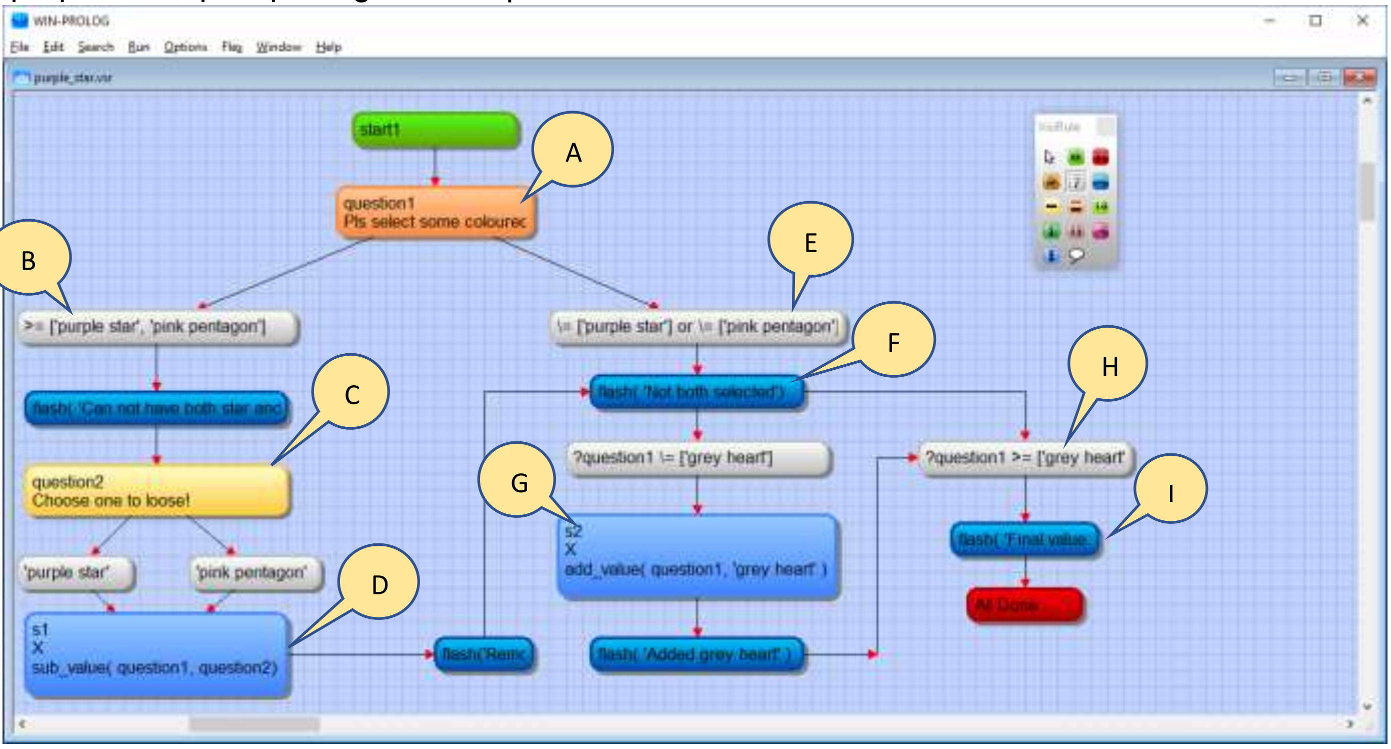
‘purple star', 'pink pentagon' example


WIN-PROLOG
File Edit Search Run Options Flex Window Help
purple_star.vsr
start1
A
question1
Pls select some colourec
B
>= ['purple star', 'pink pentagon']
E
\= ['purple star'] or \= ['pink pentagon']
F
flash('Not both selected')
flash('Can not have both star and
C
H
question2
Choose one to loose!
?question1 \= ['grey heart']
?question1 >= ['grey heart'
G
s2
X
add_value(question1, 'grey heart')
I
flash('Final value
All Done
'purple star'
'pink pentagon'
D
s1
X
sub_value(question1, question2)
flash('Rem
flash('Added grey heart')

Multi-choice example … purple star

- 2 questions (I multiple choice and 1 single choice)
- 6 expressions
- 2 statements
- 1 conclusions

In the above chart, we ask a multi-choice (multiple selections) question and then process the answer; this example uses 2 Comparison Operators (>= and \=). It also uses structured branching.

It demonstrates how to check for inclusion/exclusion of specific items, and how to force them into the answer set using a statement box and calls to some of the routines provided by the underlying Flex expert system.

A] question1

This is a multiple choice question which has 2 expressions hanging off of it; i.e. 2 branches to be explored.

As we will see later on, this question contains 3 items, the user can choose more than 1 item.

B] >= ['purple star', 'pink pentagon']

This is an expression which checks to see if the answer to question1 contains both ‘purple star' and ‘pink pentagon'

1] The LHS is empty, i.e. there is nothing before the Comparison Operator, so VisiRule knows we are using the last defined question

2] >= is a Comparison Operator which means that the question named or implied on the LHS must contain all the items in the RHS for the expression to succeed

3] Square brackets such as '[' and ']' are used to denote sets

4] ['purple star', 'pink pentagon'] denotes the set containing 2 items

This expression will succeed if the user has chosen both items (and then the single choice question will be reached which asks the user to remove one).

If the expression fails, VisiRule will try the other branch (with [E] which is designed to succeed when the first expression fails.

C] Question2

This is a single choice questions with two choices, 'purple star' and 'pink pentagon'

The user chooses one item

D] s1

This is a statement box which subtracts the answer to question2 from question1 using the built-in Flex routine sub_value/2

So now we have either 1 or 2 items left in question1. There is a link now over to the other branch.

E] \= ['purple star'] or \= ['pink pentagon']

This is an expression which checks to see that the answer does NOT contain both 'purple star' and 'pink pentagon'

1] The LHS is empty, there is nothing before the Comparison Operator, so VisiRule knows we are using the last question before the expression, i.e. question1

2] The RHS is a compound term which contains 2 expressions linked by an 'or' which is a logical operator

3] \= is a Comparison Operator which means that the LHS must not contain the item on the RHS; so \= ['purple star'] means the answer does not contain 'purple star'

4] \= is a Comparison Operator which means that the LHS must not contain the item on the RHS; so \= ['pink pentagon'] means the answer does not contain 'pink pentagon'

5] The expression requires at least one of the above to succeed, which is true so long as we don't contain both items.

This expression will succeed if the user has not chosen both items and then the next node will be visited.

F] ?question1 \= ['grey heart']

This is an expression which checks to see if the answer to question1 does NOT contain ‘grey heart'

1] The LHS is NOT empty, i.e. we name the question to use before the Comparison Operator, as we could be coming to this expression from the statement box too

2] \= is a Comparison Operator which means that the LHS must not contain the item on the RHS; so \= ['grey heart’] means the answer does not contain ‘grey heart’

This expression will succeed if the user has not chosen the item.

NOTE: This expression

a) refers back to a previously asked question by NAME
b) introduces a new item to question1.

VisiRule will look at all the expressions in advance and will build a list of available items for each named question. This is sometimes called ‘Harvesting’

G] s2

This is a statement box which adds ‘grey heart’ to the answer from question1 using the built-in Flex routine, add_value/2

NOTE: This combination ENSURES that ‘grey heart’ is in our answer set for question1

H] ?question1 >= ['grey heart']

This is an expression which checks that the answer to question1 does contain ‘grey heart'

1] The LHS is NOT empty, i.e. we name the question to use before the Comparison Operator, as we could be coming to this expression from the statement box too

2] >= is a Comparison Operator which means that the LHS must contain the items on the RHS; so >= ['grey heart’] means the answer does contain ‘grey heart’

This expression will succeed if the user has chosen the item or it has been added.

H] flash('Final value: '- question1)

This is a code box which will ‘flash’ a short message containing some fixed text ‘Final Value: ‘ and then the name of the question: question1. This name will be replaced by the current answer to that question

Note:

After [E], we have 2 branches which are both to be tried – this is structured branching as we are already pursuing the Right-Hand branch of the first set of expressions.

Note:

When analysing sets of answers, the \= operator can be considered as the DISJOINT operator.

i.e. A \= B succeeds if A and B are disjoint …. i.e. they share NO common members