the script.
  - To the experienced tester, the test script is a tour bus. When she sees something interesting, she stops the bus and takes a closer look.
    - The skilled tester breaks away from the script.
    - The skilled tester can also notice problems that the script writer didn't anticipate, and make comparisons that the script doesn't call for. An automated test is very different from a human test
    - In effect, the script hides what the skilled tester is doing.
- **One problem with a bus trip. It's often pretty boring, and you might spend a lot of time sleeping.**

Scripting doesn't live up to its promise

- **Not repeatable**: Different people following the same script do and notice different things.
- **What expertise are you scaling?** The person who writes the tests imagines a subset of the possible issues and, by the nature of scripting, is limited to a small set of tests that will be repeated over and over. See our discussion of GUI level regression--the repeated tests lack power. Worse, the script's focus on the same expected result every time steers the script user away from noticing other dimensions of (mis)behavior.
- **All the disadvantages of GUI-level capture/replay automation:** hard to maintain, no modularity, high costs of UI changes, unlikely to expose many bugs.
- **Misses the critical advantage of automation:** If you run a bunch of tests that are unlikely to yield much information (because you've run them before), at least with automated tests, you can minimize the time a human spends running them (additionally, automated tests can be better designed for maintainability).
- **No communication of analysis:** Scripts vary on this, but most focus on procedure without explaining the point of the test. Neither the tester who follows the script nor the one who maintains it knows what was in the mind of the test designer.
- **Ineffective way to educate**: Step by step procedural instruction is a poor way to teach someone about the program or about testing. It's not how people learn.

Scripting

- Chip Groeder created scripts as pseudocode, in preparation for test automation. Won the Best Paper award at STAR 97.
- He found most (88%) of the bugs during ***creation*** of the scripts (my experience too).
- Maybe the bug-finding benefit lies in structured exploration, rather than fully-specified regression.

Never do ANY kind of scripting?

- Checklists (as distinct from scripts) have their place. For example, think of a release checklist for a product:
  - Update your virus checkers
    - *It doesn't say which ones, it doesn't say how to update a virus checker*
  - Test the product with at least 2 virus checkers
    - *It doesn't tell you how to use a virus checker*
  - Check your set of files against the output of the source control system
- The checklists cover many very different tasks
  - All of them must be done
  - The overall task is rarely done, so many steps may be forgotten.

Alternatives to scripting?

- I prefer to tell testers what to test (what issues to cover), not how to do the tests. Teaching people "how" is a matter of training, not something that I record time and time again in the test plan. A checklist will sometimes be the right way to present the list of issues.
- I think the level of detail is sufficient if I can successfully pass the section to a reasonably experienced tester who is a little familiar with the program and be confident that she can figure out what the test cases are and how to run them.

In sum

- Can we call this a **worst practice**?

  (I want to, but James won't let me)

  – There **are** (some) valid contexts for test scripts
    - For example, they can give a detailed enough description of a test that an auditor can understand (or think she understands) what was done and what was seen.
  – So we see again that we can't declare a practice best or worst, that its value depends on its context (even in this case).
- Scripting is a prototypic factory school tool
  – One smart person defines and directs the work. The rest follow instructions
    - But it is rigid, inappropriate for soliciting fresh insights and observations of unexpected events in a task that is constantly changing: The product is changing. The types of problems of interest are changing.